

User Manual

Arista Networks

www.arista.com

Arista EOS version 4.31.1F

DOC-03495-31

Headquarters	Support	Sales
5453 Great America Parkway Santa Clara, CA 95054 USA +1-408-547-5500	+1-408-547-5502 +1-866-476-0000	+1-408-547-5501 +1-866-497-0000
www.arista.com/en/	support@arista.com	sales@arista.com

© Copyright 2023 Arista Networks, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of Arista Networks in the United States and other countries. Use of the Marks are subject to Arista Network Terms of Use Policy, available at www.arista.com/en/terms-of-use. Use of marks belonging to other parties is for informational purposes only.

Contents

Chapter 1: Overview....1
1.1 Command-Line Interface (CLI)....1
1.1.1 Accessing the EOS CLI....2
1.1.2 Processing Commands....2
1.1.3 Kernel-based Virtual Machine Commands and Configuration....9
1.1.4 Switch Platforms....14
1.1.5 Command Modes....22
1.1.6 Managing Switch Configuration Settings....25
1.1.7 Other Command-Line Interfaces....27
1.1.8 Directory Structure....27
1.1.9 Command-Line Interface Commands....28

Chapter 2: Booting the Switch....57
2.1 Boot Loader Aboot....57
2.2 Configuration Files....57
2.2.1 boot-config....57
2.2.2 running-config....62
2.2.3 startup-config....62
2.3 Supervisor Redundancy....62
2.3.1 Redundancy Supervisor Protocols....62
2.3.2 Configuring Supervisor Redundancy....63
2.4 System Reset....64
2.4.1 Typical Reset Sequence....64
2.4.2 Switch Recovery....65
2.4.3 Display Reload Cause....65
2.4.4 Configuring Zero Touch Provisioning....66
2.4.5 Configuring the Networks....68
2.5 Aboot Shell....68
2.5.1 Aboot Shell Operation....68
2.5.2 Accessing the Aboot Shell....69
2.5.3 Aboot File Structure....70
2.5.4 Booting From the Aboot Shell....70
2.5.5 Aboot Commands....71
2.6 Aboot Configuration Commands....72
2.6.1 CONSOLESPEED....72
2.6.2 NET Commands....72
2.6.3 install bios source....73
2.6.4 PASSWORD (ABOOT)....74
2.6.5 SWI....74
2.7 Switch Booting Commands....75
2.7.1 boot console....75
2.7.2 boot secret....76
2.7.3 boot system....77
2.7.4 delete startup-config....78
2.7.5 protocol....78
2.7.6 redundancy....79
2.7.7 redundancy manual switchover....80
2.7.8 reload....80

2.7.9 reload (scheduled)..... 82
2.7.10 show platform bios..... 83
2.7.11 show redundancy file-replication..... 84
2.7.12 show redundancy status..... 84
2.7.13 show redundancy switchover sso..... 85
2.7.14 show reload..... 85
2.7.15 show reload cause..... 86
2.7.16 show reload fast-boot..... 86

Chapter 3: Initial Configuration and Recovery..... 88

3.1 Initial Switch Access..... 88
3.1.1 Zero Touch Provisioning..... 88
3.1.2 Manual Provisioning..... 89
3.2 Connection Management..... 93
3.3 Switch Storage Device Secure Erase..... 94
3.4 Configure Session..... 95
3.4.1 Configuration Session..... 95
3.4.2 Configure Replace..... 96
3.4.3 Configuration CLI..... 96
3.5 Recovery Procedures..... 97
3.5.1 Removing the Enable Password from the Startup Configuration..... 97
3.5.2 Reverting the Switch to the Factory Default Startup Configuration..... 98
3.5.3 Restoring the Factory Default EOS Image and Startup Configuration..... 98
3.5.4 USB Support for ZeroTouch Provisioning..... 99
3.5.5 Restoring the Configuration and Image from a USB Flash Drive..... 101
3.6 Session Management Commands..... 101
3.6.1 configure replace..... 102
3.6.2 configure session..... 103
3.6.3 domain (XMPP Management)..... 104
3.6.4 idle-timeout (Console Management)..... 104
3.6.5 idle-timeout (SSH Management)..... 105
3.6.6 idle-timeout (Telnet Management)..... 106
3.6.7 management api http-commands..... 107
3.6.8 management console..... 107
3.6.9 management ssh..... 108
3.6.10 management telnet..... 109
3.6.11 management xmpp..... 110
3.6.12 protocol http (API Management)..... 110
3.6.13 protocol https (API Management)..... 111
3.6.14 protocol https certificate (API Management)..... 112
3.6.15 reset system storage secure..... 112
3.6.16 server (XMPP Management)..... 113
3.6.17 session privilege (XMPP Management)..... 114
3.6.18 show inventory..... 114
3.6.19 show xmpp neighbors..... 115
3.6.20 show xmpp status..... 116
3.6.21 show xmpp switch-group..... 116
3.6.22 shutdown (API Management)..... 117
3.6.23 shutdown (Telnet Management)..... 117
3.6.24 shutdown (XMPP Management)..... 118
3.6.25 switch-group (XMPP Management)..... 118
3.6.26 username (XMPP Management)..... 119
3.6.27 vrf (API Management)..... 120
3.6.28 vrf (XMPP Management)..... 121
3.6.29 xmpp send..... 121

3.6.30 xmpp session....122

Chapter 4: Administering the Switch....124

4.1 Managing the Switch Name....124
4.1.1 Assigning a Name to the Switch....124
4.1.2 Specifying DNS Addresses....125
4.2 System Clock and Time Protocols....126
4.2.1 Configuring the Time Zone....126
4.2.2 Setting the System Clock Manually....127
4.2.3 Displaying the Time....127
4.2.4 Network Time Protocol (NTP)....127
4.3 Managing Display Attributes....130
4.3.1 Banners....130
4.3.2 Configuring prompt....131
4.4 Logging of Event Notifications....131
4.4.1 Managing TCAM Capacity Warnings....131
4.5 Event Monitor....132
4.5.1 Description....132
4.5.2 Configuring the Event Monitor....133
4.5.3 Querying the Event Monitor....134
4.5.4 Accessing Event Monitor Database Records....135
4.6 Managing EOS Extensions....136
4.6.1 Installing EOS Extensions....136
4.6.2 Installing EOS Extensions on a Dual-Supervisor Switch....136
4.6.3 Verifying EOS Extensions Installation....137
4.6.4 Uninstalling an EOS Extension....137
4.7 Switch Administration Commands....138
4.7.1 banner login....139
4.7.2 banner motd....140
4.7.3 clear ptp interface counters....141
4.7.4 clock set....141
4.7.5 clock timezone....142
4.7.6 dns domain....143
4.7.7 email....143
4.7.8 event-monitor backup max-size....143
4.7.9 event-monitor backup path....144
4.7.10 event-monitor buffer max-size....145
4.7.11 event-monitor clear....146
4.7.12 event-monitor interact....146
4.7.13 event-monitor sync....146
4.7.14 event-monitor....147
4.7.15 hostname....148
4.7.16 ip domain lookup....148
4.7.17 ip domain-list....149
4.7.18 ip host....149
4.7.19 ip name-server....150
4.7.20 ipv6 host....151
4.7.21 logging format....152
4.7.22 logging persistent....153
4.7.23 logging repeat-messages....153
4.7.24 no event-monitor....154
4.7.25 ntp authenticate....155
4.7.26 ntp authentication-key....155
4.7.27 ntp local-interface....156
4.7.28 ntp serve all....157

4.7.29 ntp serve....157
4.7.30 ntp server....158
4.7.31 ntp trusted-key....160
4.7.32 power enable module....160
4.7.33 prompt....161
4.7.34 show banner....163
4.7.35 show clock....163
4.7.36 show event-monitor arp....164
4.7.37 show event-monitor igmpsnooping....165
4.7.38 show event-monitor mac....166
4.7.39 show event-monitor mroute....167
4.7.40 show event-monitor neighbor....168
4.7.41 show event-monitor route6....169
4.7.42 show event-monitor route....170
4.7.43 show event-monitor sqlite....171
4.7.44 show event-monitor stpunstable....172
4.7.45 show hostname....172
4.7.46 show hosts....173
4.7.47 show ip domain-name....173
4.7.48 show ip name-server....173
4.7.49 show local-clock time-properties....174
4.7.50 show ntp associations....174
4.7.51 show ntp status....175

Chapter 5: 7130 Layer 1 Configuration....176

5.1 l1 source....176
5.2 show l1 source....177
5.3 show l1 source capabilities....178
5.4 show l1 destination....178
5.5 show l1 matrix....179
5.6 show l1 path....182

Chapter 6: Timing Protocols....183

6.1 Setting the PTP Mode....183
6.2 Enabling PTP on an Interface....184
6.3 Configuring the PTP Domain....184
6.4 Configuring the Offset Hold Time....184
6.5 Setting the PTP Priority....185
6.6 Configuring the Source IP....185
6.7 Configuring the TTL for PTP Packets....185
6.8 Configuring PTP Monitoring....185
6.9 Setting the PTP Announce Interval....186
6.10 Setting the PTP Timeout Interval....187
6.11 Configuring the PTP Delay Mechanism....187
6.12 Setting the Delay Request Interval....187
6.13 Setting the Peer Delay Request Interval....187
6.14 Setting the Peer Link Propagation Threshold....188
6.15 Setting the Interval for Sending Synchronization Messages....188
6.16 Setting the PTP Transport Type....188
6.17 Viewing PTP Settings and Status....191
6.17.1 Displaying General PTP Information....192
6.17.2 Displaying PTP Clock Properties....192
6.17.3 Displaying PTP Foreign Master....192
6.17.4 Displaying PTP Information for all Interfaces....192

6.17.5 Displaying PTP Interface Counters..... 193
6.17.6 Displaying PTP Local Clock and Offset..... 193
6.17.7 Displaying PTP Masters Information..... 194
6.17.8 Displaying PTP Monitoring Information..... 194
6.17.9 Displaying PTP Source IP..... 194
6.18 Precision Time Protocol (PTP) Commands..... 195
6.18.1 ptp announce interval..... 195
6.18.2 ptp announce timeout..... 196
6.18.3 ptp delay-mechanism..... 197
6.18.4 ptp delay-req interval..... 197
6.18.5 ptp domain..... 198
6.18.6 ptp enable..... 198
6.18.7 ptp forward-v1..... 199
6.18.8 ptp hold-ptp-time..... 199
6.18.9 ptp local-priority..... 200
6.18.10 ptp mode..... 200
6.18.11 ptp monitor threshold mean-path-delay..... 201
6.18.12 ptp monitor threshold offset-from-master..... 202
6.18.13 ptp monitor threshold skew..... 202
6.18.14 ptp monitor..... 203
6.18.15 ptp pdelay-neighbor-threshold..... 203
6.18.16 ptp pdelay-req interval..... 204
6.18.17 ptp priority1..... 204
6.18.18 ptp priority2..... 205
6.18.19 ptp role..... 206
6.18.20 ptp source ip..... 206
6.18.21 ptp sync timeout..... 207
6.18.22 ptp sync-message interval..... 207
6.18.23 ptp transport..... 208
6.18.24 ptp ttl..... 209
6.18.25 ptp unicast-negotiation..... 209
6.18.26 show ptp..... 210
6.18.27 show ptp foreign-master-record..... 210
6.18.28 show ptp interface counters..... 211
6.18.29 show ptp interface..... 211
6.18.30 show ptp local-clock..... 212
6.18.31 show ptp masters..... 213
6.18.32 show ptp monitor..... 213
6.18.33 show ptp source ip..... 214
6.18.34 show ptp unicast-negotiation..... 214
6.18.35 show ptp unicast-negotiation profile..... 215

Chapter 7: Switch Environment Control..... 216
7.1 Environment Control Introduction..... 216
7.2 Environment Control Overview..... 216
7.2.1 Temperature..... 216
7.2.2 Fans..... 216
7.2.3 Power..... 217
7.3 Configuring and Viewing Environment Settings..... 217
7.3.1 Overriding Automatic Shutdown..... 217
7.3.2 Viewing Environment Status..... 219
7.3.3 Locating Components on the Switch..... 220
7.4 Environment Commands..... 221
7.4.1 environment fan-speed..... 221
7.4.2 environment insufficient-fans action..... 222

7.4.3 environment overheat action..... 223
7.4.4 locator-led.....224
7.4.5 show environment power.....225
7.4.6 show environment temperature..... 225
7.4.7 show locator-led.....227
7.4.8 show system environment all..... 227
7.4.9 show system environment cooling.....228
7.4.10 show system environment power budget..... 229

Chapter 8: Upgrades and Downgrades.....230
8.1 Upgrade/Downgrade Overview.....230
8.2 Smart System Upgrade.....230
8.2.1 Upgrading the EOS image with Smart System Upgrade.....231
8.3 Standard Upgrades and Downgrades.....236
8.3.1 Upgrading or Downgrading the EOS on a Single-Supervisor Switch..... 236
8.3.2 Upgrading or Downgrading the EOS on a Dual-Supervisor Switch.....240
8.4 Upgrade/Downgrade Commands..... 244
8.4.1 install.....244
8.4.2 reload fast-boot.....244
8.4.3 reload hitless.....245

Chapter 9: Security..... 247
9.1 User Security.....247
9.1.1 AAA Configuration..... 247
9.2 Control Plane Security.....316
9.2.1 Transport Layer Security..... 316
9.2.2 802.1X Port Security.....344
9.3 Data Plane Security.....373
9.3.1 IP NAT..... 373
9.3.2 Media Access Control Security.....398
9.3.3 Internet Protocol Security (IPsec).....436
9.3.4 Macro-Segmentation Service (CVX).....446

Chapter 10: Quality of Service and Traffic Management..... 475
10.1 Quality of Service..... 475
10.1.1 Quality of Service Conceptual Overview..... 475
10.1.2 QoS Configuration: Platform-Independent Features.....483
10.1.3 QoS Configuration: Arad Platform Switches.....485
10.1.4 QoS Configuration: Jericho Platform Switches.....494
10.1.5 QoS Configuration: FM6000 Platform Switches..... 503
10.1.6 QoS Configuration: Petra Platform Switches.....510
10.1.7 QoS Configuration: Trident and Tomahawk Platform Switches.....517
10.1.8 QoS Configuration: Trident II and Helix Platform Switches.....526
10.1.9 Support for Configuring Color Extended Communities.....535
10.1.10 ACL based QoS Configuration..... 537
10.1.11 Configuring IPv6 Flow Label Matches for QoS..... 539
10.1.12 Differentiated MMU Discard Counters.....541
10.1.13 Quality of Service Commands.....542
10.1.14 Chipset Mapping for QoS..... 613
10.2 Traffic Management.....616
10.2.1 Traffic Management Conceptual Overview.....617
10.2.2 Traffic Management Configuration Arad Platform Switches..... 630
10.2.3 Traffic Management Configuration FM6000 Platform Switches.....638

10.2.4 Traffic Management Configuration Petra Platform Switches..... 647
10.2.5 Traffic Management Configuration Trident Platform Switches.....650
10.2.6 Traffic Management Configuration Trident II Platform Switches.....660
10.2.7 Traffic Management Configuration Commands..... 663

Chapter 11: Interface Configuration.....722
11.1 Ethernet Ports.....722
11.1.1 Ethernet Ports Introduction..... 722
11.1.2 Ethernet Standards..... 722
11.1.3 Ethernet Physical Layer.....724
11.1.4 Interfaces.....735
11.1.5 MRU Enforcement..... 740
11.1.6 Ethernet Configuration Procedures.....741
11.1.7 Ethernet Configuration Commands.....776
11.2 Port Channels and LACP.....829
11.2.1 Port Channel Introduction.....829
11.2.2 Port Channel Conceptual Overview..... 829
11.2.3 Port Channel Configuration Procedures.....831
11.2.4 Load Balancing Hash Algorithms..... 839
11.2.5 Port Channel and LACP Configuration Commands..... 845
11.3 Multi-Chassis Link Aggregation.....885
11.3.1 MLAG Introduction.....885
11.3.2 MLAG Conceptual Overview..... 886
11.3.3 MLAG Maintenance..... 887
11.3.4 MLAG Dual Primary Detection and Release.....890
11.3.5 Configuring MLAG..... 894
11.3.6 EVPN - MLAG Single-homed Hosts.....901
11.3.7 MLAG Implementation Example.....902
11.3.8 MLAG Commands..... 908
11.4 Data Transfer.....920
11.4.1 Data Transfer Introduction.....921
11.4.2 Data Transfer Methods.....921
11.4.3 MAC Address Table..... 924
11.4.4 Configuring Ports..... 928
11.4.5 Monitoring Links.....944
11.4.6 PHY test pattern CLI..... 956
11.4.7 Data Transfer Commands..... 961
11.5 Octal Port Renumber to Four Interfaces.....1022
11.5.1 Configuring Octal Port Renumber to 4 Interfaces..... 1023
11.5.2 Show Commands.....1024
11.5.3 Limitations..... 1024
11.5.4 Commands.....1024

Chapter 12: Layer 2 Configuration.....1025
12.1 Spanning Tree Protocol.....1025
12.1.1 Introduction to Spanning Tree Protocols..... 1025
12.1.2 Spanning Tree Overview..... 1025
12.1.3 Configuring a Spanning Tree.....1031
12.1.4 STP Commands.....1045
12.2 Link Layer Discovery Protocol.....1087
12.2.1 LLDP Introduction..... 1088
12.2.2 LLDP Overview.....1088
12.2.3 LLDP Configuration Procedures..... 1089
12.2.4 LLDP Configuration Commands..... 1094

12.3 Virtual LANs (VLANs).....1105
12.3.1 VLAN Introduction.....1106
12.3.2 VLAN Conceptual Overview.....1106
12.3.3 VLAN Configuration Procedures.....1109
12.3.4 VLAN Configuration Commands.....1119
12.4 DCBX and Flow Control.....1143
12.4.1 Introduction.....1144
12.4.2 Overview.....1144
12.4.3 DCBX Configuration and Verification.....1144
12.4.4 Configuring Priority-Flow-Control (PFC).....1146
12.4.5 Configuring PFC Watchdog.....1146
12.4.6 DCBX and Flow Control Commands.....1149
12.5 IP Locking.....1163
12.5.1 IP Locking.....1164
12.6 Layer 2 Protocol Forwarding.....1174
12.6.1 Configuring L2 Protocol Forwarding.....1174
12.6.2 L2 Protocol Forwarding Limitations.....1181
12.6.3 L2 Protocol Forwarding Show commands.....1181
12.7 Layer 2 Subinterfaces.....1182
12.7.1 Configurations.....1183
12.7.2 QoS Show Commands.....1187

Chapter 13: Layer 3 Configuration.....1189
13.1 IPv4.....1189
13.1.1 IPv4 Addressing.....1189
13.1.2 IPv4 Routing.....1198
13.1.3 IPv4 Multicast Counters.....1202
13.1.4 Route Management.....1205
13.1.5 IPv4 Route Scale.....1213
13.1.6 IP Source Guard.....1219
13.1.7 DHCP Server.....1225
13.1.8 DHCP Relay Global Configuration Mode.....1236
13.1.9 DHCP Relay Across VRF.....1238
13.1.10 DHCP Relay in VXLAN EVPN.....1241
13.1.11 DHCP Snooping with Bridging.....1242
13.1.12 TCP MSS Clamping.....1243
13.1.13 IPv4 GRE Tunneling.....1248
13.1.14 GRE Tunneling Support.....1251
13.1.15 BfRuntime to use non-default VRFs.....1256
13.1.16 IPv4 Commands.....1256
13.2 IPv6.....1332
13.2.1 Introduction.....1332
13.2.2 IPv6 Description.....1333
13.2.3 Configuring IPv6.....1335
13.2.4 IPv6 Commands.....1345
13.3 Ingress and Egress Per-Port for IPv4 and IPv6 Counters.....1381
13.3.1 Configuration.....1381
13.3.2 Show Commands.....1382
13.3.3 Dedicated ARP Entry for TX IPv4 and IPv6 Counters.....1382
13.3.4 Limitations.....1383
13.4 ACLs and Route Maps.....1383
13.4.1 ACL, Service ACL, Route Map, Prefix List, and RACL Divergence Introduction.....1383
13.4.2 Access Control Lists.....1383
13.4.3 Service ACLs.....1401
13.4.4 Sub-interface ACLs.....1405

13.4.5 RACL Sharing on SVIs....1406
13.4.6 Route Maps....1408
13.4.7 Prefix Lists....1413
13.4.8 Port ACLs with User-Defined Fields....1417
13.4.9 ACL, Route Map, and Prefix List Commands....1419
13.5 VRRP and VARP....1477
13.5.1 VRRP and VARP Conceptual Overview....1478
13.5.2 VRRP and VARP Implementation Procedures....1479
13.5.3 VRRP and VARP Implementation Examples....1486
13.5.4 VRRP and VARP Configuration Commands....1490
13.6 DirectFlow....1509
13.6.1 Introduction....1509
13.6.2 DirectFlow Configuration....1510
13.6.3 DirectFlow Feature Interactions....1512
13.6.4 DirectFlow Commands....1515
13.7 Decap Groups....1527
13.7.1 Decap Groups Description....1528
13.7.2 Decap Groups Configuration....1529
13.7.3 Decap Commands....1530
13.8 Nexthop Groups....1534
13.8.1 Nexthop Group Description....1534
13.8.2 Nexthop Group Configuration....1535
13.8.3 Support for IPv6 Link-Local Addresses in Nexthop Groups Entries....1538
13.8.4 Nexthop Group Commands....1539
13.9 Global Knob to Set MTU for all Layer 3 Interfaces....1546
13.9.1 Global Knob to Set MTU for all Layer 3 Interfaces Configuration....1547
13.9.2 Limitations....1547
13.9.3 Show Commands....1547
13.10 Support for Layer 3 MTU on 7280R3/7500R3/7800R3....1547
13.10.1 Layer 3 MTU Configuration....1548
13.10.2 Layer 3 MTU Show Commands....1548
13.10.3 L3 MTU Limitations....1548
13.11 Segment Security....1549
13.11.1 Overview of MSS-Group....1549
13.11.2 Configuring MSS-Group....1549
13.11.3 Limitations....1552
13.11.4 Show Commands....1552
13.11.5 Segment Security Commands....1554

Chapter 14: IP Services....1568
14.1 CloudVision eXchange (CVX)....1568
14.1.1 Upgrading CVX....1568
14.1.2 CVX Overview....1569
14.1.3 CVX Services....1570
14.1.4 Deploying CVX....1571
14.1.5 CVX Configuration....1578
14.1.6 CVX Secure out-of-band Connection....1584
14.1.7 CVX High Availability....1588
14.1.8 CVX VIP....1593
14.1.9 CVX Commands....1594

Chapter 15: Routing Protocols....1613
15.1 Routing Information Protocol (RIP)....1613
15.1.1 RIP Conceptual Overview....1613

15.1.2 Running RIP on the Switch....1614
15.1.3 Configuring RIP on Multiple VRFs....1617
15.1.4 RIP Commands....1617
15.2 Open Shortest Path First – Version 2....1626
15.2.1 OSPFv2 Introduction....1626
15.2.2 OSPFv2 Conceptual Overview....1626
15.2.3 Configuring OSPFv2....1629
15.2.4 OSPFv2 Configuration Examples....1656
15.2.5 OSPFv2 Commands....1665
15.3 Open Shortest Path First – Version 3....1719
15.3.1 OSPFv3 Introduction....1719
15.3.2 OSPFv3 Conceptual Overview....1720
15.3.3 Configuring OSPFv3....1723
15.3.4 OSPFv3 Configuration Examples....1737
15.3.5 OSPFv3 Commands....1745
15.4 IS-IS....1791
15.4.1 IS-IS Introduction....1791
15.4.2 IS-IS Segment Routing....1791
15.4.3 IS-IS Graceful Restart....1793
15.4.4 IS-IS Dynamic Flooding....1793
15.4.5 IS-IS Configuration....1794
15.4.6 IS-IS Commands....1827
15.5 Border Gateway Protocol (BGP)....1876
15.5.1 BGP Conceptual Overview....1876
15.5.2 Configuring BGP....1891
15.5.3 BGP IPv6 Link Local Peers Discovery....1945
15.5.4 BGP Examples....1945
15.5.5 BGP Commands....1948
15.6 Maintenance Mode....2055
15.6.1 Overview....2055
15.6.2 Maintenance Mode Elements....2056
15.6.3 Maintenance Mode Features....2060
15.6.4 Maintenance Mode Configuration....2062
15.6.5 Maintenance Mode Commands....2070
15.7 Bidirectional Forwarding Detection....2104
15.7.1 Introduction....2104
15.7.2 BFD Configuration....2105
15.7.3 Hardware Accelerated BFD Transmit....2111
15.7.4 BFD Commands....2114

Chapter 16: Multicast....2128

16.1 Multicast Architecture....2128
16.1.1 Overview....2128
16.1.2 Multicast Architecture Description....2128
16.1.3 Multicast Listener Discovery (MLD)....2130
16.1.4 Multicast Route Counters....2136
16.1.5 Multicast (S,G) Counters....2137
16.1.6 Static IP Mroute....2140
16.1.7 Static Multicast....2143
16.1.8 Configuring Multicast....2144
16.1.9 Multicast Commands....2153
16.2 IGMP and IGMP Snooping....2181
16.2.1 IGMP Snooping....2181
16.2.2 IGMP Host Proxy Description....2182
16.2.3 Supported Features....2182

16.2.4 IGMP Protocols....2182
16.2.5 Configuring IGMP....2183
16.2.6 Configuring IGMP Snooping....2185
16.2.7 IGMP Host Proxy....2195
16.2.8 IGMP and IGMP Snooping Commands....2197
16.3 Protocol Independent Multicast....2254
16.3.1 Introduction....2254
16.3.2 Overview....2254
16.3.3 Configuring PIM....2256
16.3.4 Multicast Example....2263
16.3.5 PIM Commands....2266
16.4 Multicast Source Discovery Protocol (MSDP)....2296
16.4.1 MSDP Introduction....2296
16.4.2 MSDP Description....2296
16.4.3 MSDP Configuration....2298
16.4.4 MSDP Commands....2306
16.5 Audio Video Bridging (AVB)....2323
16.5.1 AVB Overview....2323
16.5.2 AVB Protocols....2324
16.5.3 AVB Configuration....2325
16.5.4 AVB Commands....2329

Chapter 17: Virtual Extensible LANs (VXLANs)....2337
17.1 VXLAN Introduction....2337
17.2 VXLAN Description....2337
17.2.1 VXLAN Architecture....2337
17.2.2 VXLAN Gateway....2338
17.2.3 VXLAN Processes....2339
17.2.4 Multicast and Broadcast over VXLAN....2339
17.2.5 VXLAN and MLAG....2340
17.2.6 VXLAN Bridging and Routing Support....2343
17.2.7 Data Structures....2343
17.3 VXLAN Configuration....2343
17.3.1 Configuring the VTI....2344
17.3.2 Head End Replication Configuration....2346
17.3.3 VXLAN Routing Configuration....2346
17.3.4 Configuring VXLAN Routing with Overlay VRFs....2352
17.3.5 Configuring VXLAN over MLAG....2352
17.3.6 Configuring VXLAN Control Service....2353
17.3.7 Configuring VXLAN Multicast Decapsulation....2353
17.3.8 VXLAN Rules Support for Mirror ACLs Configuration....2354
17.3.9 Configuring EVPN VXLAN....2354
17.3.10 Displaying VXLAN Configuration....2359
17.3.11 Displaying VXLAN Bridging and Routing Support....2362
17.4 VXLAN Commands....2364
17.4.1 clear vxlan counters....2364
17.4.2 designated-forwarder election hold-time....2365
17.4.3 interface vxlan....2365
17.4.4 ip address virtual....2366
17.4.5 redistribute attached-host....2367
17.4.6 show arp....2368
17.4.7 show interfaces vxlan....2369
17.4.8 show service vxlan....2370
17.4.9 show vxlan address-table....2371
17.4.10 show vxlan counters....2372

17.4.11 show vxlan flood vtep....2373
17.4.12 show vxlan vtep....2373
17.4.13 vxlan flood vtep....2374
17.4.14 vxlan multicast-group decap....2375
17.4.15 vxlan source-interface....2376
17.4.16 vxlan udp-port....2377
17.4.17 vxlan vlan vni....2377
17.4.18 vxlan vni notation dotted....2378

Chapter 18: Ethernet VPN (EVPN)....2380
18.1 EVPN Overview....2380
18.1.1 EVPN Terminology....2381
18.1.2 EVPN Service Models....2384
18.1.3 VCS and EVPN in DCI....2386
18.1.4 EVPN MPLS LAYER 3 VPN (Type-5 Route)....2387
18.1.5 EVPN VxLAN IPV6 Overlay....2388
18.2 EVPN Layer 3 Core Operations....2388
18.2.1 MAC Address Learning....2388
18.2.2 ARP Suppression....2390
18.2.3 MAC Mobility....2390
18.2.4 MAC Address Damping....2391
18.2.5 Broadcast and Multicast Traffic....2391
18.3 Integrated Routing and Bridging....2393
18.3.1 IP VPN....2398
18.4 VPN MPLS Transport Options....2402
18.4.1 LDP....2405
18.4.2 ISIS-SR....2407
18.4.3 BGP-LU (BGP-SR)....2408
18.5 EVPN Type-5 Routes: IP Prefix Advertisement....2410
18.6 BGP PIC Edge for EVPN VXLAN Routes for Remote VTEP Failures....2412
18.6.1 Configuring BGP PIC Edge for EVPN VXLAN Routes for Remote VTEP Failures....2413
18.6.2 Show Commands....2414
18.6.3 Troubleshooting....2416
18.6.4 Limitations....2416
18.7 VXLAN DSCP Mapping....2416
18.8 EVPN IGP Cost for VTEP Reachability....2417
18.8.1 Configuration Example....2418
18.8.2 Show Commands....2419
18.9 EVPN VXLAN Single-Gateway Centralized Routing....2421
18.9.1 Configuration....2422
18.9.2 Show Commands....2425
18.9.3 Limitations....2428
18.9.4 Flood Traffic Filtering with EVPN....2428
18.10 Inter-VRF Local Route Leaking....2429
18.10.1 Inter-VRF Local Route Leaking using BGP VPN....2429
18.10.2 Inter-VRF Local Route Leaking using VRF-leak Agent....2433
18.11 Static Inter-VRF Route....2433
18.11.1 Configuration....2434
18.11.2 Show Commands....2434
18.11.3 Limitations....2434
18.12 VCS to EVPN Hitless Migration....2434
18.12.1 Configuration....2435
18.12.2 Show Commands....2438
18.12.3 Limitations....2441
18.13 Configuring EVPN....2441

18.13.1 Configuring BGP-EVPN and VCS on CVX....2441
18.13.2 EVPN MPLS Virtual Private Wire Service (VPWS)....2442
18.14 Sharing Equivalence Class entry across multiple VRF....2447
18.15 Sample Configurations....2448
18.15.1 EVPN VXLAN IRB Sample Configuration....2448
18.15.2 Multi-Tenant EVPN VXLAN IRB Sample Configuration....2453
18.15.3 EVPN MPLS Sample Configuration....2475
18.15.4 EVPN VxLAN IPv6 Overlay....2490
18.15.5 IP VPNs Sample Configuration....2496
18.16 EVPN and VCS Commands....2513
18.16.1 bfd vtep evpn....2514
18.16.2 encapsulation vxlan layer-3 set next-hop igp-cost....2515
18.16.3 leak routes....2515
18.16.4 next-hop resolution disabled....2516
18.16.5 redistribute bgp evpn vxlan....2516
18.16.6 redistribute router-mac next-hop vtep primary....2517
18.16.7 redistribute service vxlan....2517
18.16.8 route-target....2518
18.16.9 route-target export....2518
18.16.10 route-target import....2519
18.16.11 route-target route-map....2520
18.16.12 router general....2521
18.16.13 show bgp evpn....2521
18.16.14 show ip bgp vrf....2523
18.16.15 show ip route vrf....2523
18.16.16 show ipv6 bgp vrf....2524
18.16.17 show ipv6 route vrf....2525
18.16.18 show l2rib input all....2525
18.16.19 show l2Rib input vxlan-control-service....2526
18.16.20 show l2rib output....2526
18.16.21 show service vxlan address-table....2527
18.16.22 show vrf leak flapping....2528
18.16.23 show vxlan control-plane....2529
18.16.24 vlan (VLAN-AWARE-Bundle configuration mode)....2529
18.16.25 vni-aware-bundle....2530

Chapter 19: Multiprotocol Label Switching (MPLS)....2531
19.1 MPLS....2531
19.1.1 MPLS Description....2531
19.1.2 MPLS Configuration....2533
19.2 BGP/MPLS L3 VPN....2544
19.2.1 Operation....2544
19.2.2 Configuration....2546
19.2.3 Show Commands....2552
19.2.4 Limitations....2556
19.3 EVPN MPLS Shared ESI Label....2556
19.3.1 Configuring EVPN MPLS Shared ESI Label....2556
19.3.2 EVPN MPLS Shared ESI Label Show Commands....2557
19.3.3 Limitations....2558
19.4 RSVP-TE LSR....2558
19.4.1 RSVP-TE LSR Configuration....2558
19.5 RSVP-TE LER....2576
19.6 LDP Pseudowire....2582
19.6.1 Configuring LDP Pseudowire....2582
19.6.2 LDP Pseudowire Limitations....2584

19.6.3 LDP Pseudowire Show Commands..... 2584
19.7 LDP Entropy Label..... 2585
19.7.1 Configuring LDP Entropy Label..... 2585
19.7.2 Show Commands..... 2586
19.7.3 Limitations..... 2587
19.8 MPLS Commands..... 2587
19.8.1 authentication..... 2587
19.8.2 cspf delay..... 2588
19.8.3 entropy-label..... 2589
19.8.4 fast-reroute..... 2589
19.8.5 fast-reroute reversion..... 2590
19.8.6 hello interval..... 2591
19.8.7 mpls ip..... 2591
19.8.8 mpls shared index..... 2592
19.8.9 mpls static..... 2593
19.8.10 mpls static vrf-label..... 2595
19.8.11 mpls tunnel termination..... 2595
19.8.12 mpls tunnel termination (vrf qos map)..... 2596
19.8.13 ping mpls rsvp session..... 2596
19.8.14 preemption method..... 2598
19.8.15 refresh method..... 2599
19.8.16 show mpls route summary..... 2599
19.8.17 show mpls route..... 2600
19.8.18 show mpls rsvp..... 2601
19.8.19 show mpls rsvp counters..... 2602
19.8.20 show mpls rsvp neighbor..... 2603
19.8.21 show mpls rsvp session detail..... 2603
19.8.22 show mpls rsvp session summary..... 2604
19.8.23 show mpls rsvp session..... 2604
19.8.24 show mpls tunnel termination qos maps..... 2605
19.8.25 show traffic-engineering cspf path..... 2605
19.8.26 show traffic-engineering database..... 2606
19.8.27 shutdown..... 2608
19.8.28 srlg..... 2609
19.8.29 vrf (MPLS tunnel termination)..... 2609

Chapter 20: Visibility and Monitoring Services..... 2611
20.1 Test Access Point Aggregation..... 2611
20.1.1 TAP Aggregation Introduction..... 2611
20.1.2 TAP Aggregation Description..... 2611
20.1.3 TAP Aggregation Extra MPLS Pop (4 to 6 Labels)..... 2614
20.1.4 TAP Aggregation Configuration..... 2616
20.1.5 TAP Aggregation Traffic Steering..... 2629
20.1.6 TAP Aggregation GUI..... 2638
20.1.7 TAP Aggregation Keyframe and Timestamp Configuration..... 2640
20.1.8 TapAgg GRE Tunnel Termination..... 2642
20.1.9 Tap Aggregation Hardware Forwarding Profile..... 2648
20.1.10 TAP Aggregation MPLS Pop..... 2649
20.1.11 TAP Aggregation 802.1br EVN Tag Stripping..... 2650
20.1.12 TAP Aggregation Commands..... 2652
20.2 Latency Analyzer (LANZ)..... 2683
20.2.1 Introduction to LANZ..... 2684
20.2.2 LANZ Overview..... 2684
20.2.3 Configuring LANZ..... 2685
20.2.4 LANZ Commands..... 2694

20.3 Sampled Flow Tracking....2718
20.3.1 Sampled Flow Tracking Overview....2718
20.3.2 Configuring Sampled Flow Tracking....2719
20.3.3 Hardware Flow Tracking with IPFIX Export....2721
20.3.4 Postcard Telemetry....2725
20.3.5 Inband Network Telemetry (INT) Support....2729
20.3.6 Sampled Flow Tracking Configuration Examples....2731
20.3.7 Sampled Flow Tracking Commands....2732
20.4 sFlow....2743
20.4.1 sFlow Conceptual Overview....2744
20.4.2 sFlow Configuration Procedures....2749
20.4.3 sFlow Subinterfaces....2752
20.4.4 QinQ L3 Subinterfaces....2752
20.4.5 sFlow Commands....2754
20.5 SNMP....2763
20.5.1 SNMP Introduction....2764
20.5.2 SNMP Conceptual Overview....2764
20.5.3 Configuring SNMP....2765
20.5.4 SNMP Commands....2774
20.6 VM Tracer....2793
20.6.1 VM Tracer Introduction....2793
20.6.2 VM Tracer Description....2793
20.6.3 VM Tracer Configuration Procedures....2795
20.6.4 VM Tracer Commands....2799
20.7 MapReduce Tracer....2816
20.7.1 MapReduce Tracer Introduction....2816
20.7.2 MapReduce Tracer Configuration....2817
20.7.3 Displaying MapReduce Tracer Results....2822
20.7.4 MapReduce Tracer Commands....2828
20.8 Transceiver Performance Monitoring....2858
20.8.1 Transceiver Performance Monitoring Overview....2858
20.8.2 Configuring Transceiver Performance Monitoring....2858
20.8.3 Transceiver Performance Monitoring Limitations....2858
20.8.4 Transceiver Performance Monitoring Show Commands....2859
20.8.5 Transceiver Performance Monitoring Commands....2862
20.9 AVA Sensor....2868
20.9.1 NDR Sensor Extension Installation....2868
20.9.2 Configuration....2869
20.9.3 Upgrade EOS and/or NDRSensor.swix Extension....2872
20.9.4 Show Commands....2873
20.9.5 Limitations....2875
20.9.6 AVA Sensor Commands....2875

Chapter 1

Overview

Arista Networks features switches with high-density, non-blocking Ethernet ports that are controlled through an extensible, Linux-based, modular network operating system. The intended audience for this manual is network administrators who configure Arista switches. A working knowledge of network administration is assumed.

New Features

This guide may not describe the features added in the most recent EOS version. For information on undocumented features, consult the TOI documents here: https://www.arista.com/en/support/toi.

Switch Platforms

A list of Arista switches and detailed information about each is available online here: https://www.arista.com/en/products.

Recently released switches may not appear in the list, but can be found in the most recent Release Notes (found under Active Releases here: https://www.arista.com/en/support/software-download).

Open-Source Licenses

Portions of Arista Networks software are covered by open-source licenses including the GNU General Public License (GNU GPL).

See https://www.arista.com/en/support/product-documentation/gpl.

CloudVision

The configuration tasks required for deployment of CloudVision, CVX, and CVP are described in the CloudVision Configuration Guide. The latest version can be found here: https://www.arista.com/en/cg-cv/.

Supported Features

For the complete supported-features list in the latest EOS release, see https://www.arista.com/en/support/product-documentation/supported-features.

For details on a specific release, see the Release Notes (found under Active Releases here: https://www.arista.com/en/support/software-download).

This section contains the following topic:

Command-Line Interface (CLI)

1.1 Command-Line Interface (CLI)

The command-line interface (CLI) is one tool for controlling the switch and displaying information about its status and configuration. This chapter describes the use of the CLI.

This chapter includes these sections:

- Accessing the EOS CLI
- Processing Commands
- Kernel-based Virtual Machine Commands and Configuration

- Switch Platforms
- Command Modes
- Managing Switch Configuration Settings
- Other Command-Line Interfaces
- Directory Structure
- Command-Line Interface Commands

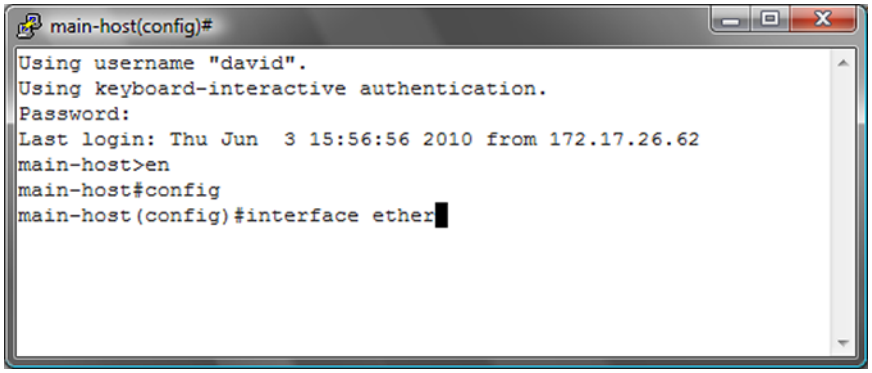
1.1.1 Accessing the EOS CLI

You can open an EOS CLI session through these connections:

- Ethernet management ports
- console port
- Telnet connections
- Secure Shell (SSH)

The following figure displays the EOS CLI in a Secure Shell connection.

Figure 1-1: EOS Command-Line Interface



1.1.2 Processing Commands

1.1.2.1 Command Execution

Command keywords are not case-sensitive. The CLI also accepts truncated keywords that uniquely correspond to one command.

- The command abbreviation **con** does not execute a command in Privileged EXEC mode because the names of two commands begin with these letters: **configure** and **connect**.

```
switch# con
% Ambiguous command
```

- The command abbreviation **conf** executes **configure** in Privileged EXEC mode because no other command name begins with **conf**.

```
switch# conf
switch(config)#
```

1.1.2.2 Creating an Alias

The alias command creates an alias for a CLI command. Entering the alias in the CLI executes the corresponding command.

Example

- This command makes **srie** an alias for the command **`show running-config interface ethernet 1-5`**.

```
switch(config)# alias srie show running-config interface ethernet 1-5
switch(config)# srie
interface Ethernet1
   switchport access vlan 33
   storm-control broadcast level 1
   spanning-tree portfast
   spanning-tree bpduguard enable
interface Ethernet2
   switchport access vlan 33
   spanning-tree portfast
interface Ethernet3
   switchport access vlan 33
   spanning-tree portfast
   spanning-tree bpduguard enable
interface Ethernet4
interface Ethernet5
   shutdown
```

1.1.2.3 Cursor Movement Keystrokes

EOS supports these cursor movement keystrokes:

- **Ctrl-B** or the **Left Arrow** key: moves cursor to the left.
- **Ctrl-F** or the **Right Arrow** key: moves cursor to the right.
- **Ctrl-A**: moves cursor to beginning of line.
- **Ctrl-E**: moves cursor to end of line.
- **Esc-B**: moves cursor left one word.
- **Esc-F**: moves cursor right one word.

1.1.2.4 History Substitution Keystrokes

The history buffer retains the last 20 commands entered. History substitution keystrokes that access previously entered commands include:

- **Ctrl-P** or the **Up Arrow** key: Recalls the most recent buffered commands. Repeat to recall older commands.
- **Ctrl-N** or the **Down Arrow** key: Recalls more recent commands after using the **Ctrl-P** or the **Up Arrow**. Repeat to recall newer commands.

The **`show history`** command in Privileged EXEC mode displays the history buffer contents.

```
switch#show history
  en
  config
  exit
  show history
```

1.1.2.5 Command Lists and Syntax Assistance

EOS CLI uses widely followed conventions for providing command lists and syntax assistance. These conventions are available in all command modes.

- To display all commands available at this level, type a question mark (**?**):

```
switchName>?
  clear       Reset functions
```

```
  connect     Open a terminal connection
  default     Set a command to its defaults
  disable     Turn off privileged commands
  enable      Turn on privileged commands
  exit        Exit from the EXEC
  logout      Exit from the EXEC
  no          Negate a command or set its defaults
  ping        Send echo messages
  show        Show running system information
  ssh         Open ssh connection
  tcpdump     Monitor packets with tcpdump
  telnet      Open a telnet connection
  terminal    Configure the terminal
  traceroute  Trace route to destination
  watch       Execute a command repeatedly
  who         Display information about terminal lines
  zerotouch   ZeroTouch configuration
```

- To display a list of commands beginning with a specific character sequence, type the sequence followed by a question mark.

```
switch# di?
diff  dir  disable
```

- To display a command's keywords or arguments, type a question mark as an argument.

```
switch> ping ?
WORD  Ping destination address or hostname
ip    IPv4 echo
ipv6  IPv6 echo
mpls  Send echo messages for LSP
vrf   Ping in a VRF
```

- The switch accepts an address-mask or CIDR notation (address-prefix) in commands that require an IP address and mask. For example, these commands are processed identically:

```
switch(config)# ip route 0.0.0.0 255.255.255.255 10.1.1.254
switch(config)# ip route 0.0.0.0/32 10.1.1.254
```

- The switch accepts an address-wildcard or CIDR notation in commands requiring an IP address and wildcard. Wildcards use zeros to mask portions of the IP address and are found in some protocol configuration statements, including OSPF. The switch processes these commands identically:

```
switch(config-router-ospf)# network 10.255.255.1 0.0.0.255 area 15
switch(config-router-ospf)# network 10.255.255.1/24 area 15
```

1.1.2.6 Regular Expressions

A regular expression is a search pattern composed of symbols, letters and numbers. Some CLI parameters are defined as regular expressions for specifying more expressive search criteria. The switch uses regular expression pattern matching in several BGP commands.

The functionality of a regular expression for an AS-Path varies based on BGP regex asn and string mode configurations in the **`ip as-path regex-mode`** command.

The following tables describe the behavior of special characters in asn and string modes respectively.

Special Characters	Characters Names	Description	Examples
.	Period	Matches any AS number.	'.' matches '200'. '10.20' matches '10 30 20', but does not match '10 20'.
^	Caret	Matches the specified expression at the beginning of an input string. Also used to exclude expressions in brackets while matching.	'^123' matches '123', '123 456', '123 456 789', and so on; but does not match '1234'. '[^12]' matches '1', '2', '3', and so on; but does not match '12'.
*	Asterisk	Matches an entire AS number that appears either zero or more times.	'200_100*_300' matches '200 300', '200 100 300', '200 100 100 300', and so on. '^100*$' matches empty AS path, '100', '100 100', '100 100 100', and so on.
+	Plus sign	Matches an entire AS number appearing either one or more times.	'10_20+_30' matches '10 20 30', '10 20 20 30' and so on; but does not match '10 200 30'.
$	Dollar sign	Matches the specified expression at the end of an input string.	'1_2_3$' matches '1 2 3', but does not match '1 2 3 4'.
[]	Brackets	Matches either an AS number, or a range of AS numbers separated by a hyphen.	'[10_20_30-39]' matches '10', '20', '30', '31',...'39'.
?	Question mark	Matches either zero or one occurrence of the pattern but the previous operand or entire AS number may appear zero or one time.	'100_200?' matches '100' and '100 200'. '100_200?$' does not match with '100 20'.
|	Pipe	Matches the specified AS number on either side of the vertical bar.	'6400|6500' matches either '6400' or '6500'.
( )	Parenthesis	Nests specified AS numbers for matching.	'^(100(200|300))$' matches either '100 200' or '100 300'. '^100_200\|300_400$' matches AS path either "100 200" or "300 400".
_	Underscore	Matches specified AS numbers that are converted into AS number delimiters.	'_123_456_' matches '123 456'. '_333_444_' matches '111 222 {333 444}'.

Note: Precede the question mark (**?**) with **Ctrl+V** sequence to prevent it from being interpreted as a help command.

Special Characters	Characters Names	Description	Examples
.	Period	Matches any single character.	‘1.2’ matches ‘102’.
^	Caret	Matches the specified expression at the beginning of an input string.	‘^123’ matches ‘123’, ‘1234’, ‘12345’, and so on. It also matches ‘123 456’, ‘123 456 789’, and so on.
*	Asterisk	Matches either zero or more sequences of the expression preceding the asterisk.	'^5*$' matches an empty AS path, '5', '55', '555', and so on.
+	Plus sign	Matches either one or more sequences of the expression preceding the plus sign.	‘5+’ matches to ‘5’, ‘55’, ‘555’, and so on.
$	Dollar sign	Matches the expression at the end of an input string.	‘123$’ matches ‘123’, but does not match ‘1234’.
[]	Brackets	Matches either characters or a range of characters separated by a hyphen, within left and right brackets.	‘[025-7]’ matches ‘0’, ‘2’, and digits from ‘5’ to ‘7’; but does not match digits from ‘1’, ‘3’, ‘4’, ‘8’, and ‘9’.
?	Question mark	Matches either zero or one occurrence of the pattern.	'12?3' matches '13' and '123'.
|	Pipe	Matches either one of the expressions or expression patterns on either side of the vertical bar.	‘14(36|75)12’ matches either ‘143612’ or ‘147512’; but does not match ‘1412’, ‘14367512’, ‘14363612’ or ‘14757512’.
( )	Parenthesis	Nests specified expressions for matching.	‘(17)*’ matches any number of the two-character string ‘17’.
_	Underscore	For AS-Path regex, '_' matches curly brackets '{}', the beginning of input string, the end of input string, or space.	‘_1300_’ matches ‘100 {1300 1400}’, ‘100 1300 200’, and so on.
{}	Braces	Matches repetitions of the previous expression with the number of repetitions provided in braces.	'10{2,3}' matches '100' and '1000'.
\	Backslash	Matches the character following the backslash and special characters.	‘\(42’ matches ‘(42’.

Precede the question mark (**?**) with **Ctrl+V** sequence to prevent it from being interpreted as a help command.

1.1.2.7 Scheduling CLI Commands

The **schedule** command facilitates the periodic execution of the specified CLI command. Command parameters configure the time to start script execution, the interval between consecutive execution instances, the maximum time to execute the script, and the maximum number of files log that needs to be created.

The **schedule config** command sets configuration parameters to the CLI scheduler.

The **show schedule** command lists the commands currently scheduled for periodic execution and displays the summary of the specified scheduled command.

Examples

- This command schedules the execution of a script once every ***12*** hours and the script execution is terminated if it exceeds ***40*** minutes. When max-log-files is set to zero, the script output is not logged.

```
switch# schedule ms_1 interval 720 timeout 40 max-log-files 0 command bash /
mnt/flash/myscript.sh
```

- This command saves the running configuration contents to a log file every hour, terminates the script execution if it exceeds 30 minutes and creates up to ***24*** log files.

```
switch#schedule backup-test interval 60 max-log-files 24 command show
 running-config
```

- This command allows the switch to concurrently execute up to ***2*** scheduled commands.

```
switch(config)#schedule config max-concurrent-jobs 2
switch(config)#
```

- This command lists the commands that are scheduled for periodic execution.

```
switch(config)#schedule config max-concurrent-jobs 3
switch(config)#show schedule summary
Maximum concurrent jobs  3
Prepend host name to logfile: No
Name            At time      Last     Interval   Timeout   Max     Logfile Location               Status
                             time      (mins)     (mins)    log
                                                           files
------------- ------------- ------- ---------- --------- -------- ------------------------------ ------
tech-support       now        00:29       60         30      100     flash:schedule/tech-support/   Success
thelp           12:02:00      00:02       60         40      100     flash:schedule/thelp/          Fail
                06/05/2018
switch(config)#
```

1.1.2.8 Running Bash Shell Commands Automatically with Event Handlers

Event handlers execute a Linux Bash shell command in response to a specific system event. An event handler consists of a Bash command, a trigger and a delay; when the trigger event occurs, the action is scheduled to run after ***delay*** seconds.

To create an event handler, use the **event-handler** command. This creates a new event handler and places the CLI in event handler configuration mode for that handler. Use the **action bash** command to configure a Bash command to run when the handler is triggered, and the **trigger** command to specify the trigger. Event handlers can be triggered by various events, including:

- system booting
- a change in a specified interface's operational status or IP address
- a change in the ***startup-config*** file
- a state change in a virtual machine monitored by VM Tracer

To change the delay period between the trigger and the action, use the delay command.

When an action is run, certain information is passed to it through environment variables. For the boot trigger, no variables are set. For the **interface** triggers, the following variables are set and passed to the action:

- ***$INTF*** interface name.
- ***$OPERSTATE*** current operational status of the specified interface.
- ***$IP-PRIMARY*** current primary IP address of the specified interface.

To execute more than one Bash command in response to a trigger, create a script containing the desired commands and enter the file path to the script as the argument of the action bash command.

To display information about all event handlers or about a specific event handler, use the **show event-handler** command.

The **no event-handler** command deletes an event handler.

Examples

- These commands create an event handler named ***eth_4*** which sends an email to a specified address when there is a change in the operational status of ***Ethernet interface 4***:

```
switch(config)# event-handler eth_4
switch(config-event-eth_4)# action bash email x@yz.com -s "Et4 $OPERSTATE"
switch(config-event-eth_4)# trigger on-intf ethernet 4 operstatus
switch(config-event-eth_4)# delay 60
switch(config-event-eth_4)# exit
switch(config)#
```

  The above handler uses the ***$OPERSTATE*** variable to include the current operational state (“linkup” or “linkdown”) in the subject of the email. Note that the action will only function if email has been configured on the switch.
- These commands create an event handler named ***onStartup*** which executes a user-defined script 60 seconds after the system boots.

```
switch(config)# event-handler onStartup
switch(config-event-onStartup)# action bash /mnt/flash/startupScript1
switch(config-event-onStartup)# trigger onboot
switch(config-event-onStartup)# delay 60
switch(config-event-onStartup)# exit
switch(config)#
```

  The above handler will also be executed on exiting from event-handler configuration mode.
- This command displays information about all event handlers configured on the system.

```
switch# show event-handler
Event-handler onStartup
Trigger: onBoot delay 60 seconds
Action: /mnt/flash/startupScript1
Last Trigger Activation Time: 1 minutes 51 seconds ago
Total Trigger Activations: 1
Last Action Time: 51 seconds ago
Total Actions: 1 switch#
```

- This command deletes the event handler named ***onStartup***.

```
switch(config)# no event-handler onStartup
switch(config)#
```

1.1.2.9 Running Adverse Drop Counters Monitor with Event Handlers

A monitoring capability for adverse drop counters can be used as a warning that the system is encountering an abnormal condition. The adverse drop counter monitor runs periodically (with a default of 60 seconds) and performs the following actions:

- Reads the values of adverse drop counters.
- Compares each value to the value read in the previous run.
- If counter values increase more than a certain threshold (with a default of 100), it is considered as a threshold violation.
- If any counter has more than a certain number of threshold violations within a specific time window (with a default of 3 violations within 15 minutes) a syslog message is logged.

No configuration is required to enable adverse drop counters monitor with event handlers. It is enabled by default and can be disabled, and can be customized for duration of time window and threshold levels. To customize the delay, polling interval, and condition for width, violation count, and threshold of this event handler, use the **event-handler DropCountersHandler** command. To display details of this event handler, use the **show event-handler DropCountersHandler** command.

Examples

- These commands customize the delay, polling interval, and condition for width, violation count, and threshold of this event handler. Each parameter may be customized separately, with all other parameters remaining unchanged.

```
switch(config)# event-handler DropCountersHandler
switch(config-DropCountersHandler)# action bash DropCounterLog.py -l
switch(config-DropCountersHandler)# delay 0
switch(config-DropCountersHandler)# trigger on-counters
switch(config-DropCountersHandler-counters)# poll interval 60
switch(config-DropCountersHandler-counters)# condition
bashCmd."DropCounterMonitor.py" -w 800" > 0
switch(config-DropCountersHandler-counters)# condition
bashCmd."DropCounterMonitor.py" -c 5" > 0
switch(config-DropCountersHandler-counters)# condition
bashCmd."DropCounterMonitor.py" -t 200" > 0
switch(config-DropCountersHandler-counters)#
```

- This command disables this event handler.

```
switch(config)# no event-handler DropCountersHandler
switch(config)#
```

- This command displays details of this event-handler.

```
switch(config)# show event-handler DropCountersHandler
Event-handler DropCountersHandler (BUILT-IN)
Trigger: on-counters delay 0 seconds
  Polling Interval: 60 seconds
  Condition: bashCmd."DropCounterMonitor.py" > 0
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: DropCounterLog.py -l
Action expected to finish in less than 20 seconds
Total Polls: 39
Last Trigger Detection Time: 38 minutes 22 seconds ago
Total Trigger Detections: 1
Last Trigger Activation Time: 38 minutes 22 seconds ago
Total Trigger Activations: 1
Last Action Time: Never
Total Actions: 1

switch(config)#
```

1.1.3 Kernel-based Virtual Machine Commands and Configuration

Arista's EOS has leveraged its unmodified Linux kernel, and embraced open source standards-based technology that has brought operating system virtualization to Ethernet switching, utilizing the kernel-based virtual machine (KVM) as follows:

- The hypervisor is the Linux kernel.
- The core virtualization infrastructure is provided by the kernel module.
- The CPU-specific implementation is provided by the processor-specific module (Intel or AMD).

- The generic machine emulator and virtualizer KVM is provided by a Modified Quick Emulator (QEMU), which transforms the Linux kernel into the hypervisor.

The standard Linux kernel is the hypervisor, resulting in changes to the standard kernel (such as memory support and scheduler). Optimizations to these Linux components (such as a new scheduler in the 2.6 kernel) benefit both the hypervisor (host operating system) and Linux guest operating systems. With the kernel acting as the hypervisor, the switch can run other operating systems, such as Windows or Linux.

All components required are pre-installed with the Arista EOS software image, requiring only the download of the image. A few additional configuration steps get the KVM fully operational.

This chapter contains the following sections:

- KVM Commands
- KVM Configuration

1.1.3.1 KVM Commands

The following table covers KVM commands used throughout the configuration.

Table 1: KVM Commands

Command	Description
comment	Up to 240 character comment for this mode.
default	Set a command to its defaults.
disk-image	Add Virtual Machine disk image.
enable	Enable VM.
exit	Exit from Virtual Machine configuration mode.
memory-size	Set memory size.
no	Negate a command or set its defaults.
show	Show running system information.
virtual-nic	Add virtual NIC.
vnc-port	Set VNC server port.
! !	Append to comment.

1.1.3.1.1 CLI Commands

The following KVN CLI commands are used throughout the configuration.

vm

In **`config`** mode, the `vm` CLI command creates or deletes a KVM configuration, or enters **`config-vm`** mode. A newly created KVM will have an empty config file path and is disabled.

The CLI command syntax is as follows:

```
[no] vm NAME
```

config-file

In **`config-vm`** mode, the **`config-file`** CLI command sets the path of the **`libvirt`** config file, using standard file syntax (e.g. **`flash:vm/NetscalerVPX.xml`** or **`sata1:vm/NetscalerVPX.xml`** or **`/mnt/`**

sata1/vm/NetscalerVPX.xml). Changing this value does not affect the state of a currently enabled KVM. To use the new file, the user must disable and then re-enable the KVM.

The CLI command syntax is as follows:

```
config-file [PATH]
```

enabled

In **config-vm** mode, the **enabled** CLI command allows enabling a currently disabled VM, causing it to start up immediately. If a VM is enabled in the **startup-config**, it starts up automatically when EOS boots (or when **VirtAgent** starts).

The CLI command syntax is as follows:

```
[no] enabled
```

Disabling a currently enabled VM initiates a shutdown process in the following sequence:

- Attempt to shut down the VM politely if the guest OS supports ACPI.
- If the VM is still running after 30 seconds, terminate it.

show vm

In **enable** mode, the **show vm** CLI command prints information about the configuration and status of a KVM, or of all KVMs if **NAME** is omitted, as follows:

- Configuration: Name, config file path, and enabled.
- Status: PID, log file path, and serial console pty path.
- Current resource usage: RES, CPU%.
- (Detailed only) contents of the log file.

(Detailed only) contents of the config file.

The CLI command syntax is as follows:

```
show vm [detailed] [NAME]
```

attach vm

In **enable** mode, the **attach vm** CLI command connects to a KVM's serial console pty (using virsh console).



Note: Press **Ctrl-]** to exit to the CLI.

The CLI command syntax is as follows:

```
attach vm [NAME]
```

show tech-support

The CLI command syntax is as follows:

```
show tech-support [detailed] [NAME]
```

reload

In **enable** mode, the **reload** CLI command is executed before restarting the system, and will shut down currently enabled KVMs using the same process as the **no enabled** command in **config-vm** mode.

The CLI command syntax is as follows:

```
reload
```

1.1.3.2 KVM Configuration

Arista EOS enables kernel-based virtual machine (KVM) instances by running KVM on the control-plane CPU of the switch. KVM instances can be defined from the CLI.

To configure a KVM, you must download the virtual machine image and configure the EOS.

This section contains the following topics:

- Configuring a KVM
- Configuring a Guest KVM

1.1.3.2.1 Configuring a KVM

To configure a KVM, perform the following steps:

1. Download the Virtual Machine Image to **`/mnt/flash`**
2. Name the virtual machine: **`switch(config)# virtual-machine [kvm_name]`**

   Example:

   ```
   switch(config)# virtual-machine foo
   ```

3. Provide a pointer to the image: **`switch(config-vm-foo)# disk-image [file:[path] image-format [format]`**

   Example:

   ```
   switch(config-vm-foo)# disk-image file:/mnt/flash/fedora.img image-format
    qcow2
   ```

4. Define the amount of memory allocated: **`switch(config-vm-foo)#memory-size [size in bytes]`**
5. Bind the virtual NIC to an SVI (or management interface): **`switch(config-vm-foo)#virtual-nic 1 vlan [1-4] switch(config-vm-foo)# virtual-nic 1 management [1-4]`**
6. Create the VNC server's tcp port (display): **`switch(config-vm-foo)# vnc-port [vnc-port number]`**
7. Enable the virtual machine: **`switch(config-vm-foo)# enable`**

   Optionally attach to the virtual machine via VNC client pointed to the switch's IP address. However, if Kernel **`hair-pinning`** is currently not enabled, preventing communication directly with the local switch, all traffic must have a destination on another networked device (such as a router, switch, or server).

For specifics about KVM visit http://www.linux-kvm.org/.



Note: In the Real VNC Viewer for **`Options`**, **`Expert`**, and **`ColorLevel`**, if the default value is ***pal8***, establishing a session may fail. If this occurs, set this value to **full** and **reconnect**.

Example

```
switch#copy http://berrange.fedorapeople.org/images/2012-02-
29/f16-x86_64-openstack-sda.qcow2
(http://berrange.fedorapeople.org/images/2012-02-29/f16-x86_64-
openstack-sda.qcow2) flash:
...
switch(config)# virtual-machine foo
switch(config-vm-foo)# disk-image file:/mnt/flash/fedora.img image-format qcow2
switch(config-vm-foo)# memory-size 512
switch(config-vm-foo)# virtual-nic 1 vlan 1
switch(config-vm-foo)# virtual-nic 2 management 1
switch(config-vm-foo)# vnc-port 5900
switch(config-vm-foo)# enable
```

1.1.3.2.2 Configuring a Guest KVM

To configure a guest KVM, perform the following steps:

1. Download the Virtual Machine Image to **/mnt/flash**
2. Name the virtual machine: **switch(config)# virtual-machine [guest_name]**

   Example

```
switch(config)# virtual-machine guest123
```

3. Provide a pointer to the image: **switch(config-vm-guest123)# disk-image [file:[path] image-format [format]**

   Example

```
switch(config-vm-guest123)# disk-image flash:f16-x86_64-openstack-sda.qcow2
 image-format ?
iso iso image format   qcow qcow image format   qcow2 qcow2 image format
 raw raw image format
vmdk vmdk image format
switch(config-vm-guest123)# disk-image
flash:f16-x86_64-openstack-sda.qcow2 image-format qcow2
```

4. Define the amount of memory allocated: **switch(config-vm-guest123)#memory-size [size in bytes]**
5. Bind the virtual NIC to an SVI (or management interface): **switch(config-vm-guest123)# virtual-nic 1 vlan [1-4] switch(config-vm-guest123)# virtual-nic 2 management [1-4]**
6. Create the VNC server's tcp port (display): **switch(config-vm-guest123)# vnc-port [vnc-port number]**
7. Enable the virtual machine: **switch(config-vm-guest123)# enable**

   Example

```
switch#copy http://berrange.fedorapeople.org/images/2012-02-
29/f16-x86_64-openstack-sda.qcow2
(http://berrange.fedorapeople.org/images/2012-02-29/f16-x86_64-
openstack-sda.qcow2) flash:
...
switch# config terminal
switch(config)# virtual-machine ?
    WORD Virtual Machine name
switch(config)# virtual-machine foo
switch(config-vm-foo)# disk-image flash:f16-x86_64-openstack-sda.qcow2
 image-format ?
    iso           iso image format
    qcow          qcow image format
    qcow2         qcow2 image format
    raw             raw image format
    vmdk          vmdk image format
switch(config-vm-foo)# disk-image flash:f16-x86_64-openstack-sda.qcow2
 image-format qcow2
switch(config-vm-foo)# memory-size 1024
switch(config-vm-foo)# virtual-nic ?
    <1-4>         Virtual NIC Id
switch(config-vm-foo)# virtual-nic 1 ?
    Management    Management interface
    Vlan          Vlan interface
switch(config-vm-foo)# virtual-nic 1 vlan 1
switch(config-vm-foo)# virtual-nic 2 management 1
switch(config-vm-foo)# enable
```

```
switch(config-vm-foo)#
switch(config-vm-foo)# ^Z
switch# write mem
switch#
switch# show virtual-machine detail
Virtual Machine: foo
    Enabled:           Yes
    State:             Running
    Disk Image:        /mnt/flash/f16-x86_64-openstack-sda.qcow2
    Disk Image Format: qcow2
    Memory Size: 1024MB
    VNC port: 5900
    Virtual Nic: vnic1
        Mac Address:   52:54:00:ee:11:c9
        Device:        Vlan1
        Model Type:    e1000
    Virtual Nic:       vnic2
        Mac Address:   52:54:00:df:2a:e1
        Device:        Management1
        Model Type:    e1000 switch#
```

1.1.4 Switch Platforms

Features and CLI commands vary by switch platform. CLI options may also vary by switch platform for commands that are available on all platforms. Command descriptions in this manual describe feature availability and command parameters on the basis of switch platform, noting exceptions that exist among models that use a common platform.

- https://www.arista.com/en/products/switches lists the Arista switches and platforms upon which they operate.
- https://www.arista.com/en/support/product-documentation/supported-features lists Arista switch feature availability by switch platform. For the latest features, also consult the Release Notes, available at https://www.arista.com/en/support/software-download.

These sections describe the following topics:

- Viewing the Model Number
- Modular System Platforms – 7500 and 7500E Series Switches
- Viewing Modules on 7300 Series Modular Switches
- Multi-Chip Devices

1.1.4.1 Viewing the Model Number

To view the switch's model number through the CLI, enter the **`show version`** command.

Example

This command displays the model number, serial number, system MAC address, and manufacturing information of a DCS-7150S-64 switch.

```
switch> show version
Arista DCS-7150S-64-CL-F
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.13.2F
Architecture:           i386
Internal build version: 4.13.2F-1649184.4132F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1
```

```
Uptime:                 16 hours and 39 minutes
Total memory:           4017088 kB
Free memory:            1348228 kB switch>
```

1.1.4.2 Modular System Platforms – 7500 and 7500E Series Switches

Modular switch platforms depend on their installed modules along with the fabric and forwarding software modes. The **show module** command displays the fabric modules in the switch. System performance in switches containing both module types is based on first-generation fabric capabilities. Best practice is to avoid switch configurations with mixed fabric modules.

These sections describe modular switch components and software modes that program their capacities.

1.1.4.2.1 Fabric Modules and Fabric Mode – 7500 and 7500E Series Switches

Each modular switch fabric module is categorized as first-generation or E-Series:

- First-generation fabric modules support all basic switch functions.
- E-Series fabric modules support faster fabric link speeds, greater internal table capacities, and advanced encoding formatting.

Fabric mode determines the switch's fabric performance capabilities. This mode must match the fabric modules in the switch. Fabric mode settings include:

- **fe600**: Supports first-generation fabric modules.
- **fe1600**: Supports E-Series fabric modules.

E-series fabric modules can operate in **fe600** mode, but are limited to first-generation fabric performance. First-generation modules cannot operate in **fe1600** mode. Switches containing both types of modules must be set to **fe600** mode. Best practice is to avoid switch configurations with mixed fabric modules.

When a switch reloads, fabric mode is determined by the following (in order of precedence):

1. Switches reloading in **petraA** forwarding compatibility mode (Linecard Modules and Forwarding Compatibility Mode – 7500 and 7500E Series) also reload in **fe600** fabric mode.
2. As specified by the platform sand fabric mode (7500 and 7500E Series) statement in ***running-config***.
3. The first fabric module that becomes operational as the switch reloads.

In switches with a homogeneous module set, the fabric mode matches its fabric modules. Switches with a mixed set of modules are typically reloaded in **fe600** mode because first generation modules are usually operational before E-Series modules. However, the fabric mode in mixed module switches that are reloading cannot be guaranteed in the absence of the first two conditions.

Example

This command configures the switch to reload in **fe1600** fabric mode to support E-series fabric modules. After issuing this command, the switch should be reset only after exchanging all switch fabric modules to E-series modules.

```
switch(config)#platform sand fabric mode fe1600
switch(config)#exit
switch#show platform sand compatibility
                    Configuration        Status
Forwarding mode     None                 Arad
Fabric mode         Fe1600               Fe600 switch#
```

1.1.4.2.2 Determining a Switch's Operating Platform

FM6000 Platforms

To determine the operating platform on switch, display **`platform`** command options from Global Configuration command mode.

This command displays the operating platform of a switch operating on the FM6000 platform (7150 Series switches).

```
switch(config)# platform ?
  fm6000  FM6000 chip

switch(config)#platform
```

Arad and Petra Platforms

The **platform ? command** displays the same options on Arad and Petra platform switches. Refer to Viewing the Model Number to determine the switch's model number.

- Fixed system switches (DCS-7048 Series) operate on the Petra platform.
- Modular switches (DCS-7500 Series) operate on Arad and Petra platforms. Modular System Platforms – 7500 and 7500E Series Switches describe platform usage on these switches.

Arad and Petra platform switch typically utilize multiple chips. Multi-Chip Devices describe methods of determining the port distribution on multi-chip platforms.

Example

These commands display platform options of a switch operating on either Petra or Arad platforms.

```
switch(config)# platform ?
  arad    Arad switch chip
  fe1600  Fe1600 chip
  fe600   Fe600 fabric chip
  petraA  PetraA switch chip
  ptp     Precision Time Protocol
  sand    Sand platform

switch(config)#platform
```

Trident and Trident II Platforms

The **`platform ?`** command returns **trident** on switches that operate on Trident or Trident II platforms. Trident II platform switches include options that configure the forwarding and routing tables. To determine the Trident platform that a switch uses, display **platform trident** options.

These commands indicate that the switch is operating on the Trident II platform:

```
switch(config)# platform ?
  ptp      Precision Time Protocol
  trident  Trident chip

switch(config)# platform trident ?
  fabric            Fabric configuration
  forwarding-table  Forwarding table configuration
  mmu               Trident MMU configuration
  routing-table     Routing table configuration

switch(config)#platform trident
```

Fixed and Modular switches are available that operate on the Trident II platform. Refer to Viewing the Model Number to determine the switch's model number. Viewing Modules on 7300 Series Modular Switches displays the modules on a Trident II platform modular switch.

Trident II platform switches typically utilize multiple chips. Multi-Chip Devices describes methods of determining port distribution on multi-chip platforms.

1.1.4.2.3 Linecard Modules and Forwarding Compatibility Mode – 7500 and 7500E Series

Each modular switch linecard module is categorized as first-generation or E-Series:

- First-generation linecard modules support all basic switch functions.
- E-Series linecard modules support provide faster data processing, greater internal table capacities, and advanced encoding formatting.

The forwarding compatibility mode determines the switch's performance capabilities when forwarding data between linecard interfaces. Forwarding compatibility mode settings include:

- **PetraA**: Supports first-generation linecard modules.
- **Arad**: Supports E-Series linecard modules.

Forwarding compatibility mode determines the operational capacity of installed linecards. The following table lists the affect of the forwarding compatibility mode on linecard module types.

Table 2: Linecard Module and Forwarding Mode Performance

Linecard Module Type	Forwarding Compatibility Mode	Linecard Operating Capacity
First-generation	petraA	First-generation performance capacity.
First-generation	arad	Linecard is powered-down.
E-Series	petraA	First-generation performance capacity.
E-Series	arad	E-series performance capacity.

Note: Switches must contain E-Series fabric modules to operate at E-Series performance capacities.

The forwarding compatibility mode is configured by the **`platform sand forwarding mode (7500 and 7500E Series)`** command. This command may be required after exchanging a linecard for a different module type or in switches containing first-generation and E-series linecards.

Without a **`platform sand forwarding mode (7500 and 7500E Series)`** command, forwarding compatibility mode is determined by the first linecard that is operational after reloading the switch. In a switch that is reloaded with a homogeneous module set, forwarding compatibility mode matches its linecards. Switches with a mixed set of modules are typically reloaded in **petraA** mode because first generation modules are usually operational before E-Series modules. However, forwarding compatibility mode in mixed module switches that are reloading is not guaranteed without a **`platform sand forwarding mode`** command.

Example

This command changes the forwarding software mode to support E-series linecard modules. This command should be run only after exchanging all linecards to E-series modules.

```
switch(config)# platform sand forwarding mode arad
switch(config)#
```

1.1.4.2.4 Viewing Modules – 7500 and 7500E Series

The **show module** command displays the model number of all installed modules.

- This command displays the modules of a 7504 switch that contains first-generation modules.

```
switch> show module
Module    Ports Card Type                          Model           Serial No.
--------- ----- ------------------------------------ --------------- -----------
1         2     DCS-7500 Series Supervisor Module    7500-SUP        JSH11440327
2         1     Standby supervisor                   Unknown         Unknown
3         48    48-port SFP+ 10GigE Linecard         7548S-LC        JSH10449938
4         48    48-port SFP+ 10GigE Linecard         7548S-LC        JSH11091247
5         48    48-port SFP+ 10GigE Linecard         7548S-LC        JSH11211614
6         48    48-port SFP+ 10GigE Linecard         7548S-LC        JSH11520288
Fabric1   0     DCS-7504 Fabric Module               7504-FM         JSH11451230
Fabric2   0     DCS-7504 Fabric Module               7504-FM         JSH11451210
Fabric3   0     DCS-7504 Fabric Module               7504-FM         JSH11410115
Fabric4   0     DCS-7504 Fabric Module               7504-FM         JSH11380318
Fabric5   0     DCS-7504 Fabric Module               7504-FM         JSH11340955
Fabric6   0     DCS-7504 Fabric Module               7504-FM         JSH11410128

Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:73:03:06:ac - 00:1c:73:03:06:ac  07.06   4.12.1  Active
2                                                        4.12.1  Standby
3         00:1c:73:03:80:44 - 00:1c:73:03:80:73  06.00           Ok
4         00:1c:73:03:e4:34 - 00:1c:73:03:e4:63  07.10           Ok
5         00:1c:73:12:0b:3f - 00:1c:73:12:0b:6e  07.30           Ok
6         00:1c:73:12:b6:3f - 00:1c:73:12:b6:6e  08.00           Ok
Fabric1                                          05.03           Ok
Fabric2                                          05.03           Ok
Fabric3                                          05.02           Ok
Fabric4                                          05.02           Ok
Fabric5                                          05.02           Ok
Fabric6                                          05.02           Ok switch>
```

- This command displays modules of a 7504 switch that contains E-Series modules.

```
switch> show module
Module    Ports Card Type                          Model           Serial No.
--------- ----- ------------------------------------ --------------- -----------
1         3     DCS-7500E-SUP Supervisor Module      7500E-SUP       JAS13060306
3         72    48 port 10GbE SFP+ & 2x100G Linecard 7500E-72S-LC    JAS12410019
4         72    48 port 10GbE SFP+ & 2x100G Linecard 7500E-72S-LC    JPE13041458
5         72    48 port 10GbE SFP+ & 2x100G Linecard 7500S-72S-LC    JAS12380089
Fabric1   0     DCS-7504-E Fabric Module             7504E-FM        JAS12370008
Fabric2   0     DCS-7504-E Fabric Module             7504E-FM        JAS12380012
Fabric3   0     DCS-7504-E Fabric Module             7504E-FM        JAS12370014
Fabric4   0     DCS-7504-E Fabric Module             7504E-FM        JAS12380008
Fabric5   0     DCS-7504-E Fabric Module             7504E-FM        JAS12380017
Fabric6   0     DCS-7504-E Fabric Module             7504E-FM        JAS12370009

Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:73:00:f4:cd - 00:1c:73:00:f4:ce  00.00   4.12.3  Active
3         00:1c:73:00:9c:7b - 00:1c:73:00:9c:c2  00.00           Ok
4         00:1c:73:28:a0:57 - 00:1c:73:28:a0:9e  00.00           Ok
5         00:1c:73:00:9a:cb - 00:1c:73:00:9b:12  02.07           Ok
Fabric1                                          00.00           Ok
Fabric2                                          00.00           Ok
Fabric3                                          00.00           Ok
Fabric4                                          00.00           Ok
Fabric5                                          00.00           Ok
Fabric6                                          00.00           Ok switch>
```

1.1.4.3 Viewing Modules on 7300 Series Modular Switches

7300 Series Modular switches operate on Trident II platform. The show module command displays the model number of all installed modules.

```
switch> show module
Module    Ports Card Type                              Model           Serial No.
--------- ----- ------------------------------------ --------------- -----------
1         3     Supervisor 7300X SSD                 DCS-7300-SUP-D  JAS13340024
3         128   32 port 40GbE QSFP+ LC               7300X-32Q-LC    JPE13440416
4         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310113
5         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13340033
6         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310103
Fabric1   0     7304X Fabric Module                  7304X-FM        JAS13320077
Fabric2   0     7304X Fabric Module                  7304X-FM        JAS13350043
Fabric3   0     7304X Fabric Module                  7304X-FM        JAS13350050
Fabric4   0     7304X Fabric Module                  7304X-FM        JAS13350056
```

```
Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:73:36:4b:71 - 00:1c:73:36:4b:72  01.01   4.13.3F Active
3         00:1c:73:58:d4:68 - 00:1c:73:58:d4:87  03.04           Ok
4         00:1c:73:36:05:61 - 00:1c:73:36:05:94  02.02           Ok
5         00:1c:73:36:0a:e1 - 00:1c:73:36:0b:14  02.03           Ok
6         00:1c:73:36:02:e1 - 00:1c:73:36:03:14  02.02           Ok
Fabric1                                          00.00           Ok
Fabric2                                          00.00           Ok
Fabric3                                          00.00           Ok
Fabric4                                          00.00           Ok switch>
```

1.1.4.4 Multi-Chip Devices

Trident II, Petra, and Arad platform switches and linecards utilize multiple chips, with Ethernet ports evenly distributed among the chips. Creating multi-port data structures (including port channels) that include ports from multiple chips protects against the failure of an individual chip on a device.

The following sections describe methods of determining port distribution on various switch platforms

Petra Fixed Switches

7048-Series switches are Petra platform devices that distribute ports among two PetraA chips. The **`show platform petraA port-info routing`** command displays the ports that are controlled by each chip.

Example

This command displays the following Ethernet port distribution on a DCS-7048-T switch:

- ***Petra0*** chip controls ***Ethernet 1*** through ***Ethernet 32***.
- ***Petra1*** chip controls ***Ethernet 33*** through ***Ethernt 52***.

```
switch# show platform petraA port-info routing
Petra0 Port Routing Information:
================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType    v4 v6
================================================================================
 CpuTm                     2       0    Cpu       Tm          1  1

 Ethernet1                29       2    Nif       Ethernet    1  1
 Ethernet2                30       3    Nif       Ethernet    1  1

 Ethernet31               59      32    Nif       Ethernet    1  1
 Ethernet32               60      33    Nif       Ethernet    1  1

 RawPetra0/70           2118      70    Recycling Raw         1  1
Petra1 Port Routing Information:
================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType    v4 v6
================================================================================
 CpuTm                     2       0    Cpu       Tm          1  1

 Ethernet33               66       2    Nif       Ethernet    1  1

 Ethernet52               85      21    Nif       Ethernet    1  1
 L3SecondHop1Petra1       86      22    Recycling Ethernet    1  1

 RawPetra1/70           2118      70    Recycling Raw         1  1 switch#
```

Petra Modular Switches

Linecards on 7500-Series modular switches distribute Ethernet ports among multiple petraA chips. The **`show platform petraA port-info routing`** command displays the ports that are controlled by each chip on all PetraA linecards or on a single linecard.

Example

This command displays the following Ethernet port distribution on ***linecard 4*** of a DCS-7504 switch:

- ***Petra4/0*** chip controls ***Ethernet 4/1*** through ***Ethernet 4/8***.
- ***Petra4/1*** chip controls ***Ethernet 4/9*** through ***Ethernet 4/16***.
- ***Petra4/2*** chip controls ***Ethernet 4/17*** through ***Ethernet 4/24***.
- ***Petra4/3*** chip controls ***Ethernet 4/25*** through ***Ethernet 4/32***.
- ***Petra4/4*** chip controls ***Ethernet 4/33*** through ***Ethernet 4/40***.
- ***Petra4/5*** chip controls ***Ethernet 4/41*** through ***Ethernet 4/48***.

```
switch(s1)# show platform petra module 4 port-info routing
Petra4/0 Port Routing Information:
=================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType    v4 v6
=================================================================================
 CpuTm                    2       0    Cpu       Tm           1  0

 Ethernet4/1              221       2    Nif       Ethernet    1  0
 Ethernet4/2              222       3    Nif       Ethernet    1  0
 Ethernet4/3              223       4    Nif       Ethernet    1  0
 Ethernet4/4              224       5    Nif       Ethernet    1  0
 Ethernet4/5              225       6    Nif       Ethernet    1  0
 Ethernet4/6              226       7    Nif       Ethernet    1  0
 Ethernet4/7              227       8    Nif       Ethernet    1  0
 Ethernet4/8              228       9    Nif       Ethernet    1  0

 RawPetra4/0/70           2118      70    Recycling Raw         1  0
Petra4/1 Port Routing Information:
=================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType    v4 v6
=================================================================================
 CpuTm                    2       0    Cpu       Tm           1  0

 Ethernet4/9              253       2    Nif       Ethernet    1  0

Petra4/5 Port Routing Information:
=================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType    v4 v6
=================================================================================

 Ethernet4/41             381       2    Nif       Ethernet    1  0
 Ethernet4/42             382       3    Nif       Ethernet    1  0
 Ethernet4/43             383       4    Nif       Ethernet    1  0
 Ethernet4/44             384       5    Nif       Ethernet    1  0
 Ethernet4/45             385       6    Nif       Ethernet    1  0
 Ethernet4/46             386       7    Nif       Ethernet    1  0
 Ethernet4/47             387       8    Nif       Ethernet    1  0
 Ethernet4/48             388       9    Nif       Ethernet    1  0

switch(s1)#
```

Arad Modular Switches

7500-E Series linecards distribute Ethernet ports among multiple Arad chips. The **`show platform arad port-info routing`** command displays the ports that are controlled by each chip on all Arad linecards.

Example

This command displays the following Ethernet port distribution on the 7500E-72S-LC linecard that is inserted as ***module 3*** in a DCS-7508E switch:

- ***Arad3/0*** chip: ***Ethernet 3/1***– ***Ethernet 3/20***.
- ***Arad3/1 chip***: ***Ethernet 3/21*** – ***Ethernet 3/34*** and ***Ethernet 3/49/1*** – ***Ethernet 3/49/12***.
- ***Arad3/2*** chip: ***Ethernet 3/35*** – ***Ethernet 3/48*** and ***Ethernet 3/50/1*** – ***Ethernet 3/50/12***.

```
switch# show platform arad mapping
```

```
Arad3/0           Port                       SysPhyPort Voq  (Fap,FapPort)
Xlge  Serdes
-------------------------------------------------------------------------------
                  CpuTm                        2    32    (0  ,  0)     n/a   n/a

                  Ethernet3/1                 28   240    (0  ,  2)     n/a   (16)
                  Ethernet3/2                 29   248    (0  ,  3)     n/a   (17)
                  Ethernet3/3                 30   256    (0  ,  4)     n/a   (18)
                  Ethernet3/4                 31   264    (0  ,  5)     n/a   (19)
                  Ethernet3/5                 32   272    (0  ,  6)     n/a   (20)
                  Ethernet3/6                 33   280    (0  ,  7)     n/a   (21)
                  Ethernet3/7                 34   288    (0  ,  8)     n/a   (22)
                  Ethernet3/8                 35   296    (0  ,  9)     n/a   (23)
                  Ethernet3/9                 36   304    (0  , 10)     n/a   (24)
                  Ethernet3/10                37   312    (0  , 11)     n/a   (25)
                  Ethernet3/11                38   320    (0  , 12)     n/a   (26)
                  Ethernet3/12                39   328    (0  , 13)     n/a   (27)
                  Ethernet3/13                40   336    (0  , 14)     n/a   (4)
                  Ethernet3/14                41   344    (0  , 15)     n/a   (5)
                  Ethernet3/15                42   352    (0  , 16)     n/a   (6)
                  Ethernet3/16                43   360    (0  , 17)     n/a   (7)
                  Ethernet3/17                44   368    (0  , 18)     n/a   (0)
                  Ethernet3/18                45   376    (0  , 19)     n/a   (1)
                  Ethernet3/19                46   384    (0  , 20)     n/a   (2)
                  Ethernet3/20                47   392    (0  , 21)     n/a   (3)

                  RawArad3/0/56             2104 16848    (0  , 56)     n/a   n/a

Arad3/1           Port                       SysPhyPort Voq  (Fap,FapPort)
Xlge  Serdes
-------------------------------------------------------------------------------

                  Ethernet3/21                60   496    (1  ,  2)     n/a   (16)

                  Ethernet3/34                73   600    (1  , 15)     n/a   (13)
                  Ethernet3/49/1              74   608    (1  , 16)     n/a   (0)

                  Ethernet3/49/12             85   696    (1  , 27)     n/a   (11)


Arad3/2           Port                       SysPhyPort Voq  (Fap,FapPort)
Xlge  Serdes
-------------------------------------------------------------------------------

                  Ethernet3/35                92   752    (2  ,  2)     n/a   (16)

                  Ethernet3/48               105   856    (2  , 15)     n/a   (13)
                  Ethernet3/50/1             106   864    (2  , 16)     n/a   (0)

                  Ethernet3/50/12            117   952    (2  , 27)     n/a   (11)

switch#
```

Trident II Fixed Switches

Trident II platform devices distribute their ports among multiple Trident II chips. The **`show platform trident system port`** command displays the ports that are controlled by each chip.

Example

This command displays the following Ethernet port distribution on a DCS-7250QX-64-F switch:

- ***Trident 0*** chip controls ***Ethernet 1/1*** through ***Ethernet 16/4***.
- ***Trident 1*** chip controls ***Ethernet 17/1*** through ***Ethernet 32/4***.
- ***Trident 2*** chip controls ***Ethernet 33/1*** through ***Ethernet 48/4***.
- ***Trident 3*** chip controls ***Ethernet 49/1*** through ***Ethernet 64/4***.

```
switch# show platform trident system port

                                                          Port
      Intf              Chip            ModId       Logical       Physical   MMU
-------------------- ----------------- ----------- ------------- ---------- ---
      Ethernet1/1       Linecard0/0      1           1             17         9
      Ethernet1/2       Linecard0/0      1           2             18         10

      Ethernet16/3      Linecard0/0      1           60            107        98
      Ethernet16/4      Linecard0/0      1           61            108        99

      Ethernet64/2      Linecard0/3      4           62            106        97
      Ethernet64/3      Linecard0/3      4           63            107        98
      Ethernet64/4      Linecard0/3      4           64            108        99
```

```
--------------------------------------------------------------------------------
switch#
```

Trident II Modular Switches

Linecards on 7300-Series modular switches distribute Ethernet ports among multiple Trident II chips. The **`show platform trident system port`** command can display the ports that are controlled by each chip on all linecards or on a single chip.

This command displays the following Ethernet port distribution on DCS-7304-F switch that contains a 7300X-32Q-LC linecard as ***module 3***:

- ***Trident 0*** chip controls ***Ethernet 1/1*** through ***Ethernet 16/4*** (on ***module 3***).
- ***Trident 1*** chip controls ***Ethernet 17/1*** through ***Ethernet 32/4*** (on ***module 3***).

```
switch# show platform trident system port

--------------------------------------------------------------------------------------
                                                                  Port
     Intf              Chip             ModId       Logical       Physical       MMU
--------------------- ----------------- ----------- ------------- -------------  ---
    Ethernet3/1/1      Linecard3/0       5           1             17             4
    Ethernet3/2/1      Linecard3/0       5           2             21             5

    Ethernet3/16/3     Linecard3/0       5           51            111            102
    Ethernet3/16/4     Linecard3/0       5           52            112            103

    Ethernet3/32/3     Linecard3/1       6           63            111            102
    Ethernet3/32/4     Linecard3/1       6           64            112            103

--------------------------------------------------------------------------------------
switch#
```

1.1.5 Command Modes

Command modes define the user interface state. Each mode is associated with commands that perform a specific set of network configuration and monitoring tasks.

- Mode Types lists the available modes.
- Navigating Through Command Modes lists mode entry and exit commands.
- Command Mode Hierarchy describes the mode structure.
- Group-Change Configuration Modes describes editing aspects of these modes.

1.1.5.1 Mode Types

The switch includes these command modes:

- EXEC: EXEC mode commands display system information, perform basic tests, connect to remote devices, and change terminal settings. When logging into EOS, you enter EXEC mode.

  EXEC mode prompt: `switch>`
- **Privileged EXEC**: Privileged EXEC mode commands configure operating and global parameters. The list of Privileged EXEC commands is a superset of the EXEC command set. You can configure EOS to require password access to enter Privileged EXEC from EXEC mode.

  Privileged EXEC mode prompt: `switch#`
- **Global Configuration**: Global Configuration mode commands configure features that affect the entire system, such as system time or the switch name.

  Global Configuration mode prompt: `switch(config)#`
- **Interface Configuration**: Interface configuration mode commands configure or enable Ethernet, VLAN, and Port-Channel interface features.

  Interface Configuration mode prompt: `switch(config-if-Et24)#`

- **Protocol specific mode**: Protocol specific mode commands modify global protocol settings. Protocol specific mode examples include **ACL Configuration** and **Router BGP Configuration**.

  The prompt indicates the active command mode. For example, the Router BGP command prompt is `switch(config-router-bgp)#`

1.1.5.2 Navigating Through Command Modes

To change the active command mode, perform one of these actions:

- To enter EXEC mode, log into the switch.
- To enter Privileged EXEC mode from EXEC, type **enable** (or **en**) followed, if prompted, by the **enable** password:

```
switch>en
Password:
switch#
```

- To enter Global Configuration mode from Privileged EXEC, type **configure** (or **config**):

```
switch#config
switch(config)#
```

- To enter Interface Configuration mode from Global Configuration, type **interface** and the name of the interface to be modified:

```
switch(config)#interface Et24
switch(config-if-Et24)#
```

- To enter a protocol specific configuration mode from Global Configuration, type the required command for the desired mode.

```
switch(config)#router bgp 100
switch(config-router-bgp)#
```

- To return one level from any configuration mode, type **exit**.

```
switch(config)#exit
switch#
```

- To return to Privileged EXEC mode from any configuration mode, type **end** or **Ctrl-Z**.

```
switch(config-if-Et24)#<Ctrl-z>
switch#
```

- To return to EXEC mode from Privileged EXEC mode, type **disable** (or **dis**).

```
switch#dis
switch>
```

- To exit EOS and log out of the CLI, type **exit** from EXEC mode or Privileged EXEC mode.

```
switch#exit

login:
```

1.1.5.3 Command Mode Hierarchy

Command modes are hierarchical. The parent mode of a specified command mode is the mode that contains the command that enters the specified mode.

Example

EXEC mode contains the **enable** command, which enters Privileged EXEC mode. Therefore, EXEC is the parent mode of Privileged EXEC.

Commands that are executable in a specified command mode include all commands available in the specified mode plus all commands executable from its parent mode.

Example

EXEC mode includes the **ping** command. EXEC mode is the parent mode of Privileged EXEC mode. Therefore, Privileged EXEC mode includes **ping**.

Additionally, Privileged EXEC is the parent mode of Global Configuration mode. Therefore, Global Configuration mode also includes **ping**.

Executing a configuration mode command from a child mode may change the active command mode.

Example

Global Configuration mode contains **interface ethernet** and **ip access-list** commands, which enter Interface Configuration and Access Control List (ACL) Configuration modes, respectively. When the switch is in Interface Configuration mode, the **ip access-list** command is available and changes the active mode to ACL Configuration.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#ip access-list master-list
switch(config-acl-master-list)#
```

The **exit** command changes the active command mode to its parent mode. When executed from Privileged EXEC or EXEC modes, the exit command terminates the session.

Example

- This command exits Global Configuration mode to Privileged EXEC mode.

```
switch(config)#exit
switch#
```

- This command terminates the user session.

```
switch#exit
```

1.1.5.4 Group-Change Configuration Modes

Group-change modes apply all changes made during an edit session only after exiting the mode. Changes are stored when the user exits the mode, either through an **exit** or **end** command or through a command that enters a different configuration mode.

The **abort** command discards all changes not previously applied.

Access Control List (ACL) and Multiple Spanning Tree (MST) configuration modes are examples of group-change modes.

1.1.6 Managing Switch Configuration Settings

1.1.6.1 Verifying Settings for the Current Mode

To display only the lines of ***running-config*** that affect the current mode, use the **`active`** option of the **`show (various configuration modes)`** command. This command option is available in all configuration modes except global configuration.

Example

Type **`show active`** to display the content of ***running-config*** that affects the current mode. To include default settings in the display, type **`show active all`**.

```
switch(config-router-ospf3)#show active all
ipv6 router ospf 9
   router-id 0.0.0.0
   default-metric 10
   distance ospf intra-area 10
   area 0.0.0.200 default-cost 10
   area 0.0.0.200
   no log-adjacency-changes
   timers spf 5
switch(config-router-ospf3)#
```

To display any comments associated with the current mode, use the **comment** option of the **`show (various configuration modes)`** command.

Example

Type **`show comment`** to display any comments attached to the current mode.

```
switch(config-router-ospf3)#show comment
Comment for router-ospf3:
    Consult Thomas Morton before making changes to the OSPF configuration.
switch(config-router-ospf3)#
```

1.1.6.2 Verifying the Running Configuration Settings

The **`running-config`** command is the virtual file that stores the operating configuration. The **`show running-config`** command displays the **`running-config`**. The command is supported in Privileged EXEC mode.

Example

Type **`show running-config`** in Privileged EXEC mode. The response in the example is truncated to display only the ip route configured.

```
switch#show running-config
! Command: show running-config

!
ip route 0.0.0.0/0 192.0.2.1
!

end switch#
```

1.1.6.3 Adding a Comment to a Configuration Mode

To add a comment to most switch configuration modes, use the **comment (various configuration modes)** command. Comments cannot be modified, but can be replaced by entering the **comment** command again and entering new text. Comments cannot be added to global configuration mode

To append to an existing comment, enter !! followed by additional comment text. To display comments for the active mode, use the **show comment** command. The **no comment** and **default comment** commands remove the comment from **running-config**.

Examples

- These commands enter a comment in Router OSPF3 Mode.

```
switch(config-router-ospf3)#comment
Enter TEXT message. Type 'EOF' on its own line to end.
Consult Thomas Morton before making changes to the OSPF configuration.
EOF
switch(config-router-ospf3)#
```

- These commands append additional information to the comment entered above.

```
switch(config-router-ospf3)#!! x2735
switch(config-router-ospf3)#show comment
Comment for router-ospf3:
    Consult Thomas Morton before making changes to the OSPF configuration.
    x2735
switch(config-router-ospf3)#
```

1.1.6.4 Saving the Running Configuration Settings

startup-config is the file, stored in internal flash memory, that the switch loads when it boots. Configuration changes that are not saved to **startup-config** are lost the next time the switch is booted.

The **write** and **copy running-config startup-config** commands store the operating configuration to **startup-config**. Both commands are supported in Privileged EXEC mode.

Example

These equivalent commands save the current operating configure to the startup-config file.

```
switch#write

switch#copy running-config startup-config
```

The **show startup-config** command displays the startup configuration file. The command is supported in Privileged EXEC mode.

Example

Type **show startup-config** to display the startup configuration file. The response in the example is truncated to display only the ip route configured in Admin Username.

```
switch#show startup-config
! Command: show startup-config
! Startup-config last modified at  Wed Feb 19 08:34:31 2014 by admin
!

!
ip route 0.0.0.0/0 192.0.2.1
```

```
!

end switch#
```

1.1.7 Other Command-Line Interfaces

EOS can access other CLIs that provide switch commands, files, and services.

- Aboot Command-Line Interface describes the boot-loader CLI.
- Bash Shell describes the Bash shell CLI.

1.1.7.1 Aboot Command-Line Interface

Aboot is the switch boot loader. It reads a configuration file from the internal flash or a USB flash drive and attempts to boot a software image.

The switch opens an Aboot shell if the switch does not find a software image, the configuration is corrupted, or the user terminates the boot process. The Aboot shell provides a CLI for manually booting a software image, recovering the internal flash to its default factory state, running hardware diagnostics, and managing files.

1.1.7.2 Bash Shell

The switch provides a Linux Bash shell for accessing the underlying Linux operating system and extensions. The Bash shell is accessible in all command modes except EXEC. Mode Types describes EOC command modes.

- To enter the Bash, type **bash** at the prompt.

```
switch#bash

Arista Networks EOS shell

[admin@Switch ~]$
```

- To exit the Bash, type **logout**, **exit**, or **Ctrl-D** at the Bash prompt.

```
[admin@Switch ~]$ logout
switch#
```

1.1.8 Directory Structure

EOS operates from a flash drive root mounted as the **/mnt/flash** directory on the switch. The EOS CLI supports these file and directory commands:

- **delete**: Delete a file or directory tree.
- **copy**: Copy a file.
- **more**: Display the file contents.
- **diff**: Compares the contents of files located at specified URLs.
- **rename**: Rename a file.
- **cd**: Change the current working directory.
- **dir**: Lists directory contents, including files and subdirectories.
- **mkdir**: Create a directory.
- **rmdir**: Remove a directory.
- **pwd**: Display the current working directory.

Verify flash memory space before copying a file. When a file is copied to flash, it is first written to a temporary file and then renamed to the destination rather than directly overwriting the destination file. This protects the

integrity of the existing file if the **copy** command is interrupted, but requires more free space to complete the process.

Switch directory files are accessible through the Bash shell and Aboot. When entering the Bash shell from the switch, the working directory is located in **/home** and has the name of the user name from which Bash was entered.

Example

These commands were entered from the user name john:

```
switch#bash
[john@switch ~]$ pwd
/home/john
[john@switch ~]$
```

In this instance, the working directory is **/home/john**

When a flash drive is inserted in the USB flash port, flash drive contents are accessible through **/mnt/usb1**.

When entering Aboot, the working directory is the root directory of the boot.

1.1.9 Command-Line Interface Commands

Mode Navigation Commands

- alias
- bash
- configure (configure terminal)
- copy running-config
- daemon
- disable
- enable
- end
- exit

File Transfer Commands

- ip ftp client source-interface
- ip http client local-interface
- ip ssh client source-interface
- ip tftp client source-interface

File Management Commands

- configure checkpoint
- configure network
- copy running-config
- dir
- pwd

Modular Switch Platform Commands

- platform arad lag mode
- platform sand fabric mode (7500 and 7500E Series)
- platform sand forwarding mode (7500 and 7500E Series)
- platform sand lag hardware-only

- show platform sand compatibility
- show platform sand lag hardware-only

CLI Scheduling Commands

- schedule
- schedule config
- show schedule

Event Handler Commands

- action bash
- delay
- event-handler
- event-handler DropCountersHandler
- show event-handler
- show event-handler DropCountersHandler
- trigger

Terminal Parameter Commands

- terminal length
- terminal monitor

Display and Comment Commands

- comment (various configuration modes)
- show (various configuration modes)
- show module
- show version

1.1.9.1 action bash

The **`action bash`** command specifies a Bash shell command to be run when an event handler is triggered. When an event handler is triggered, execution of the associated shell command is delayed by a configurable period set by the delay command. Only a single Bash command may be configured for an event handler, but the command may have multiple arguments. If more than one Bash command must be executed in response to a trigger, create a script containing the desired commands and enter the file path to the script as the argument of the **action bash** command.

To specify the event that will trigger the action, use the trigger command.

If the event handler uses an **`on-intf`** trigger, the following environment variables are passed to the action and can be used as arguments to the Bash command:

- ***$INTF*** interface name.
- ***$OPERSTATE*** current operational status of the specified interface.
- ***$IP-PRIMARY*** current primary IP address of the specified interface.

Event-Handler Configuration

Command Syntax

`action bash` *command*

Parameters

command Bash shell command to be executed when the event handler is triggered.

Examples

- This command configures the event handler “onStartup” to run a script on the flash drive.

```
switch(config-handler-onStartup)#action bash /mnt/flash/myScript1
switch(config-handler-onStartup)#
```

- This command configures the event handler “eth_4” to send email to the specified address when there is a change in the operational status of ***Ethernet interface 4***.

```
switch(config-event-eth_4)#action bash email x@yz.com -s "Et4 $OPERSTATE"
switch(config-event-eth_4)#
```

  The above action uses the ***$OPERSTATE*** variable to include the current operational state (“linkup” or “linkdown”) in the subject of the email. Note that the action will only function if email has been configured on the switch.

1.1.9.2 alias

The **alias** command creates an alias for a CLI command. Entering the alias in the CLI executes the corresponding command. Once created, an alias is accessible in all modes and all user sessions, but is subject to all the restrictions of the original command.

When using a command alias, no tokens may precede the alias except the no and default keywords. However, an alias can incorporate positional parameters.

In online help, aliases are preceded by an asterisk (*****) in this format:

*alias_name=command_name

The **no alias** and **default alias** commands remove the specified alias.

Command Mode

Global Configuration

Command Syntax

`alias` *alias_name command_name*

`no alias` *alias_name*

`default alias` *alias_name*

Parameters

- ***alias_name*** the string which is to be substituted for the original command. The string can include letters, numbers, and punctuation, but no spaces. If the alias_name string is identical to an existing command, the alias will supercede the original command.
- ***command_name*** the command which is to be executed when the alias is entered in the CLI. If the original command requires additional parameters, they must be included in the *command_name* string in the following manner:

  Positional parameters are of the form “%n” and must be whitespace-delimited. The first parameter is represented by “%1” and any additional parameters must be numbered sequentially. When executing the alias a value must be entered for each parameter or the CLI will display the error “% incomplete command”.

Examples

- This command makes **e** an alias for the command **enable.**

```
switch(config)#alias e enable
```

- This command makes **srie** an alias for the command **show running-config interface ethernet 1-6.**

```
switch(config)#alias srie show running-config interface ethernet 1-6
```

- These commands make **ss** an alias for the command **show interfaces ethernet <range> status** with a positional parameter for the port range, then use the alias to display the status of ports 4/1-4/5.

```
switch(config)#alias ss show interfaces ethernet %1 status
switch(config)#ss 4/1-4/5
Port      Name            Status       Vlan       Duplex  Speed Type
Et4/1                     connected    in Po1       full  10000 10GBASE-SRL
Et4/2                     notconnect   in Po1       full  10000 10GBASE-SRL
Et4/3                     notconnect   1            full  10000 10GBASE-SRL
Et4/4                     notconnect   1            full  10000 10GBASE-SRL
Et4/5                     notconnect   1            full  10000 10GBASE-SRL
```

1.1.9.3 bash

The **bash** command starts the Linux Bash shell. The Bash shell gives you access to the underlying Linux operating system and system extensions.

To exit the Bash, type **logout**, **exit**, or **Ctrl-D** at the Bash prompt.

Command Mode

Privileged EXEC

Command Syntax

bash

Examples

- This command starts the Bash shell.

```
switch#bash

Arista Networks EOS shell

[admin@switch ~]$
```

- This command, executed within Bash, exits the Bash shell.

```
[admin@switch ~]$ logout
switch#
```

1.1.9.4 comment (various configuration modes)

The **comment** command adds a comment for the active configuration mode to ***running-config***. Comments cannot be modified, but can be replaced by entering the **comment** command again and entering new text. To append to an existing comment, enter **!!** followed by additional comment text. To display comments, use the **comment** option of the show (various configuration modes) command.

The **no comment** and **default comment** commands remove the comment from ***running-config***.

Comments cannot be added to the global configuration mode through the EOS.

Command Mode

All configuration modes except Global Configuration

Command Syntax

comment *comment_text* EOF

no comment

`default comment` !! *comment_text*

Parameters

- ***comment_text*** To create a comment, enter a message when prompted. The message may span multiple lines.
- **EOF** To end a comment, type EOF on its own line (case sensitive) and press **enter**.

Examples

- This command adds a comment to the active configuration mode.

```
switch(config-sg-radius-RAD-SV1)#comment
Enter TEXT message. Type 'EOF' on its own line to end.
Consult Thomas Morton before making changes to the RADIUS configuration.
EOF
switch(config-sg-radius-RAD-SV1)#
```

- This command appends a line to the comment for the active configuration mode.

```
switch(config-sg-radius-RAD-SV1)#!! x3452
switch(config-sg-radius-RAD-SV1)#
```

1.1.9.5 configure (configure terminal)

The **configure** command places the switch in the Global Configuration mode to configure features at the system level. You can move to Interface Configuration mode and protocol-specific mode from the ***Global Configuration*** mode. The command may also be entered as configure terminal.

Command Mode

Privileged EXEC

Command Syntax

`configure` [terminal]

Example

This command places the switch in the Global Configuration mode.

```
switch>enable
switch#configure
switch(config)#
```

1.1.9.6 configure checkpoint

The **configure checkpoint** command saves the running configuration to a checkpoint file. This checkpoint file can be used for restoring the current running configuration in future, if required.

Command Mode

Privileged EXEC

Command Syntax

`configure checkpoint` {restore *checkpoint_name* | **save** [*checkpoint_name*]}

Parameters

- **restore *checkpoint_name*** restores the running configuration from the specified checkpoint file.
- **save *checkpoint_name*** saves running configuration to the specified checkpoint file.

Guidelines

If the filename already exists, EOS overwrites the filename. If the command is entered without a checkpoint name, the switch automatically saves the checkpoint under the name ckp-date-number where date is the date in YYYYMMDD format and number increments by one for each automatically named checkpoint file.

Examples

- This command saves ***running-config*** to the ***ca_test*** checkpoint file.

```
switch#configure checkpoint save ca_test
```

- This command restores the ***running-config*** from the ***ca_test*** checkpoint file.

```
switch#configure checkpoint restore ca_test
! Preserving static routes. Use 'no ip routing delete-static-routes' to
 clear them.
```

- This command saves ***running-config*** to the ***13Aug2018*** checkpoint file. The **dir** command shows the contents of the checkpoint directory.

```
switch#configure checkpoint save
switch#dir checkpoint:
Directory of checkpoint:/
       -rw-        7426           Aug 13 12:00  ckp-20180813-17
       -rw-        7588           Aug 13 12:10  ckp-20180813-18
       -rw-        8499           Aug 13 12:13  ckp-20180813-19
       -rw-        8499           Aug 13 12:13  ckp-20180813-20
```

1.1.9.7 configure convert

The **configure convert** command converts the current configuration syntax to the specified syntax.

Command Mode

Privileged EXEC

Command Syntax

`configure convert` new-syntax

Parameter

new-syntax converts ***running-config*** to the current version of EOS.

Example

This command converts running-config to the current version of EOS.

```
switch#configure convert new-syntax

WARNING!
Converting existing configuration to new syntax will lose backward
 compatibility.
Make sure you won't downgrade to releases that only support the old syntaxes.

Proceed [ y/n ]
```

1.1.9.8 configure network

The **configure network** command is deprecated. Use the **copy** <*url*> ***running-config*** command to configure the switch from a local file or network location.

1.1.9.9 copy running-config

The current operating configuration of the switch is stored in a virtual file called running-config. The copy running-config command saves the contents of the running-config virtual file to a new location.

Command Mode

Privileged EXEC

Command Syntax

`copy running-config` **DESTINATION**

Parameters

- **DESTINATION** destination for the contents of the running-config file. Values include:
  - **startup-config** the configuration file that the switch loads when it boots.
- The **`copy running-config`**, **`startup-config`**, and **`write`** commands are equivalent.
  - **`file`**: a file in the switch file directory.
  - **`flash`**: a file in flash memory.
  - **`url`** any valid URL.

The **`copy running-config url`** and **`write network url`** commands are equivalent.

Examples

- This command copies running-config to the startup-config file.

```
switch#copy running-config startup-config
switch#
```

- This command copies running-config to a file called ***rc20110617*** in the dev subdirectory of the switch directory.

```
switch#copy running-config file:dev/rc20110617
switch#
```

1.1.9.10 daemon

The **`daemon`** command accesses daemon configuration mode for adding or removing external daemons and scripts, which are then managed by ProcMgr.

The **`no daemon`** and **`default daemon`** commands delete the deamon by removing the corresponding **`daemon`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`daemon` *daemon_name*

`no daemon` *daemon_name*

`default daemon` *daemon_name*

Parameters

daemon_name label that references the daemon configuration mode.

Example

These commands enters daemon configuration mode and initiates the daemon script.

```
switch(config)#daemon process1
switch(config-daemon-process1)#command process-script -i -m
switch(config-daemon-process1)#
```

1.1.9.11 delay

The **delay** command specifies the time in seconds the system will delay between a triggering event and the execution of an event handler action. The default delay is ***20*** seconds.

Command Mode

Event-Handler Configuration

Command Syntax

`delay` *seconds*

Parameters

seconds number of seconds to delay before executing the action. The default is 20.

Example

This command configures the event handler Eth5 to delay 10 seconds before executing.

```
switch(config-handler-Eth5)#delay 10
switch(config-handler-Eth5)#
```

1.1.9.12 dir

The **dir** command displays a list of files on a file system.

Command Mode

Privileged EXEC

Command Syntax

`dir` [**SCOPE**][**FILE TYPE**]

Parameters

- **SCOPE** the files to display. Options include:
  - ***no parameter*** lists normal files in current directory.
  - **/all** list all files, including hidden files.
  - **/recursive** list files recusively.
- **FILE TYPE** The options include:
  - ***no parameter*** lists undeleted files.
  - **all_filesystems** list files on all filesystems including deleted files, undeleted files, and files with errors.
  - **extensions** directory or file name.
  - **file** directory or file name.
  - **flash** directory or file name.
  - **supervisor-peer** directory or file name.
  - **usb1** directory or file name.
  - **system** directory or file name.

Example

This command displays the flash directory.

```
switch# dir flash:
Directory of flash:/

       -rwx   293409892           Oct 23 08:55  EOS-4.11.0.swi
       -rwx   221274543            Sep 6 13:37  EOS-4.7.5.swi
       -rwx   271453650            Sep 4 19:13  EOS_4.10.1-SSO.swi
       -rwx      135168           Dec 31  1979  FSCK0000.REC
       -rwx          26           Oct 23 13:51  boot-config
       -rwx        8570           Sep 10 12:22  cfg_sso_mst
       -rwx        5642           Sep 20 10:35  config.reset
       drwx        4096           Oct 23 13:59  debug
       -rwx          12           Oct 23 13:56  kernel-params
       drwx        4096           Oct 23 14:59  persist
       drwx        4096            Sep 6 14:50  schedule
       -rwx        5970           Oct 23 13:53  startup-config switch#
```

1.1.9.13 disable

The **disable** command exchanges the session's current command mode with the specified privilege level.

Command Mode

Privileged EXEC

Command Syntax

disable [**PRIVILEGE_LEVEL**]

Parameters

PRIVILEGE_LEVEL Session's new privilege level. Value ranges from ***0*** to ***15***. Levels ***2*** through ***15*** place the switch in Privileged EXEC mode. Values of ***0*** or ***1*** leave the switch in EXEC mode.

- ***no parameter*** Session is assigned default level of ***1***.
- <**0** to **15**> Specifies session level.

Restrictions

New privilege level must be less than the session's current level.

Example

This command exits Privileged EXEC mode level of 15 to enter EXEC mode level 1.

```
switch# disable
switch>
```

1.1.9.14 enable

The **enable** command places the switch in Privileged EXEC mode. If an **enable** password is set, the CLI displays a password prompt when a user enters the **enable** command. If the user enters an incorrect password three times, the CLI displays the EXEC mode prompt.

To set a local **enable** password, use the **enable** password command.

Command Mode

EXEC

Command Syntax

enable [**PRIVILEGE_LEVEL**]

Parameters

- **PRIVILEGE_LEVEL** Session's privilege level. Values range from ***0*** to ***15***. Values of ***0*** or ***1*** places the switch in EXEC mode. Any level above ***1*** leaves the switch in Privileged EXEC mode.
  - ***no parameter*** Session is assigned default level of ***15***.
  - <**0** to **15**> Specifies session level.

Example

This command places the switch in Privileged EXEC mode with the default privilege level of ***15***.

```
switch>enable
switch#
```

1.1.9.15 end

The **end** command exits to Privileged Exec mode from any Configuration mode. If the switch is in a group-change mode (such as ACL-Configuration mode or MST-Configuration mode), the **end** command also saves all pending changes made in that mode to ***running-config***.

Command Mode

All configuration modes

Command Syntax

```
end
```

Example

This command exits to Privileged Exec mode.

```
switch(config-if-Et25)#end
switch#
```

1.1.9.16 event-handler DropCountersHandler

The **event-handler DropCountersHandler** command enables the adverse drop counters monitor with event handlers. The DropCountersHandler event handler is enabled by default, and can be customized for duration of time window and threshold levels.

The **no event-handler DropCountersHandler** command disables the adverse drop counters monitor with event handlers. The **default event-handler DropCountersHandler** command resets the DropCountersHandler event handler to the system default.

Command Mode

Global Configuration

Command Syntax

```
event-handler DropCountersHandler
```

```
no event-handler DropCountersHandler
```

```
default event-handler DropCountersHandler
```

Examples

- These commands customize the delay, polling interval, and condition for width (-w), violation count (-c), and threshold (-t) of this event handler. Each parameter may be customized separately, with all other parameters remaining unchanged.

```
switch(config)#event-handler DropCountersHandler
```

```
switch(config-DropCountersHandler)#action bash DropCounterLog.py -l
switch(config-DropCountersHandler)#delay 0
switch(config-DropCountersHandler)#trigger on-counters
switch(config-DropCountersHandler-counters)#poll interval 60
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -w 800" > 0
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -c 5" > 0
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -t 200" > 0
switch(config-DropCountersHandler-counters)#
```

- This command disables this event handler.
-
```
switch(config)#no event-handler DropCountersHandler
switch(config)#
```

1.1.9.17 event-handler

An event handler executes a Linux Bash shell command in response to a specific system event. An event handler consists of a Bash command, a trigger and a delay; when the trigger event occurs, the action is scheduled to run after ***delay*** seconds.

The **`event-handler`** command places the switch in event-handler configuration mode for the specified event handler. If the named event handler does not already exist, this command creates it. Event-handler configuration mode is a group change mode that configures event handlers.

Changes made in a group change mode are saved by leaving the mode through the **`exit`** command or by entering another configuration mode.

These commands are available in event-handler configuration mode:

- action bash
- delay
- trigger

The **`no event-handler`** and **`default event-handler`** commands delete the specified event handler by removing it from ***running config***.

Command Mode

Global Configuration

Command Syntax

`event-handler` *name*

`no event-handler` *name*

`default event-handler` *name*

Parameters

name name of the event handler to be configured. If the named event handler does not already exist, this command will create it.

Example

This command places the switch in event-handler configuration mode for an event handler called ***Eth_5***.

```
switch(config)#event-handler Eth_5
switch(config-handler-Eth_5)#
```

1.1.9.18 exit

The exit command places the switch in the parent of the command mode from which the exit command was entered.

- When used in Global configuration, the switch enters ***Privileged EXEC mode***.
- When used in ***EXEC*** or ***Privileged EXEC mode***, the **`exit`** command terminates the user session.
- When the command is used in a group-change mode (such as ***ACL-Configuration mode*** or ***MST-Configuration mode***), the **`exit`** command also applies all pending changes made in that mode.

Command Mode

All modes

Command Syntax

`exit`

Examples

- This command exits Global Configuration mode to ***Privileged EXEC*** mode.

```
switch(config)#exit
switch#
```

- This command terminates the user session.

```
switch#exit
```

1.1.9.19 ip ftp client source-interface

By default, the FTP (File Transfer Protocol) source IP address is selected by the switch (the IP address of the source interface if one is assigned). The **`ip ftp client source-interface`** command allows the user to override the default FTP source address.

The **`ip ftp client source-interface`** and **`ip ftp source-interface`** commands are functionally equivalent. In each case, **`ip ftp client source-interface`** is stored in ***running-config***.

The **`no ip ftp client source-interface`** and **`default ip ftp client source-interface`** commands restore default behavior by removing the **`ip ftp client source-interface`** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip ftp` [*client*] **source-interface INTERFACE** [**vrf** *vrf_name*]

`no ip ftp` [*client*] *source-interface*

`default ip ftp` [*client*] *source-interface*

Parameters

- ***client*** Parameter has no functional effect.
- **INTERFACE** Interface providing the IP address. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **tunnel *t_num*** Tunnel interface specified by ***t_num***.

- **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vrf *vrf_name*** Uses the specified user-defined VRF.

Examples

- These commands configure the ***10.10.121.15*** as the source IP address the switch uses when communicating with FTP servers.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#ip address 10.10.121.15/24
switch(config-if-Et17)#ip ftp client source-interface ethernet 17
switch(config)#
```

- This command configures the switch to use ***interface tunnel 45*** and ***vrf vrf01*** when communicating with FTP servers.

```
switch(config)#ip ftp client source-interface tunnel 45 vrf vrf01
switch(config)#
```

1.1.9.20 ip http client local-interface

The **ip http client local-interface** command specifies the source IP address for hypertext transfer protocol (HTTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **no ip http client local-interface** and **default ip http client local-interface** commands restore default behavior by removing the **ip http client local-interface** statement from ***running-config*** .

Command Mode

Global Configuration

Command Syntax

`ip http client local-interface` **INTERFACE** [**vrf** *vrf_name*]

`no ip http client local-interface`

`default ip http client local-interface`

Parameters

- **INTERFACE** Interface providing the IP address. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vrf *vrf_name*** Uses the specified user-defined VRF.

Examples

- These commands configure the ***10.15.17.9*** as the source IP address the switch uses when communicating with HTTP servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.15.17.9/24
switch(config-if-Vl10)#ip http client local-interface vlan 10
switch(config)#
```

- This command configures the switch to use ***interface tunnel 45*** and ***vrf vrf01*** when communicating with HTTP servers.

```
switch(config)#ip http client local-interface tunnel 45 vrf vrf01
switch(config)#
```

1.1.9.21 ip ssh client source-interface

The **ip ssh client source-interface** command specifies the source IP address for secure shell (SSH) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **ip ssh client source-interface** and **ip ssh source-interface** commands are functionally equivalent. In each case, **ip ssh client source-interface** is stored in ***running-config***.

The **no ip ssh client source-interface** and **default ip ssh client source-interface** commands restore default behavior by removing the **ip ssh client source-interface** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip ssh` [client] `source-interface` **INTERFACE** [**vrf** *vrf_name*]

`no ip ssh` [client] `source-interface`

`default ip ssh` [client] `source-interface`

Parameters

- **client** Parameter has no functional effect.
- **INTERFACE** Interface providing the IP address. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vrf *vrf_name*** Uses the specified user-defined VRF.

Examples

- These commands configure the ***10.17.17.9*** as the source IP address the switch uses when communicating with SSH servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.17.17.9/24
switch(config-if-Vl10)#ip ssh client source-interface vlan 10
switch(config)#
```

- This command configures the switch to use ***interface tunnel 45*** and ***vrf vrf01*** when communicating with SSH servers.

```
switch(config)#ip ssh client source-interface tunnel 45 vrf vrf01
switch(config)#
```

1.1.9.22 ip tftp client source-interface

The **ip tftp client source-interface** command specifies the source IP address for Trivial File Transfer Protocol (TFTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **ip tftp client source-interface** and **ip tftp source-interface** commands are functionally equivalent. In each case, **ip tftp client source-interface** is stored in ***running-config***.

The **no ip tftp client source-interface** and **default ip tftp client source-interface** commands restore default behavior by removing the **ip tftp client source-interface** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip tftp` [client] `source-interface` **INTERFACE** [vrf *vrf_name*]

`no ip tftp` [client] `source-interface`

`default ip tftp` [client] `source-interface`

Parameters

- **client** Parameter has no functional effect.
- **INTERFACE** Interface providing the IP address. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vrf *vrf_name*** Uses the specified user-defined VRF.

Examples

- These commands configure the ***10.15.17.9*** as the source IP address the switch uses when communicating with TFTP servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.15.17.9/24
switch(config-if-Vl10)#ip tftp client source-interface vlan 10
switch(config)#
```

- This command configures the switch to use ***interface tunnel 45*** and ***vrf vrf01*** when communicating with TFTP servers.

```
switch(config)#ip tftp client source-interface tunnel 45 vrf vrf01
switch(config)#
```

1.1.9.23 platform arad lag mode

The **platform arad lag mode** command allows configuration of LAGs with more than 16 members.

Command Mode

Global Configuration

Command Syntax

platform arad lag mode [**1024x16** | **256x64** | **512x32**]

Examples

- This command configures 1024 LAGs with 16 members each.

```
switch(config)#platform arad lag mode 1024x16
! Change will take effect only after switch reboot.
switch(config)#
```

- This command configures 256 LAGs with 64 members each.

```
switch(config)#platform arad lag mode 256x64
! Change will take effect only after switch reboot.
switch(config)#
```

- This command configures 512 LAGs with 32 members each.

```
switch(config)#platform arad lag mode 512x32
! Change will take effect only after switch reboot.
switch(config)#
```

1.1.9.24 platform sand fabric mode (7500 and 7500E Series)

The **`platform sand fabric mode`** command specifies the fabric mode under which the switch operates after the next system reload. The command has no operational effect until the switch reloads.

The fabric mode determines the modular switch's fabric performance capabilities and must be compatibile with the installed fabric modules. Fabric mode settings include:

- **fe600**: Supports first-generation fabric modules.
- **fe1600**: Supports E-Series fabric modules.



Note: Switches that reload in **petraA** forwarding compatibility mode (platform sand forwarding mode (7500 and 7500E Series)) also reload in fe600 fabric mode regardless of the presence of a **`platform sand fabric mode`** statement in ***running-config***.

The switch's fabric mode setting must match the capabilities of its installed fabric modules. Reloading the switch in a different mode may be required after exchanging fabric modules for a different module type. The show module command displays the fabric modules in the switch.

Each fabric module is categorized as first-generation or E-Series:

- First-generation fabric modules support all basic switch functions.
- E-Series fabric modules support faster fabric link speeds, greater internal table capacities, and advanced encoding formatting.

E-series fabric modules can operate in **fe600** mode, but are limited to first-generation fabric performance. First-generation modules cannot operate in **fe600** mode. Switches containing both types of modules must be set to **fe600** mode. Best practice is to avoid switch configurations with mixed fabric modules.

When a switch reloads, fabric mode is determined by the following (in order of precedence):

1. Switches reloading in **petraA** forwarding compability mode also reload in **fe600** fabric mode .
2. As specified by the **platform sand fabric mode** statement in ***running-config***.
3. The first fabric module that becomes operational as the switch reloads.

In switches with a homogeneous module set, the fabric mode matches its fabric modules. Switches with a mixed set of modules are typically reloaded in **fe600** mode because first generation modules are usually operational before E-Series modules. However, the fabric mode in mixed module switches that are reloading cannot be guaranteed in the absence of the first two conditions.

The **`no platform sand fabric mode`** and **`default platform sand fabric mode`** commands remove the **`platform sand fabric mode`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`platform sand fabric mode` [**MODE_SETTING**]

`no platform sand fabric mode`

`default platform sand fabric mode`

Parameters

- **MODE_SETTING** Specifies the switch's fabric mode. Options include:
  - **fe16000** E-Series fabric mode.
  - **fe600** First-generation fabric mode.

Example

This command configures the switch to reload in **fe1600** fabric mode to support E-series fabric modules. After issuing this command, the switch should be reset only after exchanging all switch fabric modules to E-series modules.

```
switch(config)#platform sand fabric mode fe1600
switch(config)#exit
switch#show platform sand compatibility
                    Configuration        Status
Forwarding mode     None                 Arad
Fabric mode         Fe1600               Fe600 switch#
```

1.1.9.25 platform sand forwarding mode (7500 and 7500E Series)

The **`platform sand forwarding mode`** command specifies the forwarding compatibility mode under which the switch operates after the next system reload. The command has no operational effect until the switch reloads.

Forwarding compatibility mode specifies switch forwarding capabilities and configures performance capacity of installed linecards. Forwarding compatibility modes settings include:

- **petraA**: Supports first-generation fabric modules.
- **arad**: Supports E-Series fabric modules.



Note: Switches that reload in **petraA** forwarding compatibility mode also reload in **fe600** fabric mode regardless of the presence of a platform sand fabric mode (7500 and 7500E Series) statement in ***running-config***.

This command may be required after exchanging a linecard for a different module type or in switches containing first-generation and E-series linecards. The show module command displays the linecard modules in the switch.

Each modular switch linecard module is categorized as first-generation or E-Series:

- First-generation linecards support all basic switch functions.
- E-Series linecards support provide faster data processing, greater internal table capacities, and advanced encoding formatting.

The forwarding compatibility mode determines the operational capacity of installed linecards. The following table lists the affect of the forwarding compatibility mode on all linecard module types.

Table 3: Linecard Module and Forwarding Mode Performance

Linecard Module Type	Forwarding Software Mode	Linecard Operating Capacity
First-generation	petraA	Linecard performs at first-generation performance capacity.
First-generation	arad	Linecard is powered-down.
E-Series	petraA	Linecard performs at first-generation performance capacity.
E-Series	arad	Linecard performs at E-series performance capacity.



Note: Linecards operate at E-Series performance capacities only on switches that contain E-Series fabric modules and have a fabric mode setting of **fe1600** fabric mode (platform sand fabric mode (7500 and 7500E Series)).

Without a platform sand fabric mode (7500 and 7500E Series) command, forward compatibility mode is determined by the first linecard that becomes operational after reloading the switch. In a switch that is reloaded with a homogeneous module set, forwarding compatibility mode matches its linecards. Switches with a mixed set of modules are typically reloaded in **petraA** mode because first generation modules are usually operational before E-Series modules. However, forwarding compatibility mode in mixed module switches that are reloading is not guaranteed without a **`platform sand forwarding mode`** command.

The **`no platform sand forwarding mode`** and **`default platform sand forwarding mode`** commands restore the **platform sand forwarding mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`platform sand forwarding mode` [**MODE_SETTING**]

`no platform sand forwarding mode`

`default platform sand forwarding mode`

Parameters

- **MODE_SETTING** Specifies the switch's software forwarding mode. Options include:
  - **arad** the switch supports E-Series linecard capabilities.
  - **petraA** the switch supports first-generation linecard capabilities.

Example

This command changes the forwarding software mode to support E-series linecard modules. This command should be run only after exchanging all linecards to E-series modules.

```
switch(config)#platform sand forwarding mode arad
switch(config)#
```

1.1.9.26 platform sand lag hardware-only

The **`platform sand lag hardware-only`** command specifies that all LAGs will use hardware resources including single member LAGs. Hardware resource allocation and deallocation traffic disruption occurs on the first member addition or deletion, rather than the second member addition or deletion.

The **no platform sand lag hardware-only** and **default platform sand lag hardware-only** commands specify that LAGs are not required to be implemented in hardware, and therefore some LAGs may be implemented in software. Permitting both hardware and software LAGs may increase the total number of port-channels because we have no resource limit on the number of software LAGs.

Command Mode

Global Configuration

Command Syntax

```
platform sand lag hardware-only

no platform sand lag hardware-only

default platform sand lag hardware-only
```

Examples

- This command configures all LAGs to use hardware resources. All existing one member LAGs will be allocated hardware resources, when available.

```
switch(config)#platform sand lag hardware-only
switch(config)#
```

- This command allows certain LAGs (single member LAGs) to not consume hardware resources. All existing one member LAGs will release their hardware resources.

```
switch(config)#no platform sand lag hardware-only
switch(config)#
```

1.1.9.27 pwd

The **pwd** command displays the working directory.

Command Mode

Privileged EXEC

Command Syntax

```
pwd
```

Example

This command shows that the working is Flash.

```
switch# pwd
flash:/
switch#
```

1.1.9.28 schedule config

The schedule config command sets configuration parameters to the CLI scheduler.

The **no schedule config max-concurrent-jobs** and **default schedule config max-concurrent-jobs** commands reset the limit of maximum concurrent jobs to the default value of 1 by removing the corresponding schedule config max-concurrent-jobs statement from running-config.

The **no schedule config prepend-hostname-logfile** and **default schedule config prepend-hostname-logfile** commands reset the log filenames to the default state.

Command Mode

Global Configuration

Command Syntax

`schedule config`{max-concurrent-jobs *limit* | prepend-hostname-logfile}

`no schedule config` {max-concurrent-jobs *limit* | prepend-hostname-logfile}

`default schedule config` {max-concurrent-jobs *limit* | prepend-hostname-logfile}

Parameters

- **max-concurrent-jobs *limit*** specifies the maximum number of concurrent commands that can run on the switch. The maximum concurrent jobs ranges from ***1*** to ***4***. The default value is ***1***.
- **prepend-hostname-logfile** enables prepending hostnames to log filenames. By default, this option is enabled.

Examples

- This command configures to concurrently run a maximum of three commands on the switch.

```
switch(config)#schedule config max-concurrent-jobs 3
switch(config)#show schedule summary
Maximum concurrent jobs  3
Prepend host name to logfile: No
Name            At time      Last    Interval   Timeout   Max     Logfile Location
Status
                             time     (mins)    (mins)    log
                                                         files
------------- ------------- ------- ---------- -------- -------- ------------------------------- ------
tech-support      now        00:29      60        30      100     flash:schedule/tech-support/   Success
thelp           12:02:00     00:02      60        40      100     flash:schedule/thelp/
Fail
                06/05/2018
switch(config)#
```

- This command enables prepending the hostname to log filenames.

```
switch(config)#schedule config prepend-hostname-logfile
switch(config)#show schedule summary
Maximum concurrent jobs  3
Prepend host name to logfile: Yes
Name            At time      Last    Interval   Timeout   Max     Logfile Location
Status
                             time     (mins)    (mins)    log
                                                         files
------------- ------------- ------- ---------- -------- -------- ------------------------------- ------
tech-support      now        00:29      60        30      100     flash:schedule/tech-support/   Success
thelp           12:02:00     00:02      60        40      100     flash:schedule/thelp/
Fail
                06/05/2018
switch(config)#
```

1.1.9.29 schedule

The schedule command facilitates the periodic execution of a specified CLI command. Command parameters configure the start time of periodic execution, the interval between consecutive execution instances, the maximum time allotted for command execution, and the maximum number of log files that can be created.

The **`no schedule`** and **`default schedule`** commands disable execution of the specified command.

Command Mode

Global Configuration

Command Syntax

`schedule` *schedule_name* **PERIOD** {max-log-files *count* | timeout *timeout_interval*}{command *cmd* | logging verbose | loglocation *flash:*}

`no schedule` *schedule_name*

`default schedule` *schedule_name*

Parameters

- ***schedule_name*** Label associated with the scheduled command.

- **PERIOD** Start time for execution and interval between consecutive execution instances. The interval ranges from ***2*** to ***1440*** minutes. The default interval while scheduling the **`show tech-support`** command is ***60*** minutes. Options include:
  - **at** Start time for execution. Options include:
    - ***hh:mm:ss*** **interval** ***interval*** The command execution starts at the specified time and repeats at the specified interval.
    - ***hh:mm:ss mm/dd/yyyy*** **interval** ***interval*** The command execution starts at the specified time on the specified day and repeats at the specified interval.
    - ***hh:mm:ss*** **once** The command execution starts at the specified time and does not repeat.
    - ***hh:mm:ss mm/dd/yyyy*** **once** The command execution starts at the specified time on the specified day and does not repeat.
    - ***hh:mm:ss yyyy-mm-dd*** **interval** ***interval*** The command execution starts at the specified time on the specified day and repeats at the specified interval.
    - ***hh:mm:ss yyyy-mm-dd*** **once** The command execution starts at the specified time on the specified day and does not repeat.
  - **interval** ***interval*** The command execution starts immediately and repeats at the specified interval.
  - **now interval** ***hh:mm:ss*****interval** ***interval******hh:mm:ss mm/dd/yyyy*****interval** ***interval*****once*****hh:mm:ss******hh:mm:ss mm/dd/yyyy*****once*****hh:mm:ss yyyy-mm-dd*****interval** ***interval*****once*****hh:mm:ss yyyy-mm-dd*****interval** ***interval*****now interval** ***interval*** The command execution starts immediately and repeats at the specified interval.
- **max-log-files** ***count*** Maximum number of log files command generates for command output. The count of maximum log files ranges from ***1*** to ***10000***. The default count of maximum log files while scheduling the **`show tech-support`** command is ***100***.
- **timeout** ***timeout_interval*** Maximum time allotted for the script execution. The timeout interval ranges from ***1*** to ***480*** minutes. The default timeout is ***30*** minutes.

  Note: The command execution is terminated if it exceeds the specified timeout interval. The timeout allotted for the scheduled command must not be greater than the corresponding interval.
- **command** ***cmd*** The command that needs to be executed.
- **logging verbose** Sets the logging level to “verbose.” A syslog entry is added after the execution of the scheduled command, regardless of whether the scheduled command has succeeded or failed. In the absence of **logging verbose**, the syslog entry is added only if the execution of the scheduled command fails with an error.
- **loglocation** ***destination*** The flash destination for scheduled command output files.

Guidelines

Log files created by the command are stored in the **flash:/schedule/** ***scheduled_name*** **/** directory. Empty log files are created for commands that do not generate any output.

Examples

- This command saves the running configuration contents to the log file every hour with immediate effect and creates a maximum of 24 log files.

```
switch(config)#schedule backup-test interval 60 max-log-files 24 command
 show running-config
```

- This command starts the script execution at 12:00:00 and repeats every 720 minutes. The script execution is terminated if it exceeds 20 minutes. It generates a maximum of one log file because the specified bash command does not have an output.

```
switch(config)#schedule ms1 at 12:00:00 interval 720 timeout 20 max-log-
files 1 command bash /mnt/flash/myscript.sh
```

- The show schedule command lists the commands currently scheduled for periodic execution and displays the summary of the specified scheduled command.

```
switch#show schedule summary
Maximum concurrent jobs  1
Prepend host name to logfile: Yes

Name At time Last  Interval Timeout  Max    Logfile Location    Status
             time  (mins)   (mins)   log
                                     files
---- ------- ----- -------- -------- ------ -------------------  -------
ms1   now    23:03   720       20      1    flash:/schedule/ms1 Success switch#
```

1.1.9.30 show (various configuration modes)

The **show** command, when executed within a configuration mode, can display data in ***running-config*** for the active configuration mode.

Command Mode

All configuration modes except Global Configuration

Command Syntax

`show` [**DATA_TYPE**]

Parameters

- **DATA_TYPE** Specifies display contents. Values include:
  - **active** Displays ***running-config*** settings for the configuration mode.
  - **active all** Displays ***running-config*** plus defaults for the configuration mode.
  - **active all detail** Displays ***running-config*** plus defaults for the configuration mode.
  - **comment** Displays comment entered for the configuration mode.

Related Commands

The **show** commands in ACL-configuration mode and MST-configuration mode include the **active** and **comment** options along with additional mode-specific options.

Example

This command shows the server-group-TACACS+ configuration commands in ***running-config***.

```
switch(config-sg-tacacs+-TAC-GR)#show active
   server TAC-1
   server 10.1.4.14
switch(config-sg-tacacs+-TAC-GR)#
```

1.1.9.31 show event-handler DropCountersHandler

The show event-handler command displays details of the DropCountersHandler event handler.

Command Mode

Privileged EXEC

Command Syntax

`show event-handler DropCountersHandler`

Example

This command displays details of this event handler.

```
switch(config)#show event-handler DropCountersHandler
Event-handler DropCountersHandler (BUILT-IN)
```

```
Trigger: on-counters delay 0 seconds
  Polling Interval: 60 seconds
  Condition: bashCmd."DropCounterMonitor.py" > 0
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: DropCounterLog.py -l
Action expected to finish in less than 20 seconds
Total Polls: 39
Last Trigger Detection Time: 38 minutes 22 seconds ago
Total Trigger Detections: 1
Last Trigger Activation Time: 38 minutes 22 seconds ago
Total Trigger Activations: 1
Last Action Time: Never
Total Actions: 1

switch(config)#
```

1.1.9.32 show event-handler

The show event-handler command displays the contents and activation history of a specified event handler or all event handlers.

Command Mode

Privileged EXEC

Command Syntax

`show event-handler` [*handler_name*]

Parameters

handler_name optional name of an event handler to display. If no parameter is entered, the command displays information for all event handlers configured on the system.

Example

This command displays information about an event handler called "*eth_5*".

```
switch#show event-handler Eth_5
Event-handler Eth_5
Trigger: on-intf Ethernet5 on ip delay 20 seconds
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: :
Device-health Action: None
Action expected to finish in less than 10 seconds
Last Trigger Detection Time: 15 days 2 hours 19 minutes ago
Total Trigger Detections: 1
Last Trigger Activation Time: 15 days 2 hours 19 minutes ago
Total Trigger Activations: 1
Last Action Time: 15 days 2 hours 19 minutes ago
Total Actions: 1 switch#
```

1.1.9.33 show module

The **`show module`** command displays information that identifies the supervisor, fabric, and linecard modules in a modular switch, including model number, serial number, hardware version number, software version (supervisors only), MAC address (supervisors and linecards), and operational status.

Command Mode

EXEC

Command Syntax

`show module` [**MODULE_NAME**]

Parameters

- **MODULE_NAME** Specifies modules for which data is displayed. Options include:
  - ***no parameter*** All modules (identical to **all** option).
  - **fabric *fab_num*** Specified fabric module. Number range varies with switch model.
  - **linecard *line_num*** Linecard module. Number range varies with switch model.
  - **supervisor *super_num*** Supervisor module. Number range varies with switch model.
  - ***mod_num*** Supervisor (1 to 2) or linecard (3 to 18) module.
  - **all** All modules.

Related Command

show version displays model and serial numbers of modular system components.

Examples

- This command displays information about all installed modules on a DCS-7504 switch.

```
switch#show module
Module    Ports Card Type                           Model           Serial No.
--------- ----- ----------------------------------- --------------- -----------
1         2     DCS-7500 Series Supervisor Module   7500-SUP        JSH11440327
2         1     Standby supervisor                  Unknown         Unknown
3         48    48-port SFP+ 10GigE Linecard        7548S-LC        JSH10315938
4         48    48-port SFP+ 10GigE Linecard        7548S-LC        JSH11665247
5         48    48-port SFP+ 10GigE Linecard        7548S-LC        JSH11834614
6         48    48-port SFP+ 10GigE Linecard        7548S-LC        JSH11060688
Fabric1   0     DCS-7504 Fabric Module              7504-FM         JSH11244430
Fabric2   0     DCS-7504 Fabric Module              7504-FM         JSH11892120
Fabric3   0     DCS-7504 Fabric Module              7504-FM         JSH11941115
Fabric4   0     DCS-7504 Fabric Module              7504-FM         JSH11661618
Fabric5   0     DCS-7504 Fabric Module              7504-FM         JSH11757555
Fabric6   0     DCS-7504 Fabric Module              7504-FM         JSH11847728

Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:23:03:06:ac - 00:1c:23:03:06:ac  07.06   4.12.1  Active
2                                                        4.12.1  Standby
3         00:1c:23:03:80:44 - 00:1c:23:03:80:73  06.00           Ok
4         00:1c:23:03:e4:34 - 00:1c:23:03:e4:63  07.10           Ok
5         00:1c:23:12:0b:3f - 00:1c:23:12:0b:6e  07.30           Ok
6         00:1c:23:12:b6:3f - 00:1c:23:12:b6:6e  08.00           Ok
Fabric1                                          05.03           Ok
Fabric2                                          05.03           Ok
Fabric3                                          05.02           Ok
Fabric4                                          05.02           Ok
Fabric5                                          05.02           Ok
Fabric6                                          05.02           Ok switch#
```

- This command displays information about all installed modules on a DCS-7304 switch.

```
switch#show module
Module    Ports Card Type                           Model           Serial No.
--------- ----- ----------------------------------- --------------- -----------
1         3     Supervisor 7300X SSD                DCS-7300-SUP-D  JAS13340024
3         128   32 port 40GbE QSFP+ LC              7300X-32Q-LC    JPE13440416
4         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310113
5         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13340033
6         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310103
Fabric1   0     7304X Fabric Module                 7304X-FM        JAS13320077
Fabric2   0     7304X Fabric Module                 7304X-FM        JAS13350043
Fabric3   0     7304X Fabric Module                 7304X-FM        JAS13350050
Fabric4   0     7304X Fabric Module                 7304X-FM        JAS13350056

Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:73:36:4b:71 - 00:1c:73:36:4b:72  01.01   4.13.3F Active
3         00:1c:73:58:d4:68 - 00:1c:73:58:d4:87  03.04           Ok
4         00:1c:73:36:05:61 - 00:1c:73:36:05:94  02.02           Ok
5         00:1c:73:36:0a:e1 - 00:1c:73:36:0b:14  02.03           Ok
6         00:1c:73:36:02:e1 - 00:1c:73:36:03:14  02.02           Ok
```

```
Fabric1                                          00.00          Ok
Fabric2                                          00.00          Ok
Fabric3                                          00.00          Ok
Fabric4                                          00.00          Ok switch#
```

1.1.9.34 show platform sand compatibility

The **show sand platform compatibility** command displays the fabric and forwarding modes. These modes determine switch forwarding capabilities and programs performance capacity of installed linecards

sinformation that identifies the supervisor, fabric, and linecard modules in the modular switch, including model number, serial number, hardware version number, software version (supervisors only), MAC address (supervisors and linecards), and operational status.

Command Mode

Privileged EXEC

Command Syntax

```
show platform sand compatibility
```

Related Commands

- platform sand fabric mode (7500 and 7500E Series) specifes the fabric software mode.
- platform sand forwarding mode (7500 and 7500E Series) specifes the forwarding software mode.

Example

This command indicates that the switch is in Fe600 fabric mode and PetraA forwarding mode.

```
switch#show platform sand compatibility
                    Configuration        Status
Forwarding mode     None                 PetraA
Fabric mode         None                 Fe600 switch#
```

1.1.9.35 show platform sand lag hardware-only

The **show platform sand lag hardware-only** command displays whether or not LAGs are hardware-only.

Command Mode

Privileged EXEC

Command Syntax

```
show platform sand lag hardware-only
```

Examples

- This command indicates that LAGs are hardware-only.

```
switch(config)#platform sand lag hardware-only
switch(config)#exit
switch#show platform sand lag hardware-only
Hardware resources are used for all LAGs: True switch#
```

- This command indicates that LAGs are not hardware-only.

```
switch(config)#no platform sand lag hardware-only
switch(config)#exit
switch#show platform sand lag hardware-only
Hardware resources are used for all LAGs: False switch#
```

1.1.9.36 show schedule

The **show schedule** command displays logging output on the terminal during the current terminal session. This command affects only the local monitor. The no terminal monitor command disables direct monitor display of logging output for the current terminal session.

The **show schedule** command displays the list of active scheduled commands and the summary of specified scheduled command.

Command Mode

Global Configuration

Command Syntax

`show schedule` {*schedule_name* | summary}

Parameters

- ***schedule_name*** displays the summary of the specified scheduled command.
- **summary** displays the list of active scheduled commands.

Examples

- This command displays the summary of the ***thelp*** schedule.

```
switch(config)#show schedule thelp
The last CLI command failed with exit status 1
CLI command "show THelp" is scheduled next at "02:02:35 06/19/2018", interval is
60 minutes
Timeout is 40 minutes
Maximum of 100 log files will be stored
Verbose logging is off
100 log files currently stored in flash:/schedule/thelp

Start time                 Size          Filename
-------------------------- ------------- ----------------------------------
Jun 19 2018 01:02       60.0 bytes       ro301_thelp_2018-06-19.0102.log.gz
Jun 19 2018 00:02       60.0 bytes       ro301_thelp_2018-06-19.0002.log.gz
Jun 18 2018 23:02       60.0 bytes       ro301_thelp_2018-06-18.2302.log.gz
Jun 18 2018 22:02       60.0 bytes       ro301_thelp_2018-06-18.2202.log.gz
Jun 18 2018 21:02       60.0 bytes       ro301_thelp_2018-06-18.2102.log.gz

switch(config)#
```

- This command displays the summary of scheduled commands.

```
switch(config)#show schedule summary
Maximum concurrent jobs  1
Prepend host name to logfile: Yes
Name            At time      Last    Interval   Timeout   Max     Logfile Location
Status
                             time     (mins)    (mins)    log
                                                         files
------------- ------------- ------- ---------- --------- -------- -------------------------------- ------
tech-support      now        00:29      60         30      100     flash:schedule/tech-support/   Success
thelp           12:02:00     00:02      60         40      100     flash:schedule/thelp/
Fail
                06/05/2018
switch(config)#
```

1.1.9.37 show version

The **show version** command displays information that identifies the switch, including its model number, serial number, and system MAC address. The command also provides hardware and software manufacturing information, along with the available memory and elapsed time from the most recent reload procedure.

Command Mode

EXEC

Command Syntax

`show version` [**INFO_LEVEL**]

Parameters

INFO_LEVEL Specifies information the command displays. Options include:

- ***no parameter*** Model and serial numbers, manufacturing data, uptime, and memory.
- **detail** Data listed ***no parameter*** option plus version numbers of internal components.

Related Command

show module displays model and serial numbers of modular system components.

Example

This command displays the switch's model number, serial number, hardware and software manufacturing information, uptime, and memory capacity.

```
switch>show version
Arista DCS-7150S-64-CL-F
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.13.2F
Architecture:           i386
Internal build version: 4.13.2F-1649184.4132F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

Uptime:                 1 hour and 36 minutes
Total memory:           4017088 kB
Free memory:            1473280 kB switch>
```

1.1.9.38 terminal length

The terminal length command overrides automatic pagination and sets pagination length for all show commands on a terminal. If the output of a show command is longer than the configured terminal length, the output will be paused after each screenful of output, prompting the user to continue.

To disable pagination for an SSH session, set terminal length to 0. By default, all console sessions have pagination disabled.

The no terminal length and **`default terminal length`** commands restore automatic pagination by removing the terminal length command from ***running-config***.

The pagination setting is persistent if configured from Global Configuration mode. If configured from EXEC mode, the setting applies only to the current CLI session. Pagination settings may also be overridden when you adjust the size of the SSH terminal window, but can be reconfigured by running the terminal length command again.

Command Mode

EXEC

Command Syntax

`terminal length lines`

`no terminal length`

`default terminal length`

Parameters

lines number of lines to be displayed at a time. Values range from ***0*** through ***32767***. A value of ***0*** disables pagination.

Examples

- This command sets the pagination length for the current terminal session to 10 lines.

```
switch#terminal length 10
Pagination set to 10 lines.
```

- This command configures the switch to paginate terminal output automatically based on screen size for the current terminal session.

```
switch#no terminal length
```

- These commands disable pagination globally.

```
switch#configure
switch(config)#terminal length 0
Pagination disabled.
```

1.1.9.39 terminal monitor

The terminal monitor command enables the display of logging output on the terminal during the current terminal session. This command affects only the local monitor. The no terminal monitor command disables direct monitor display of logging output for the current terminal session.

Command Mode

Privileged EXEC

Command Syntax

`terminal monitor`

`no terminal monitor`

`default terminal monitor`

Example

This command enables the display of logging to the local monitor during the current terminal session.

```
switch#terminal monitor
switch#
```

1.1.9.40 trigger

The **`trigger`** command specifies what event will trigger the event handler. Handlers can be triggered either by the system booting or by a change in a specified interface's IP address or operational status.

To specify the action to be taken when the handler is triggered, use the action bash command.

Command Mode

Event-Handler Configuration

Command Syntax

`trigger` **EVENT**

Parameters

- **EVENT** event which will trigger the configuration mode event handler. Values include:
  - **onboot** triggers when the system reboots, or when you exit event-handler configuration mode. This option takes no further arguments, and passes no environment variables to the action triggered.
    - on-intf **INTERFACE CHANGE** triggers when a change is made to the specified interface.

    - on-startup-config triggers when a change is made to the ***startup-config*** file.
    - vm-tracer vm triggers when a virtual machine monitored by VM Tracer changes state.
- **INTERFACE** the triggering interface. Values include:
    - **ethernet *number*** Ethernet interface specified by ***number***.
    - **loopback *number*** loopback interface specified by ***number***.
    - **management *number*** management interface specified by ***number***.
    - **port-channel *number*** channel group interface specified by ***number***.
    - **vlan *numver*** VLAN interface specified by ***number***.
- **CHANGE** the change being watched for in the triggering interface. Values include:
    - **ip** triggers when the IPv4 address of the specified interface is changed.
    - **ip6** triggers when the IPv6 address of the specified interface is changed.
    - **operstatus** triggers when the operational status of the specified interface changes.

Examples

- This command configures the event handler ***Eth5*** to be triggered when there is a change in the operational status or IP address of ***Ethernet interface 5***.

```
switch(config-handler-Eth5)#trigger on-intf Ethernet 5 operstatus ip
switch(config-handler-Eth5)#
```

- This command configures the event handler “onStartup” to be triggered when the system boots, or on exiting ***event-handler configuration mode***.

```
switch(config-handler-onStartup)#trigger onboot
switch(config-handler-onStartup)#
```

Chapter 2

Booting the Switch

This chapter describes the switch boot process, describes configuration options, and lists the components it requires, including the boot loader, the boot loader shell, and other configuration files.

This chapter includes the following sections:

- Boot Loader Aboot
- Configuration Files
- Supervisor Redundancy
- System Reset
- Aboot Shell
- Aboot Configuration Commands
- Switch Booting Commands

2.1 Boot Loader Aboot

Aboot is the boot loader for Arista switches. In addition to booting the switch EOS, Aboot provides a shell for changing boot parameters, restoring default switch settings, diagnosing hardware problems, and managing switch files. Aboot Shell describes the Aboot shell.

The boot process loads an EOS image file, initiates switch processes, performs self tests, restores interface settings, and configures other network parameters. The replacement image file can be in the switch's flash or on a device in the flash drive port. Configuration files stored in flash memory specify boot parameters.

Aboot supports most available USB flash drive models. The flash drive must be formatted with the FAT or VFAT file system. Windows NT File System (NTFS) is not supported.

Aboot initiates a system reboot upon a **`reload`** command or by restoring power to the switch. Before loading the EOS image file, Aboot provides an option to enter the Aboot shell. The user can either enter the shell to modify boot parameters or allow the switch to boot.

The boot process can be monitored through a terminal connected to the console port. The console port is configured to interact with the terminal by configuration file settings.

2.2 Configuration Files

Three files define boot and running configuration parameters.

- ***boot-config***: contains the location and name of the image to be loaded.
- ***running-config***: contains the current switch configuration.
- ***startup-config***: contains the switch configuration that is loaded when the switch boots.

The ***running-config*** and ***startup-config*** are different from one another when configuration changes have not been saved since the last boot.

2.2.1 boot-config

The ***boot-config*** file is an ASCII file that Aboot uses to configure console communication settings, locate the EOS flash image, and specify initial network configuration settings.

Aboot attempts to boot the EOS flash software image (with the extension .swi) referenced by ***boot-config*** if the user does not interrupt the boot process. See Aboot Shell describes how Aboot uses ***boot-config***.

You can view and edit the ***boot-config*** file contents. Viewing and editing options include:

- View ***boot-config*** file contents with the `more boot-config` command:

```
switch(config)#more boot-config
SWI=flash:/EOS.swi
CONSOLESPEED=2400
Aboot password (encrypted): $1$A8dZ3GLZ$knKrBpTyg5dhmtGdCdwNM.
switch(config)#
```

- View ***boot-config*** settings with the `show boot-config` command:

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 2400
Aboot password (encrypted): $1$A8dZ3GLZ$knKrBpTyg5dhmtGdCdwNM.
Memory test iterations: (not set)
switch(config)#
```

- Modify file settings from the command line with EOS `boot` commands.

  See Programming boot-config from the Programming boot-config from the CLI for a list of `boot` commands.
- Edit the file directly by using vi from the Bash shell.

See boot-config Command Line Content for a list of ***boot-config*** parameters.

2.2.1.1 boot-config File Structure

Each line in the ***boot-config*** file specifies a configuration setting and has this format:

NAME=VALUE

- NAME is the parameter label.
- VALUE indicates the parameter's bootup setting.

The NAME and VALUE fields cannot contain spaces.

Aboot ignores blank lines and lines that begin with a # character.

2.2.1.2 boot-config Command Line Content

Aboot configuration commands that ***boot-config*** files can contain include:

- SWI specifies the location and file name of the EOS image file that Aboot loads when booting, using the same format as the boot command to designate a local or network path.

  Example

```
SWI=flash:EOS.swi             <---flash drive location
SWI=usb1:/EOS1.swi            <---USB drive location
SWI=file:/tmp/EOSexp.swi      <---switch directory location
SWI=/mnt/flash/EOS.swi
SWI=http://foo.com/images/EOS.swi
SWI=ftp://foo.com/images/EOS.swi
SWI=tftp://foo.com/EOS.swi
SWI=nfs://foo.com/images/EOS.swi
```

- CONSOLESPEED specifies the console baud rate. To communicate with the switch, the connected terminal must match the specified rate. Baud rates options include ***1200***, ***2400***, ***4800***, ***9600***, ***19200***, ***38400***, ***57600***, and ***115200***. The default baud rate is ***9600***.

Example

```
CONSOLESPEED=19200
```

- PASSWORD (ABOOT) specifies the Aboot password, as described in Accessing the Aboot Shell. If ***boot-config*** does not contain a **PASSWORD** command, the Aboot shell does not require a password.

  Example

  ```
  PASSWORD=$1$CdWp5wfe$pzNtE3ujBoFEL8vjcq7jo/
  ```

- NET commands are used by Aboot during switch booting to configure the network interface that will be used for switch configuration. These commands can also be entered manually in Aboot.

  NETDEV indicates which network interface is being configured. If ***boot-config*** does not contain a **NETDEV** command, the booting process does not attempt to configure a network interface. Other **NET** commands specify settings that Aboot uses to configure the interface.

Examples

- This **NETDEV** command specifies ***management port 1*** as the network interface to be configured by ***boot-config***:

  ```
  NETDEV=ma1
  ```

- This **NETAUTO** command instructs the switch to configure the network interface through a DHCP server, ignoring other NET settings:

  ```
  NETAUTO=dhcp
  ```

- These **NET** commands configure the network interface:

  ```
  NETIP=10.12.15.10
  NETMASK=255.255.255.0
  NETGW=10.12.15.24
  NETDOMAIN=mycompany.com
  NETDNS=10.12.15.13
  ```

2.2.1.3 Programming boot-config from the CLI

The switch CLI provides **boot** commands for editing ***boot-config*** contents. The **boot** commands are not accessible from a console port CLI. Parameters not configurable from a **boot** command can be modified by directly editing the ***boot-config*** file.

Commands that configure boot parameters include **boot system**, **boot secret**, and **boot console**.

boot system

The **boot system** command provides the EOS image file location to Aboot.

Examples

- This command specifies EOS1.swi on USB flash memory as the software image load file.

  ```
  switch(config)#boot system usb1:EOS1.swi
  switch(config)#
  ```

- The **boot system** command above adds this line to ***boot-config***.

  ```
  SWI=usb1:/EOS1.swi
  ```

- This command designates EOS.swi, on the switch flash, as the EOS software image load file.

```
switch(config)#boot system flash:EOS.swi
switch(config)#
```

- The **boot system** command above adds this line to ***boot-config***:

```
SWI=flash:/EOS.swi
```

boot secret

The **boot secret** command sets the Aboot password.

Examples

- These equivalent commands set the Aboot password to xr19v.

```
switch(config)#boot secret xr19v
switch(config)#
```

```
switch(config)#boot secret 0 xr19v
switch(config)#
```

- This command shows the password that has been set.

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
switch(config)#
```

  The **boot secret** commands above add this line to **boot-config**:

```
PASSWORD=$1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
```

  The user must enter xr19v at the login prompt to access the Aboot shell.
- This command sets the Aboot password to xr123. The encrypted string was previously generated with xr123 as the clear-text seed.

```
switch(config)#boot secret 5 $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
switch(config)#
```

- This command shows the password that has been set.

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
switch(config)#
```

- The **boot secret** command above adds this line to ***boot-config***:

```
PASSWORD=$1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

  The user must enter xr123 at the login prompt to access the Aboot shell.
- This command removes the Aboot password; subsequent Aboot access is not authenticated.

```
switch(config)#no boot secret
switch(config)#
```

- This command shows that there is now no Aboot password.

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): (not set)
switch(config)#
```

boot console

The **boot console** command sets console settings for attaching devices.

Examples

- This command sets the console speed to ***4800*** baud:

```
switch(config)#boot console speed 4800
switch(config)#
```

- This command shows the console speed.

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 4800
Aboot password (encrypted): (not set)
switch(config)#
```

- The **boot console** command above adds this line to ***boot-config***:

```
CONSOLESPEED=4800
```

install bios source

The **install bios source** command loads an Aboot Update File (AUF) to provide a signed method of upgrading Aboot.

Examples

- This command installs the file *update.auf* stored in **/mnt/flash**:

```
switch#install bios source flash:/update.auf
switch(config)#
```

- This command performs a reboot directly after the installation:

```
switch#install bios source flash:/update.auf reload
switch(config)#
```

- This command skips the prompts and performs the installation automatically:

```
switch#install bios source flash:/update.auf now
switch(config)#
```

- This command performs the installation only on the standby suprvisor:

```
switch#install bios source flash:/update.auf standby
switch(config)#
```

Note: On a modular system with two supervisors, Aboot gets upgraded on both by default.

2.2.2 running-config

The ***running-config*** is a virtual file that contains the system's operating configuration, formatted as a command sequence. Commands entered from the CLI modify ***running-config***, and copying a file to ***running-config*** updates the operating configuration by executing the commands in the copied file.

Commands for viewing and copying ***running-config*** include:

- **`show running-config`** displays the contents of ***running-config***.
- **`copy running-config startup-config`** copies ***running-config*** contents to ***startup-config***.
- **`write`** copies ***running-config*** contents to ***startup-config***.

2.2.3 startup-config

The ***startup-config*** file is stored in flash memory and contains the configuration that the switch loads when booting. During a switch boot, ***running-config*** is replaced by ***startup-config***. Changes to ***running-config*** that are not copied to ***startup-config*** are lost when the system reboots.

Commands affecting ***startup-config*** include:

- **`show startup-config`** displays the contents of ***startup-config***.
- **`copy`** ***file_name*** **`startup-config`** copies contents of the specified file to ***startup-config***.
- **`delete startup-config`** deletes the ***startup-config*** file.

2.3 Supervisor Redundancy

On modular switches with redundant supervisor modules, control of the switch can be transferred to the standby supervisor to minimize downtime and data loss in the case of a reset, reload, or failure of the active supervisor. How the switchover takes place is determined by the redundancy protocol configured on the active supervisor.

To display the state and the current redundancy protocol of both supervisors, use the **`show redundancy status`** command. To display the state of configuration file synchronization between the supervisors, use the **`show redundancy file-replication`** command.

2.3.1 Redundancy Supervisor Protocols

There are three available supervisor redundancy protocols.

Route Processor Redundancy (RPR)

The default redundancy protocol is Route Processor Redundancy (RPR), which synchronizes ***startup-config*** files between the supervisor modules and partially boots the standby supervisor to a standby warm state, but does not synchronize ***running-config***. If the active supervisor fails, or a manual switchover is initiated with the **`redundancy manual switchover`** command, the standby supervisor will become active. Running state, including spanning tree, is lost, and all links are temporarily brought down.

Under RPR, the CLI of the standby supervisor can be accessed by SSH or through the console port, but the available command set is limited. Any configuration changes made to the standby supervisor will be lost when the supervisor reboots.

Stateful Switchover (SSO)

In Stateful Switchover (SSO) protocol, the switch synchronizes both ***startup-config*** and ***running-config*** files between the supervisor modules and fully boots the standby module to a standby hot state to speed the switchover process and minimize packet loss. If the active supervisor fails, or a manual switchover is initiated, the standby supervisor immediately becomes active, and L2 running state is maintained. An SSO switchover is largely transparent from the outside, but because L3 state is not synchronized the switchover can result in

traffic loss for traffic forwarded on routes learned by a dynamic routing protocol. Enabling nonstop forwarding can eliminate most packet loss for BGP and OSPF.

Under SSO, the CLI of the standby supervisor can be accessed only through the console port, and the command set is limited. Any configuration changes made on the standby supervisor will be lost when the supervisor reboots.

Note: When upgrading the EOS on a dual-supervisor switch to an SSO-capable version (*4.11.0* or higher) from a version that does not support SSO, both supervisors will reset simultaneously, causing several seconds of system downtime.

Simplex

When the switch is set to simplex protocol, the standby supervisor is disabled and switchover will not occur even if the active supervisor fails. Reloading the active supervisor results in system downtime while the supervisor reboots, and the standby supervisor remains disabled. To transfer control of the switch to the standby supervisor, the redundancy protocol must be changed to RPR or SSO.

Under simplex protocol, the CLI of the disabled supervisor can be accessed only through the console port, and the command set is limited. Any configuration changes made on the standby supervisor will be lost when the supervisor reboots.

2.3.2 Configuring Supervisor Redundancy

The supervisor redundancy protocol is configured using the **`protocol`** command in redundancy configuration mode (accessed with the **`redundancy`** command).

Changing the redundancy protocol on the active supervisor resets the standby supervisor regardless of redundancy protocol, and executing the **`write`** command on the active supervisor synchronizes the ***startup-config*** files between supervisors in RPR and SSO modes.

Examples

- These commands display the current redundancy state of the switch and the most recent file synchronization information.

```
switch#show redundancy state
  my state = ACTIVE
peer state = STANDBY WARM
      Unit = Primary
   Unit ID = 1

Redundancy Protocol (Operational) = Route Processor Redundancy
Redundancy Protocol (Configured) = Route Processor Redundancy
Communications = Up
Ready for switchover

  Last switchover time = 7:23:56 ago
Last switchover reason = Supervisor has control of the active supervisor
 lock switch#show redundancy file-replication
0 files unsynchronized, 2 files synchronized, 0 files failed, 2 files total.

File                   Status         Last Synchronized
---------------------- -------------- -------------------
file:persist/sys       Synchronized   0:10:04 ago
flash:startup-config   Synchronized   0:10:04 ago switch#
```

- These commands set the redundancy protocol for the active supervisor to stateful switchover (SSO).

```
switch#config
switch(config)#redundancy
```

```
switch(config-redundancy)#protocol sso
Peer supervisor will be restarted.
switch(config-redundancy)#
```

2.4 System Reset

When a reset condition exists, Aboot can either reset the switch without user intervention or facilitate a manual reset through the Aboot shell. A reset operation clears the switch, including memory states and other hardware logic

- **Fixed systems**: the power supply remains powered up through the reset. Power is removed from all other switch components for two to five seconds.
- **Modular systems**: the power supply on the active supervisor remains powered up through the reset. Power is removed from all other supervisor components for at least one second. In Stateful Switchover (SSO) and Route Processor Redundancy (RPR) modes, resetting the standby supervisor has no effect on the active supervisor, but resetting the active supervisor causes the standby supervisor to immediately become active. After the supervisor becomes functional, it manages the power-cycling of all line cards.

The **reload** command initiates an immediate reset, terminating all CLI instances not running through the console port. The console port CLI displays messages that the switch generates during a reset. On modular switches with redundant supervisors, CLI sessions on the standby supervisor are not terminated.

The **reload (scheduled)** command schedules a reset operation to initiate at a specific time or after a specified period.

2.4.1 Typical Reset Sequence

The **reload** command power cycles the switch, then resets it under Aboot control. The hard reset clears the switch, including memory states and other hardware logic.

By default, the **reload** command triggers a request to store unsaved ***running-config*** commands and an option to open the Aboot shell before starting the reboot when accessing the CLI through the console port. The switch then begins the reboot process controlled by Aboot.

The following procedure is an example of a typical restart.

1. Begin the reboot process by typing the **reload** command:

   ```
   switch#reload
   ```

   The switch sends a message to confirm the reload request:

   ```
   Proceed with reload? [confirm]
   ```

2. Press **enter** or type **y** to confirm the requested reload. Pressing any other key terminates the reload operation.

   The switch sends a series of messages, including a notification that a message was broadcast to all open CLI instances, informing them that the system is being rebooted. The reload pauses when the CLI displays the Aboot shell notification line.

   ```
   Broadcast message from root@mainStopping sshd: [ OK ]
   SysRq : Remount R/O
   Restarting system

   Aboot 1.9.0-52504.EOS2.0

   Press Control-C now to enter Aboot shell
   ```

3. To continue the reload process, do nothing. Typing **Ctrl-C** opens the Aboot shell; see **Aboot Commands** for Aboot editing instructions.
4. The switch continues the reset process, displaying messages to indicate the completion of individual tasks. The reboot is complete when the CLI displays a login prompt.

```
Booting flash:/EOS.swi
Unpacking new kernel
Starting new kernel
Switching to rooWelcome to Arista Networks EOS 4.4.0
Mounting filesystems: [ OK ]
Entering non-interactive startup
Starting EOS initialization stage 1: [ OK ]
ip6tables: Applying firewall rules: [ OK ]
iptables: Applying firewall rules: [ OK ]
iptables: Loading additional modules: nf_conntrack_tftp [ OK ]
Starting system logger: [ OK ]
Starting system message bus: [ OK ]
Starting NorCal initialization: [ OK ]
Starting EOS initialization stage 2: [ OK ]
Starting ProcMgr: [ OK ]
Completing EOS initialization: [ OK ]
Starting Power On Self Test (POST): [ OK ]
Generating SSH2 RSA host key: [ OK ]
Starting isshd: [ OK ]
Starting sshd: [ OK ]
Starting xinetd: [ OK ]
[ OK ] crond: [ OK ]

switch login:
```

5. Log in to the switch to resume configuration tasks.

2.4.2 Switch Recovery

Aboot can automatically erase the internal flash and copy the contents of a USB drive that has been inserted before powering up or rebooting the switch. This recovery method does not require access to the switch console or Aboot password entry, even if the ***boot-config*** file lists one.

Aboot invokes the recovery mechanism only if each of these two conditions is met:

- The USB drive must contain a file called “fullrecover.”
- The file’s contents are ignored; an empty text file is sufficient.
- If the USB drive contains a ***boot-config*** file, its timestamp must differ from the timestamp of the ***boot-config*** file on the internal flash.

  This prevents Aboot from invoking the recovery mechanism again on every boot if you leave the flash key inserted.

To use this recovery mechanism, set up a USB drive with the files to be installed on the internal flash for example, a current EOS software image, and a customized or empty ***boot-config*** plus an empty file named “fullrecover.”

Check that the timestamp of ***boot-config*** is current to ensure that the above conditions are met.

2.4.3 Display Reload Cause

The **`show reload cause`** command displays the cause of the most recent system reset and lists recommended actions, if any exist, to avoid future spontaneous resets or resolve other issues that may have cause the reset.

Example

To display the reset cause, type **show reload cause** at the prompt.

```
switch#show reload cause
Reload Cause 1:
-------------------
Reload requested by the user.

Recommended Action:
-------------------
No action necessary.

Debugging Information:
----------------------
None available.
switch#
```

2.4.4 Configuring Zero Touch Provisioning

Zero Touch Provisioning (ZTP) is a switch configuration method that uses files referenced by a DCHP server to initially provision the switch without user intervention. A switch enters ZTP mode when it is reloaded if flash memory does not contain a ***startup-config***.

Canceling ZTP boots the switch without using a ***startup-config*** file. When ZTP mode is canceled, a ***startup-config*** file is not stored to flash memory. Until a ***startup-config*** file is stored to flash, the switch returns to ZTP mode on subsequent reboots. This section describes steps required to implement, monitor, and cancel ZTP.

2.4.4.1 Configuring the Network for ZTP

A switch performs the following after booting in ZTP mode:

- Configures each physical interface to **no switchport** mode.
- Sends a DHCP query packet on all Ethernet and management interfaces.

After the switch receives a DHCP offer, it responds with a DHCP request for ***Option 66*** (TFTP server name), ***Option 67*** (bootfile name), and dynamic network configuration settings. When the switch receives a valid DHCP response, it configures the network settings, then fetches the file from the location listed in ***Option 67***. If ***Option 67*** returns a network URL (**http:// or ftp://**), the switch obtains the file from the network. If ***Option 67*** returns a file name, the switch retrieves the file from the TFTP server listed in ***Option 66***.

The ***Option 67*** file can be a ***startup-config*** file or a boot script. The switch distinguishes between a ***startup-config*** file and a boot script by examining the first line in the file:

- The first line of a boot file must consist of the #! characters followed by the interpreter path. The switch executes the code in the script, then reboots. The boot script may fetch an EOS software image or perform required customization tasks.

  The following boot file fetches an EOS software image and stores a startup configuration file to flash.

  ```
  #!/usr/bin/Cli -p2
  copy http://company.com/startup-config flash:startup-config
  copy http://company.com/EOS-2.swi flash:EOS-2.swi
  config
  boot system flash:EOS-2.swi
  ```

- The switch identifies any other file as a ***startup-config*** file. The switch copies the ***startup-config*** file into flash as **mnt/flash/startup-config**, then reboots.

The switch uses its system MAC address as the DHCP client identifier and **Arista** as the Vendor Class Identifier (***Option 60***). When the switch receives an http URL through ***Option 67***, it sends the following HTTP headers in the GET request:

```
X-Arista-SystemMAC:
X-Arista-HardwareVersion:
X-Arista-SKU:
X-Arista-Serial:
X-Arista-Architecture:
```

2.4.4.2 Monitoring ZTP Progress

A switch displays the following message after rebooting when it does not contain a ***startup-config*** file:

```
No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to
download the startup-config from a remote system. The device will not
be fully functional until either a valid startup-config is downloaded
from a remote system or Zero Touch Provisioning is cancelled. To cancel
Zero Touch Provisioning, login as admin and type 'zerotouch cancel'
at the CLI.

switch login:
```

The switch displays a CONFIG_DOWNLOAD_SUCCESS message after it successfully downloads a ***startup-config*** file, then continues the reload process as described in **Typical Reset Sequence**.

```
===============================================================================

Successful download
--------------------


Apr 15 21:36:46 switch ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on
 [
Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9,
Management1, Management2 ]
Apr 15 21:36:56 switch ZeroTouch: %ZTP-5-DHCP_SUCCESS: DHCP response received
 on
Ethernet24  [ Mtu: 1500; Ip Address: 10.10.0.4/16; Nameserver: 10.10.0.1;
 Domain:
aristanetworks.com; Gateway: 10.10.0.1; Boot File:
http://10.10.0.2:8080/tmp/172.17.11.196-startup-config.1 ]
Apr 15 21:37:01 switch ZeroTouch: %ZTP-5-CONFIG_DOWNLOAD: Attempting to
 download
the startup-config from http://10.10.0.2:8080/tmp/172.17.11.196-startup-conf
ig.1
Apr 15 21:37:02 switch ZeroTouch: %ZTP-5-CONFIG_DOWNLOAD_SUCCESS: Successfully
downloaded startup-config from
http://10.10.0.2:8080/tmp/172.17.11.196-startup-config.1
Apr 15 21:37:02 switch ZeroTouch: %ZTP-5-RELOAD: Rebooting the system
Broadcast messagStopping sshd: [  OK  ]
watchdog is not running
SysRq : Remount R/O
Restarting system


Aboot 1.9.0-52504.EOS2.0
```

```
Press Control-C now to enter Aboot shell
```

2.4.4.3 ZTP Failure Notification

The switch displays a **DHCP_QUERY_FAIL** message when it does not receive a valid DHCP response within 30 seconds of sending the query. The switch then sends a new DHCP query and waits for a response. The switch continues sending queries until it receives a valid response or until ZTP mode is canceled.

```
switch login:admin
admin
switch>Apr 15 21:28:21 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP
request on  [ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18,
Ethernet21, E-thernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8,
Ethernet9, Management1, Management2 ]
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-DHCP_QUERY_FAIL: Failed to get a
valid DHCP response
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-RETRY: Retrying Zero Touch
Provisioning from the begining (attempt 1)
Apr 15 21:29:22 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on

[ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9,
Management1, Management2 ]
```

2.4.4.4 Canceling ZTP Mode

To boot the switch without a ***startup-config*** file, log into the console, then cancel ZTP mode. After the switch boots, it uses all factory default settings. A ***startup-config*** file must be saved to flash memory to prevent the switch from entering ZTP mode on subsequent boots.

2.4.5 Configuring the Networks

If the ***boot-config*** file contains a NETDEV statement, Aboot attempts to configure the network interface, as specified by Network configuration commands. See **boot-config Command Line Content** for a list of commands that define the network configuration.

2.5 Aboot Shell

The Aboot shell is an interactive command-line interface used to manually boot a switch, restore the internal flash to its factory-default state, run hardware diagnostics, and manage files. The Aboot shell is similar to the Linux Bourne Again Shell (Bash).

The Aboot shell provides commands for restoring the state of the internal flash to factory defaults or a customized default state. You can use these recovery methods to:

- restore the factory-default flash contents before transferring the switch to another owner.
- restore Aboot shell access if the Aboot password is lost or forgotten.
- restore console access if baud rate or other settings are incompatible with the terminal.
- replace the internal flash contents with configuration or image files stored on a USB flash drive.

2.5.1 Aboot Shell Operation

When the switch is powered on or rebooted, Aboot reads its configuration from ***boot-config*** on the internal flash and attempts to boot an EOS software image (with the extension **.swi**) automatically if one is configured.

You can monitor the automatic boot process or enter the Aboot shell only from the console port. You can connect a PC or terminal directly to the port and run a terminal emulator to interact with the serial port or access it through a serial concentrator device.

Console settings are stored in ***boot-config***; the factory-default settings for Arista switches are ***9600*** baud, no parity, ***8*** character bits, and ***1*** stop bit. If you do not know the current settings, perform a full flash recovery to restore the factory-default settings. When the console port is connected and the terminal settings are configured properly, the terminal displays a message similar to the following a few seconds after powering up the switch:

```
Aboot 1.0.0

Press Control-C now to enter the Aboot shell
```

To abort the automatic boot process and enter the Aboot shell, press **Ctrl-C** (ASCII 3 in the terminal emulator) after the *Press Control-C now to enter Aboot shell* message appears. Pressing **Ctrl-C** can interrupt the boot process up through the starting of the new kernel.

If the ***boot-config*** file does not contain a password command, the Aboot shell starts immediately. Otherwise, you must enter the correct password at the password prompt to start the shell. If you enter the wrong password three times, Aboot displays this message:

```
Type "fullrecover" and press Enter to revert /mnt/flash to factory default

state, or just press Enter to reboot:
```

- Pressing **Enter** continues a normal soft reset without entering the Aboot shell.
- Typing **`fullrecover`** and pressing **Enter** performs a full flash recovery to restore the factory-default settings, removing all previous contents of the flash drive.

The Aboot shell starts by printing:

```
Welcome to Aboot.
```

Aboot then displays the **`Aboot#`** prompt.

Aboot reads its configuration from ***boot-config*** on the internal flash.

2.5.2 Accessing the Aboot Shell

This procedure accesses the Aboot Shell.

1. Reload the switch and press **enter** or type **y** when prompted, as described in Typical Reset Sequence.
   The command line displays this Aboot entry prompt.

   ```
   Press Control-C now to enter Aboot shell
   ```

2. Type **Ctrl-C**.
   If the ***boot-config*** file does not contain a PASSWORD command, the CLI displays an Aboot welcome banner and prompt.

   ```
   ^CWelcome to Aboot.
   Aboot#
   ```

   If the ***boot-config*** file contains a PASSWORD command, the CLI displays a password prompt. In this case, proceed to step 3. Otherwise, the CLI displays the Aboot prompt.
3. If prompted, enter the Aboot password.

   ```
   Press Control-C now to enter Aboot shell
    ^CAboot password:
   ```

```
Welcome to Aboot.
Aboot#
```

Aboot allows three attempts to enter the correct password. After the third attempt, the CLI prompts the user to either continue the reboot process without entering the Aboot shell or to restore the flash drive to the factory default state.

```
Press Control-C now to enter Aboot shell
^CAboot password:
incorrect password
Aboot password:
incorrect password
Aboot password:
incorrect password
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
state, or just press Enter to reboot: fullrecover
All data on /mnt/flash will be erased; type "yes" and press Enter to
 proceed,
or just press Enter to cancel:
```

The **`fullrecover`** operation replaces the flash contents with a factory default configuration. The CLI displays text similar to the following when performing a fullrecover, finishing with another entry option into the Aboot shell.



Note: For hardware that is purchased after June 2017, the factory default partition will not have the backup EOS software image. This is done to increase the flash size on smaller flash size disks, and other options are available in the **`fullrecover`** command functionality to restore factory default EOS image. This is applicable to both fixed system and modular system hardware.

```
Erasing /mnt/flash
Writing recovery data to /mnt/flash
boot-config
startup-config
EOS.swi
210770 blocks
Restarting system.


Aboot 1.9.0-52504.EOS2.0


Press Control-C now to enter Aboot shell
```

2.5.3 Aboot File Structure

When you enter the Aboot CLI, the current working directory is the root directory on the switch. Switch image and configuration files are at **/mnt/flash.** When exiting the Aboot shell, only the contents of **/mnt/flash** are preserved. The **/mnt** directory contains the file systems of storage devices. Aboot mounts the internal flash device at **/mnt/flash**.

When a USB flash drive is inserted in one of the flash ports, Aboot mounts its file system on **/mnt/usb1**. The file system is unmounted when the USB flash drive is removed from the port. Most USB drives contain an LED that flashes when the system is accessing it; do not remove the drive from the flash port until the LED stops flashing.

2.5.4 Booting From the Aboot Shell

Aboot attempts to boot the EOS software image (with the extension .swi) configured in ***boot-config*** automatically if you take no action during the boot process. If the boot process fails for any reason, such as

an incorrectly configured software image, Aboot enters the shell, allowing you to correct the configuration or boot a software image manually. The **boot** command loads and boots an EOS software image file.

The **boot** command syntax is

boot ***SWI***

where ***SWI*** lists the location of the EOS image that the command loads. ***SWI*** options include:

- ***device:path*** loads the image file from the specified storage device. The default ***device*** value is **flash**; other values include **file** and **usb1**.
- */PATH* loads the image file from the specified path in the switch directory.
- **http://*server/path*** loads the image file from the HTTP server on the host server.
- **ftp://*server/path*** loads the image file from the FTP server on the host server.
- **tftp://*server/path*** loads the image file from the TFTP server on the host server.
- **nfs://*server/path*** mounts the path's parent directory from the host server and loads the image file from the loaded directory.

The accepts the same commands as the SWI variable in the boot-config file. See **boot-config Command Line Content** for a list of boot command formats.

If an image file is not specified in ***boot-config***, or if booting the image results in an error condition (for example, an incorrect path or unavailable HTTP server), Aboot halts the boot process and drops into the shell.

Example

To boot EOS.swi from internal flash, enter one of these commands on the Aboot command line:

```
boot flash:EOS.swi
```

```
boot /mnt/flash/EOS.swi
```

2.5.5 Aboot Commands

To list the contents of the internal flash, enter **ls /mnt/flash** at the Aboot# prompt.

Example

```
Aboot#ls /mnt/flash
EOS.swi boot-config startup-config
```

Commonly used commands include:

- **ls** prints a list of the files in the current working directory.
- **cd** changes the current working directory.
- **cp** copies a file.
- **more** prints the contents of a file one page at a time.
- **vi** edits a text file.
- **boot** boots a software image file.
- **swiinfo** prints information about a software image.
- **recover** recovers the factory-default configuration.
- **reboot** reboots the switch.
- **udhcpc** configures a network interface automatically via DHCP.
- **ifconfig** prints or alters network interface settings.
- **wget** downloads a file from an HTTP or FTP server.
- **showtech** prints device hardware information.

Many Aboot shell commands are provided by Busybox, an open-source implementation of UNIX utilities. Busybox command help is found at **http://www.busybox.net/downloads/BusyBox.html**. Aboot provides access to only a subset of the documented commands.

Aboot can access networks through the Ethernet management ports. Aboot provides network interfaces mgmt1 and mgmt2. These ports are unconfigured by default; you can configure management port settings using Aboot shell commands like **`ifconfig`** and **`udhcpc`**. When a management interface is configured, use **`wget`** to transfer files from an HTTP or FTP server, **`tftp`** to transfer files from a TFTP server, or **`mount`** to mount an NFS filesystem.

2.6 Aboot Configuration Commands

This section describes the Aboot configuration commands that a ***boot-config*** file can contain.

- CONSOLESPEED
- NET commands

Installation Commands (Aboot)

- install bios source
- PASSWORD (ABOOT)
- SWI

2.6.1 CONSOLESPEED

The **`CONSOLESPEED`** command in the ***boot-config*** file specifies the console baud rate when the switch is booted. To communicate with the switch, the connected terminal must match the specified rate. Baud rate options include ***1200***, ***2400***, ***4800***, ***9600***, ***19200***, ***38400***, ***57600***, and ***115200***.

The default baud rate is ***9600***.

Command Syntax

`CONSOLESPEED=` *baud_rate*

Parameters

baud_rate specifies the console speed. Values include **1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600**, and **115200**.

Example

This command in a ***boot-config*** file sets the baud rate to 2400 when the switch boots.

```
CONSOLESPEED=2400
```

2.6.2 NET Commands

`NET` commands in the ***boot-config*** file are used by Aboot during switch booting to configure the network interface that will be used for switch configuration. These commands can also be entered manually in Aboot.

`NETDEV` indicates which network interface is being configured. If ***boot-config*** does not contain a **`NETDEV`** command, the booting process does not attempt to configure a network interface. Other **`NET`** commands specify settings that Aboot uses to configure the interface.

Command Syntax

`NETDEV=`*device_interface*

`NETAUTO=`*auto_setting*

`NETIP=`*interface_address*

`NETMASK=`*interface_mask*

`NETGW=`*gateway_address*

`NETDOMAIN=`*domain_name*

`NETDNS=`*dns_address*

Parameters

- ***device_interface*** the network interface. Options include:
  - **ma1** management ***port 1***.
  - **ma2** management ***port 2***.
- ***auto_setting*** the configuration method. Options include:
  - **dhcp** interface is configured through a DHCP server; other NET commands are ignored.
  - if the **NETAUTO** command is omitted, the interface is configured with other NET commands.
- ***interface_address*** interface IP address, in dotted-decimal notation.
- ***interface_mask*** interface subnet mask, in dotted-decimal notation.
- ***gateway_address*** default gateway IP address, in dotted decimal notation.
- ***domain_name*** interface domain name.
- ***dns_address*** IP address of the Domain Name Server, in dotted decimal notation.

Example

- This command specifies management ***port 1*** as the network interface to be configured for management traffic.

```
NETDEV=ma1
```

- This command instructs the switch to configure the network interface through a DHCP server, ignoring any other **NET** commands.

```
NETAUTO=dhcp
```

- These **NET** commands configure the network interface.

```
NETIP=10.12.15.10
NETMASK=255.255.255.0
NETGW=10.12.15.24
NETDOMAIN=mycompany.com
NETDNS=10.12.15.13
```

2.6.3 install bios source

install bios source command is used to install an Aboot Update File (AUF) to be used to upgrade Aboot firmware.

install bios source indicates the location of the source file and the actions to be taken.

Command Syntax

`install bios source` *SOURCE* [active | standby] [reload | now]

Parameters

SOURCE the file location.

Example

- This command installs an AUF file *update.auf* stored in */mnt/flash*.

```
switch#install bios source flash:/update.auf
```

Note: Follow the prompts after executing the command during the installation process.

- This command performs a reboot directly after installation.

```
switch#install bios source flash:/update.auf reload
```

- This command skips the prompts and performs the reboot automatically after installation.

```
switch#install bios source flash:/update.auf now
```

- This command performs the installation only on the standby supervisor.

```
switch#install bios source flash:/update.auf standby
```

Note: By default the active supervisor is upgraded followed by the standby where applicable.

2.6.4 PASSWORD (ABOOT)

PASSWORD specifies the Aboot password, as described in **Accessing the Aboot Shell**. If ***boot-config*** does not contain a **PASSWORD** command, the Aboot shell does not require a password.

The ***boot-config*** file stores the password as an MD5-encrypted string as generated from a clear-text seed by the UNIX passwd program or the crypt library function. When entering the Aboot password, the user types the clear-text seed.

There is no method of recovering the password from the encrypted string. If the clear-text password is lost, delete the corresponding **PASSWORD** command line from the ***boot-config*** file.

Note: The EOS boot secret command is the recommended method for adding or modifying the **PASSWORD** configuration line.

Command Syntax

PASSWORD = *encrypted_string*

Parameters

encrypted_string the encrypted string that corresponds to the clear-text Aboot password.

Example

This line is a **PASSWORD** command example in which the encrypted string corresponds to the clear-text password ***abcde***.

```
PASSWORD=$1$CdWp5wfe$pzNtE3ujBoFEL8vjcq7jo/
```

2.6.5 SWI

SWI specifies the location and file name of the EOS image file that Aboot loads when booting, using the same format as ***boot-config*** to designate a local or network path.

Command Syntax

SWI = **SWI_PATH**

Parameters

SWI_PATH specifies the location of the EOS image file. Options include:

- ***file_path*** specifies a file location on the internal flash drive.
- **file**:***file_path*** specifies a file location in the switch directory.
- **usb1**:***file_path*** specifies a file location on the USB drive.
- **http**:***file_path*** specifies an HTTP path to the image file.
- **ftp**:***file_path*** specifies an FTP path to the image file.
- **tftp**:***file_path*** specifies a TFTP path to the image file.
- **nfs**:***file_path*** specifies an NFS path to the image file.

Examples

- This SWI command in ***boot-config*** causes Aboot to boot the switch from the file **`EOS.swi`** on the switch's internal flash drive.

```
SWI=flash:EOS.swi
```

- This SWI command in ***boot-config*** causes Aboot to boot the switch from the file **`EOS.swi`** at the specified location on foo.com.

```
SWI=http://foo.com/images/EOS.swi
```

2.7 Switch Booting Commands

- boot console
- boot secret
- boot system
- delete startup-config
- protocol
- redundancy
- redundancy manual switchover
- reload
- reload (scheduled)
- show platform bios
- show redundancy file-replication
- show redundancy status
- show redundancy switchover sso
- show reload
- show reload cause
- show reload fast-boot

2.7.1 boot console

The **`boot console`** command configures terminal settings for serial devices connecting to the console port. At this time, the only console setting that can be specified from ***boot-config*** is **speed**.

Factory-default console settings are ***9600*** baud, no parity, ***8*** character bits, and ***1*** stop bit. If you do not know the current settings, restore the factory-default settings as described in Restoring the Factory Default EOS Image and Startup Configuration.

The **`no boot console`** and **`default boot console`** commands restore the factory default settings on the switch and remove the corresponding **`CONSOLESPEED`** command from the ***boot-config*** file.

Command Mode

Global Configuration

Command Syntax

`boot console speed` *baud_rate*

Parameters

baud_rate console baud rate. Options include ***1200***, ***2400***, ***4800***, ***9600***, ***19200***, ***38400***, ***57600***, and ***115200***.

Example

- This command sets the console speed to ***57600*** baud.

```
switch(config)#boot console speed 57600
switch(config)#
```

- This command displays the result of the console-speed change.

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 57600
Aboot password (encrypted): (not set)
switch(config)#
```

- The above **`boot console`** command adds the following line to ***boot-config***.

```
CONSOLESPEED=57600
```

2.7.2 boot secret

The **`boot secret`** command creates or edits the Aboot shell password and stores the encrypted string in the **`PASSWORD`** command line of the ***boot-config*** file.

The **`no boot secret`** and **`default boot secret`** commands remove the Aboot password from the ***boot-config*** file. When the Aboot password does not exist, no password is required to enter the Aboot shell.

Command Mode

Global Configuration

Command Syntax

`boot secret` [**ENCRYPT_TYPE**] *password*

`no boot secret`

`default boot secret`

Parameters

- **ENCRYPT_TYPE** indicates the encryption level of the password parameter. Settings include:
  - ***no parameter*** the password is clear text.
  - **0** the password is clear text. Equivalent to the ***no parameter*** case.
  - **5** the password is an MD5-encrypted string.
  - **sha512** the password is entered as an **sha512** encrypted string.
- ***password*** specifies the boot password.
  - ***password*** must be in clear text if **ENCRYPT_TYPE** specifies clear text.
  - ***password*** must be an appropriately encrypted string if **ENCRYPT_TYPE** specifies encryption.

Restrictions

The **sha512** encryption option is not available on Trident platform switches.

Examples

- These equivalent commands set the Aboot password to xr19v.

```
switch(config)#boot secret xr19v
switch(config)#
```

```
switch(config)#boot secret 0 xr19v
switch(config)#
```

This command displays the result:

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
switch(config)#
```

The **boot secret** commands above add this line to ***boot-config***:

```
PASSWORD=$1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
```

The user must enter **xr19v** at the login prompt to access the Aboot shell.

- These commands set the Aboot password to ***xr123***, then display the resulting ***boot-config*** code. The encrypted string was previously generated with ***xr123*** as the clear-text seed.

```
switch(config)#boot secret 5 $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
switch(config)#
```

The **boot secret** command above adds this line to ***boot-config***:

```
PASSWORD=$1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

The user must enter ***xr123*** at the login prompt to access the Aboot shell.

- This command removes the Aboot password, allowing access to the Aboot shell without a password.

```
switch(config)#no boot secret
switch(config)#
```

2.7.3 boot system

The **boot system** command specifies the location of the EOS software image that Aboot loads when the switch boots. The command can refer to files on flash or on a module in the USB flash port.

Command Mode

Global Configuration

Command Syntax

`boot system` **DEVICE** *file_path*

Parameters

- **DEVICE** location of the image file. Options include:
  - **file:** file is located in the switch file directory.
  - **flash:** file is located in flash memory.

- **usb1:** file is located on a drive inserted in the USB flash port. Available if a drive is in the port.
- ***file_path*** path and name of the file.

Examples

- This command designates ***EOS1.swi***, on USB flash memory, as the EOS software image load file.

```
switch(config)#boot system usb1:EOS1.swi
switch(config)#
```

The **boot system** command above adds this line to ***boot-config***.

```
SWI=usb1:/EOS1.swi
```

- This command designates ***EOS1.swi***, on the switch flash, as the EOS software image load file.

```
switch(config)#boot system flash:EOS.swi
switch(config)#
```

The **boot system** command above adds this line to ***boot-config***:

```
s
```

2.7.4 delete startup-config

The `delete startup-config` command erases or deletes the ***startup-config*** file.

Command Mode

Privileged EXEC

Command Syntax

`delete startup-config` [**CONFIRMATION**]

Parameters

CONFIRMATION confirmation for immediate erasure. Options include:

- ***no parameter*** the switch requires a confirmation before starting the erasure.
- **now** the erasure begins immediately without prompting the user to confirm the request.

Examples

- This command deletes the startup configuration from the switch. When the **delete startup-config** command is entered, the switch sends a message prompting the user to confirm the request.

```
switch#delete startup-config
Proceed with erasing startup configuration? [confirm]y
switch#
```

- This command deletes the startup configuration from the switch immediately without prompting.

```
switch#delete startup-config now
switch#
```

2.7.5 protocol

The **protocol** command configures how the supervisors on a modular switch will handle switchover events. By default, the switch is set to **route processor redundancy (RPR)**, which synchronizes ***startup-config*** files between the supervisor modules and partially boots the standby supervisor. The mode can also be set to **simplex** (manual switchover only) or to **stateful switchover (SSO)** which synchronizes both ***startup-config*** and ***running-config*** files between the supervisor modules and fully boots the standby module to speed the

switchover process and minimize packet loss. Note that SSO synchronizes L2 state between the supervisors, but that L3 state is not synchronized. This can result in traffic loss for traffic forwarded on routes learned by a dynamic routing protocol. Enabling nonstop forwarding can eliminate most packet loss for BGP and OSPF.

The **no protocol** and **default protocol** commands set the redundancy protocol to the default value (**rpr**) by removing the **protocol** command from ***running-config***.

Command Mode

Redundancy Configuration

Command Syntax

`protocol` {rpr | simplex | sso}

`no protocol`

`default protocol`

Parameters

- **rpr** route processor redundancy protocol (RPR, the default).
- **simplex** no redundancy. Switchover must be initiated manually.
- **sso** stateful switchover (SSO).

Related Commands

redundancy places switch in redundancy configuration mode.

Example

These commands enter redundancy configuration mode and set the redundancy protocol to SSO.

```
switch(config)#redundancy
switch(config-redundancy)#protocol sso
switch(config-redundancy)#
```

2.7.6 redundancy

The **redundancy** command places the switch in redundancy configuration mode.

Command Mode

Global Configuration

Command Syntax

`redundancy`

Commands Available in Redundancy Configuration Mode

protocol

Related Commands

redundancy manual switchover manually initiates a switchover.

Example

These commands enter redundancy configuration mode and set the redundancy protocol to stateful switchover.

```
switch(config)#redundancy
switch(config-redundancy)#protocol sso
switch(config-redundancy)#
```

2.7.7 redundancy manual switchover

The **redundancy manual switchover** command immediately switches control of the switch to the standby supervisor. If the redundancy mode is set to simplex or the standby supervisor is unavailable for any other reason, this command will not function.

Command Mode

Privileged EXEC

Command Syntax

`redundancy manual switchover`

Related Command

redundancy places the switch in redundancy configuration mode.

Example

This command forces a switchover to the standby supervisor. The switchover is executed immediately without further confirmation from the user.

```
switch#redundancy manual switchover
This supervisor will be restarted.
switch#
```

2.7.8 reload

The **reload** command power cycles the switch, then resets it under Aboot control. The hard reset clears the switch, including memory states and other hardware logic.



Note: The **reload fast-boot** and **reload hitless** commands are used to initiate Smart System Upgrade (SSU); for a description of this feature and the appropriate command syntax, refer to the Smart System Upgrade (Leaf SSU) section.

The command behaves differently in fixed and modular systems.

- **Fixed 1-RU systems**: the power supply remains powered up through the reset. Power is removed from all other switch components for two to five seconds.
- **Modular systems**: the power supply on the active supervisor remains powered up through the reset. Power is removed from all other supervisor components for at least one second. After the supervisor becomes functional, it manages the power-cycling of all line cards.

Command Mode

Privileged EXEC

Command Syntax

`reload` [**TARGET**] [**CONFIRMATION**]

Parameters

- **TARGET** specifies which supervisor(s) will be reset. Some options are available only on dual-supervisor switches. Options include:
  - ***no parameter*** the active supervisor is reset.
  - **all** both supervisors are reset.
  - **peer** the peer supervisor is reset.
  - **power** the active supervisor is reset.
- **CONFIRMATION** confirmation for immediate reset. Options include:
  - ***no parameter*** the switch requires a confirmation before starting the reset.
  - **now** the reset begins immediately without prompting the user to confirm the request.

Related Commands

- **reload (scheduled)** schedules a pending reload operation.
- **show reload cause** displays cause of most recent reload.

Examples

- Begin the reboot process by typing the **reload** command:

```
switch#reload
switch#
```

- When the **reload** command is entered, the switch sends a message prompting the user to save the configuration if it contains unsaved modifications, then asks the user to confirm the reload request. In this example, the user does not save modifications to the system before reloading.

```
System configuration has been modified. Save? [yes/no/cancel/diff]:n
Proceed with reload? [confirm]y
```

- The switch responds by broadcasting a series of messages, including a notification that the system is being rebooted, to all open CLI instances. The reload pauses to provide an option for the user to enter Aboot shell; the Aboot shell supports commands that restore the state of the internal flash to factory defaults or create a customized default state.

```
Broadcast message from root@mainStopping sshd: [  OK  ]
SysRq : Remount R/O
Restarting system

Aboot 1.9.0-52504.EOS2.0

Press Control-C now to enter Aboot shell
```

- No action is required to continue the reset process. The switch displays messages to indicate the completion of individual tasks. The reboot is complete when the CLI displays a login prompt.

```
Booting flash:/EOS.swi
Unpacking new kernel
Starting new kernel
Switching to rooWelcome to Arista Networks EOS 4.4.0
Mounting filesystems:  [  OK  ]
Entering non-interactive startup
Starting EOS initialization stage 1: [  OK  ]
ip6tables: Applying firewall rules: [  OK  ]
iptables: Applying firewall rules: [  OK  ]
iptables: Loading additional modules: nf_conntrack_tftp [  OK  ]
Starting system logger: [  OK  ]
Starting system message bus: [  OK  ]
Starting NorCal initialization: [  OK  ]
Starting EOS initialization stage 2: [  OK  ]
Starting ProcMgr: [  OK  ]
Completing EOS initialization: [  OK  ]
Starting Power On Self Test (POST): [  OK  ]
Generating SSH2 RSA host key: [  OK  ]
Starting isshd: [  OK  ]
Starting sshd: [  OK  ]
Starting xinetd: [  OK  ]
[  OK  ] crond: [  OK  ]

switch login:
```

2.7.9 reload (scheduled)

The **reload (scheduled)** command configures the switch to reset at a specified time or after a specified interval. Refer to **reload** for the functional details of the reset operation.

The switch prompts to save the configuration and confirm the reload request. Once the request is confirmed, the switch resumes normal operation until the reload initiates.

The **reload cancel**, **no reload**, and **default reload** commands cancel the pending reload operation.

Command Mode

Privileged EXEC

Command Syntax

`reload` [all | power | **peer**] {at *hh:mm* [*month* | *day*] | in *hh:mm*} [force | reason]

`reload cancel`

`no reload`

`default reload`

Parameters

- **all** reloads all supervisors.
- **power** resets the active supervisor.
- **peer** resets the peer supervisor.
- **at** performs reload at specified times.
  - ***hh:mm*** specifies reload time in hours (0 to 23) and minutes (0 to 59). If no month and day are specified, reload occurs at the specified time on the day the command is issued.
    - ***month*** optionally specifies month of the year (Jan, Feb, etc.)
      - ***day*** day of the month. Values range from ***1*** to ***31***.
- **in** performs the reload after a specified delay.
  - ***hh:mm*** specifies delay before reload in hours (***0*** to ***23***) and minutes (***0*** to ***59***).
- **force** performs action immediately without prompting.
- **reason** enter text to display explaining the purpose of the reload.
- **cancel** cancels any existing reload request.

Related Commands

- **reload** initiates an immediate reload operation.
- **show reload** displays time and reason of any pending reload operation.

Examples

- This command schedules a switch reset to begin in 12 hours.

```
switch#reload in 12:00
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Proceed with reload? [confirm]
Reload scheduled for Tue Mar 27 05:57:25 2012 (in 11 hours 59 minutes)
switch#
```

- This command cancels a scheduled reset.

```
switch#no reload
Scheduled reload has been cancelled switch#
```

- This command schedules a reload of the active supervisor to begin in 12 hours.

```
switch#reload power in 12:00
Reload scheduled for Thu Feb 13 18:56:01 2020 (in 11 hours 59 minutes)
switch#
```

2.7.10 show platform bios

The **show platform bios [history | detail]** command displays the BIOS versions on the switch.

Command Mode

EXEC

Command Syntax

```
show platform bios
```

Example

- This command displays the Running, Programmed and Fallback versions of Aboot.

```
switch#show platform bios
FixedSystem BIOS versions
  Location   Version
  ---------- ---------------------------------
  Running   Aboot-norcal9-9.0.5-13882759
  Programmed Aboot-norcal9-9.0.3-4core-13882759
  Fallback   Aboot-norcal9-9.0.3-4core-13882759 switch#
```

- This command displays the BIOS installation history.

```
switch#show platform bios history
Supervisor-1 BIOS version history

Timestamp           Version
------------------- ---------------------------------
2020-04-07 04:20:22 Aboot-norcal9-9.0.5-13882759
2020-04-05 08:32:03 Aboot-norcal9-9.0.3-4core-13882759

Supervisor-2 BIOS version history

Last update failed on 2020-04-15: Compatibility check failed.

Timestamp           Version
------------------- ---------------------------------
2020-04-05 08:32:09 Aboot-norcal9-9.0.5-13882759 switch#
```

- This command displays the BIOS installation history details.

```
switch#show platform bios history detail
Supervisor-1 BIOS version history

Timestamp           Version                           Checksum
------------------- --------------------------------- ---------
2020-04-07 04:20:22 Aboot-norcal9-9.0.5-13882759      3fa8...
2020-04-05 08:32:03 Aboot-norcal9-9.0.3-4core-13882759 bb39...

Supervisor-2 BIOS version history

Last update failed on 2020-04-15: Compatibility check failed.

Timestamp           Version                           Checksum
------------------- --------------------------------- ---------
```

```
2020-04-05 08:32:09 Aboot-norcal9-9.0.3-13882759       3fa8...
switch#
```

2.7.11 show redundancy file-replication

The **show redundancy file-replication** command displays the status and last synchronization date of file replication between the supervisors on the switch.

Command Mode

EXEC

Command Syntax

```
show redundancy file-replication
```

Related Commands

- **show redundancy status** displays status and redundancy protocol of supervisors.
- **show redundancy switchover sso** displays stateful switchover information since last reload.

Example

This command displays the current file replication status of the supervisors.

```
switch#show redundancy file-replication
0 files unsynchronized, 2 files synchronized, 0 files failed, 2 files total.

File                    Status         Last Synchronized
----------------------- -------------- -----------------------
file:persist/sys        Synchronized   25 days, 19:48:26 ago
flash:startup-config    Synchronized   25 days, 19:48:26 ago switch#
```

2.7.12 show redundancy status

The **show redundancy status** command displays the current status (active or standby) and the configured redundancy protocol of both supervisors, as well as summary information about the latest switchover event.

Command Mode

EXEC

Command Syntax

```
show redundancy status
```

Related Commands

- **show redundancy file-replication** displays status of file replication between supervisors.
- **show redundancy switchover sso** displays stateful switchover information since last reload.

Example

This command displays redundancy information for both supervisors and a summary of the latest switchover.

```
switch#show redundancy status
  my state = ACTIVE
peer state = STANDBY HOT
      Unit = Primary
   Unit ID = 1

Redundancy Protocol (Operational) = Stateful Switchover
Redundancy Protocol (Configured) = Stateful Switchover
Communications = Up
```

```
switchover completion timeout = 720.0 seconds (default)
Ready for switchover

  Last switchover time = 0:32:15 ago
Last switchover reason = Supervisor has control of the active supervisor lock
 switch#
```

2.7.13 show redundancy switchover sso

The **show redundancy switchover sso** command displays the number of stateful switchovers since the last reload and a log of the events in the latest stateful switchover.

Command Mode

EXEC

Command Syntax

```
show redundancy switchover sso
```

Related Commands

- **show redundancy file-replication** displays status of file replication between supervisors.
- **show redundancy status** displays status and redundancy protocol of supervisors.

Example

This command displays stateful switchover information.

```
switch#show redundancy switchover sso
Total number of Stateful Switchover completed since reload: 4

Latest Stateful Switchover occured 29 days, 12:48:22 ago @ 2012-06-09 19:47:50
(completed)
  0.000000: switchover started
  0.000235: stage PCIEAcquired started
  0.000349:   event PCIEAcquired:__dummyInternal1__ completed
  0.000394:   event PCIEAcquired:PlxPcie-system started
  0.027738:   event PCIEAcquired:PlxPcie-system completed
  0.027829: stage PCIEAcquired is complete
  0.027935: stage DmaReady started
  0.028042:   event DmaReady:ForwardingAgent started
  0.079620:   event DmaReady:ForwardingAgent completed
  0.079699: stage DmaReady is complete
  0.079781: stage TimeCriticalServices started
  0.079887:   event TimeCriticalServices:__dummyInternal1__ completed
  0.079928:   event TimeCriticalServices:Stp started
  0.208035:   event TimeCriticalServices:Stp completed
  0.208120: stage TimeCriticalServices is complete
              <-------OUTPUT OMITTED FROM EXAMPLE------->
 39.675076: stage NonCriticalServices started
 39.675145:   event NonCriticalServices:__dummyInternal1__ completed
 39.675183: stage NonCriticalServices is complete
 39.675399: switchover is complete switch#
```

2.7.14 show reload

The **show reload** command displays the time and reason of any pending reload operation. The **reload (scheduled)** command schedules a reload operation and can be used to cancel a pending reload.

Command Mode

EXEC

Command Syntax

```
show reload
```

Related Commands

- **reload (scheduled)** schedules a pending reload operation.
- **show reload cause** displays cause of most recent reload.

Example

These commands schedule a reload for 2:45 pm, display the time of the pending reload, then cancel the scheduled reload.

```
switch#reload at 14:45
Proceed with reload? [confirm]
Reload scheduled for Tue Mar 27 14:45:00 2012 ( in 4 hours 11 minutes )
switch#show reload
Reload scheduled for Tue Mar 27 14:45:00 2012 ( in 4 hours 11 minutes )
switch#reload cancel
Scheduled reload has been cancelled switch#
```

2.7.15 show reload cause

The **show reload cause** command displays the reason of the most recent reload operation. The command displays recommended actions and debug information related to the executed reload.

Command Mode

EXEC

Command Syntax

```
show reload cause
```

Related Commands

- **reload** initiates an immediate reload operation.
- **show reload** displays time and reason of all pending reload operations.

Example

This command displays the cause of the most recent reload operation.

```
switch#show reload cause
Reload Cause 1:
-------------------
Reload requested by the user.

Recommended Action:
-------------------
No action necessary.

Debugging Information:
----------------------
None available.
switch#
```

2.7.16 show reload fast-boot

The **show reload fast-boot** command verifies that the switch configuration is valid for Smart System Upgrade (SSU).

Command Mode

EXEC

Command Syntax

```
show reload fast-boot
```

Related Commands

- **reload** initiates an immediate reload operation.
- **show reload** displays time and reason of all pending reload operations.

Example

This command displays any reasons why SSU cannot be initiated and which features must be reconfigured to allow SSU to proceed.

```
switch# show reload fast-boot
'reload fast-boot' cannot proceed due to the following:
  Spanning-tree portfast is not enabled for one or more ports
  Spanning-tree BPDU guard is not enabled for one or more ports switch#
```

Chapter 3

Initial Configuration and Recovery

This chapter describes initial configuration and recovery tasks. Subsequent chapters provide details about features introduced in this chapter.

This chapter contains these sections:

- Initial Switch Access
- Connection Management
- Switch Storage Device Secure Erase
- Configure Session
- Recovery Procedures
- Session Management Commands

3.1 Initial Switch Access

Arista network switches provide two initial configuration methods:

- Zero Touch Provisioning (ZTP) configures the switch without user interaction (Zero Touch Provisioning).
- Manual provisioning configures the switch through commands entered by a user through the CLI (Manual Provisioning).

3.1.1 Zero Touch Provisioning

Zero Touch Provisioning (ZTP) configures a switch without user intervention by downloading a startup configuration file (***startup-config***) or a boot script from a location specified by a DHCP server. Configuring the Network for ZTP describes network tasks required to set up ZTP.

The switch enters ZTP mode when it boots if flash memory does not contain ***startup-config***. It remains in ZTP mode until a user cancels ZTP mode, or until the switch retrieves a ***startup-config*** or a boot script. After downloading a file through ZTP, the switch reboots again, using the retrieved file.

Security Considerations

The ZTP process cannot distinguish an approved DHCP server from a rogue DHCP server. For secure provisioning, you must ensure that only approved DHCP servers are able to communicate with the switch until after the ZTP process is complete. Arista also recommends validating the EOS image on your ZTP server by confirming that its MD5 checksum matches the MD5 checksum that can be found on the EOS download page of the Arista website.

On a UNIX server, the **md5sum** command calculates this checksum:

```
% md5sum EOS.swi
3bac45b96bc820eb1d10c9ee33108a25  EOS.swi
```

This command is also available on Arista switches from the CLI or from within the Bash shell.

```
switch#bash md5sum /mnt/flash/EOS-4.18.0F.swi
73435f0db3af785011f88743f4c01abd  /mnt/flash/EOS-4.18.0F.swi

switch#[admin@switch ~]$ md5sum /mnt/flash/EOS-4.18.0F.swi
73435f0db3af785011f88743f4c01abd /mnt/flash/EOS-4.18.0F.swi
```

```
[admin@switch ~]$
```

To provision the switch through Zero Touch Provisioning:

1. Mount the switch in its permanent location.
2. Connect at least one management or Ethernet port to a network that can access the DHCP server and the configuration file.
3. Provide power to the switch.

ZTP provisioning progress can be monitored through the console port. Console Port provides information for setting up the console port. Canceling Zero Touch Provisioning provides information for monitoring ZTP progress and canceling ZTP mode.

3.1.2 Manual Provisioning

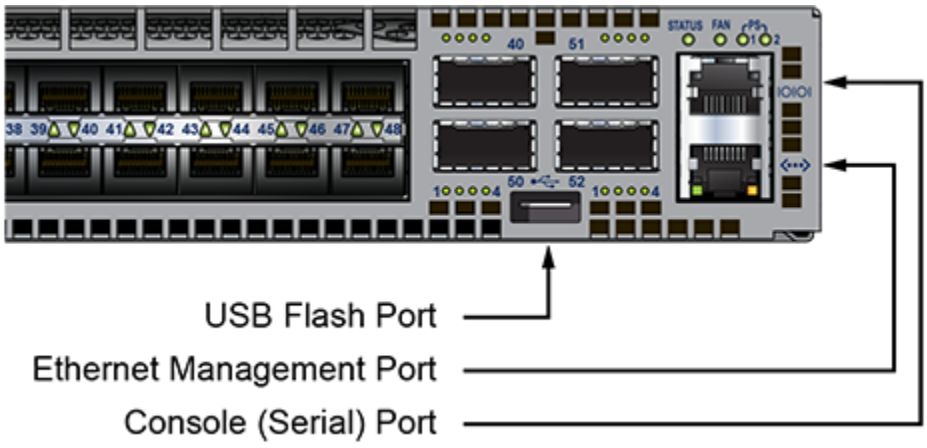
Initial manual switch provisioning requires the cancellation of ZTP mode, the assignment of an IP address to a network port, and the establishment of an IP route to a gateway. Initial provisioning is performed through the serial console and Ethernet management ports.

- The console port is used for serial access to the switch. These conditions may require serial access:
  - management ports are not assigned IP addresses.
  - the network is inoperable.
  - the password for the users log on is not available.
  - the password to access the enable mode is not available.
- The Ethernet management ports are used for out-of-band network management tasks. Before using a management port for the first time, an IP address must be assigned to that port.

3.1.2.1 Console Port

The console port is a serial port located on the front of the switch. Figure 2: Switch Ports shows the console port on the DCS-7050T-64 switch. Use a serial or RS-232 cable to connect to the console port. The accessory kit also includes an RJ-45 to DB-9 adapter cable for connecting to the switch.

Figure 3-1: Switch Ports



Port Settings

Use these settings when connecting the console port:

- 9600 baud
- no flow control
- 1 stop bit

- no parity bits
- 8 data bits

Admin Username

The initial configuration provides one username, **admin**, that is not assigned a password. When using the **admin** username without a password, you can only log into the switch through the console port. After a password is assigned to the **admin** username, it can log into the switch through any port.

The username command modifies a specified username, and can be used to create or delete usernames, including **admin**.

Example

This command assigns the password **pxq123** to the **admin** username:

```
switch(config)#username admin secret pxq123
switch(config)#
```

New and altered passwords must be saved to the startup configuration file or they will be lost when the switch is rebooted.



Note: if the configuration is saved with all usernames deleted, it will be necessary to access the Aboot shell through the console port in order to log in to the switch.

3.1.2.2 Canceling Zero Touch Provisioning

Zero Touch Provisioning (ZTP) installs a ***startup-config*** file from a network location if flash memory does not contain a ***startup-config*** when the switch reboots. Canceling ZTP is required if the switch cannot download a***startup-config*** or boot script file.

When the switch boots without a ***startup-config*** file, it displays the following message through the console port:

```
No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to
download the startup-config from a remote system. The device will not
be fully functional until either a valid startup-config is downloaded
from a remote system or Zero Touch Provisioning is cancelled. To cancel
Zero Touch Provisioning, login as admin and type 'zerotouch cancel'
at the CLI.

localhost login:
```

To cancel ZTP mode, log into the switch with the **admin** password, then enter the **zerotouch cancel** command. The switch immediately boots without installing a ***startup-config*** file.

```
localhost login: admin
admin
localhost>Apr 15 21:28:21 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP
request on  [ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18,
Ethernet21, E-thernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8,
Ethernet9, Management1, Management2 ]
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-DHCP_QUERY_FAIL: Failed to get a
valid DHCP response
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-RETRY: Retrying Zero Touch
Provisioning from the beginning (attempt 1)
Apr 15 21:29:22 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on

[ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
```

```
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9,
Management1, Management2 ]

localhost>zerotouch cancel
zerotouch cancel
localhost>Apr 15 21:29:39 localhost ZeroTouch: %ZTP-5-CANCEL: Canceling Zero
Touch Provisioning
Apr 15 21:29:39 localhost ZeroTouch: %ZTP-5-RELOAD: Rebooting the system
Broadcast messagStopping sshd: [  OK  ]
watchdog is not running
SysRq : Remount R/O
Restarting system

Aboot 1.9.0-52504.EOS2.0
Press Control-C now to enter Aboot shell
```

To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file as described in Step 8 of Ethernet Management Port.

3.1.2.3 Ethernet Management Port

Arista switches provide one or more Ethernet management ports for configuring the switch and managing the network out of band. Figure 2: Switch Ports shows the location of the Ethernet management ports on a DCS-7050T-64 switch. Only one port is required to manage the switch.

You can access the Ethernet management port(s) remotely over a common network or locally through a directly connected PC. Before you can access the switch through a remote connection, an IP address and a static route to the default gateway are required. On a modular switch with dual supervisors, a virtual IP address can also be configured to access the management port on whichever supervisor is active.

Assigning a Virtual IP Address to Access the Active Ethernet Management Port

On modular switches with dual supervisors, this procedure assigns a virtual IP address which will connect to the Ethernet management port of the active supervisor. (To assign a physical IP address to an individual Ethernet management port, see the following section.)

1. Connect a PC or terminal server to the console port. Use the settings listed in Console Port under Port Settings.
2. Type **admin** at the login prompt to log into the switch. Initial login through the console port does not require a password.

```
Arista EOS
switch login:admin
Last login: Fri Apr 9 14:22:18 on Console switch>
```

3. Type **enable** at the command prompt to enter Privileged EXEC mode.

```
switch>enable
switch#
```

4. Type **configure terminal** (or **config**) to enter global configuration mode.

```
switch#configure terminal
switch(config)#
```

5. Type **interface management 0** to enter interface configuration mode for the virtual interface which accesses management port 1 on the currently active supervisor.

```
switch(config)#interface management 0
switch(config-if-Ma0)#
```

6. Type **ip address**, followed by the desired address, to assign a virtual IP address for access to the active management port.This command assigns IP address 10.0.2.5 to management port 0.

```
switch(config-if-Ma0)#ip address 10.0.2.5/24
```

7. Type **exit** at both the interface configuration and global configuration prompts to return to ***Privileged EXEC*** mode.

```
switch(config-if-Ma0)#exit
switch(config)#exit
switch#
```

8. Type **write** (or **copy running-config startup-config)** to save the new configuration to the ***startup-config*** file.

```
switch# write
switch#
```

Assigning an IP Address to a Specific Ethernet Management Port

This procedure assigns an IP address to a specific Ethernet management port:

1. Connect a PC or terminal server to the console port. Use the settings listed in Console Port under Port Settings.
2. Type **admin** at the login prompt to log into the switch. The initial login does not require a password.

```
Arista EOS
switch login:admin
Last login: Fri Apr 9 14:22:18 on Console switch>
```

3. Type **enable** at the command prompt to enter Privileged EXEC mode.

```
switch>enable
switch#
```

4. Type **configure terminal** (or **config**) to enter global configuration mode.

```
switch#configure terminal
```

5. Type **interface management 1** to enter interface configuration mode. (Any available management port can be used in place of management port 1.)

```
switch(config)#interface management 1
switch(config-if-Ma1)#
```

6. Type **ip address**, followed by the desired address, to assign an IP address to the port. This command assigns the IP address 10.0.2.8 to management port 1.

```
switch(config-if-Ma1)#ip address 10.0.2.8/24
```

7. Type **exit** at both the interface configuration and global configuration prompts to return to Privileged EXEC mode.

```
switch(config-if-Ma1)#exit
 switch(config)#exit
switch#
```

8. Type **write** (or **copy running-config startup-config)** to save the new configuration to the ***startup-config*** file.

```
switch# write
```

```
switch#
```

Configuring a Default Route to the Gateway

This procedure configures a default route to a gateway located at 10.0.2.1.

1. Enter global configuration mode.

```
switch>enable
switch#configure terminal
```

2. Create a static route to the gateway with the IP route command.

```
switch(config)#ip route 0.0.0.0/0 10.0.2.1
```

3. Save the new configuration.

```
switch#write
switch#
```

3.2 Connection Management

The switch supports three connection methods:

- console
- SSH
- Telnet

The switch always enables console and SSH. Telnet is disabled by default.

Management commands place the switch in a configuration mode for changing session connection parameters.

Examples

- The management console command places the switch in console management mode:

```
switch(config)#management console
switch(config-mgmt-console)#
```

- The management ssh command places the switch in SSH management mode:

```
switch(config)#management ssh
switch(config-mgmt-ssh)#
```

- The management telnet command places the switch in Telnet management mode:

```
switch(config)#management telnet
switch(config-mgmt-telnet)#
```

- The **exit** command returns the switch to global configuration mode:

```
switch(config-mgmt-ssh)#exit
switch(config)#
```

The **idle-timeout** commands shown below configure the idle-timeout period for the connection type being configured. The idle timeout is the interval that the connection waits after a users most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

Examples

- This idle-timeout (SSH Management) command configures an SSH idle-timeout period of three hours.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 180
```

- This idle-timeout (Telnet Management) command disables automatic connection timeout for telnet connections.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#idle-timeout 0
```

- The shutdown (Telnet Management) command enables and disables Telnet on the switch.
- These commands enable Telnet.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#no shutdown
```

- These commands disable Telnet.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#shutdown
```

3.3 Switch Storage Device Secure Erase

Secure erase is used when all data must be securely removed from the flash and optional SSD storage device(s) within an Arista switch. It securely erases the storage devices whose partitions mount to **/mnt/crash**, **/mnt/drive**, and **/mnt/flash** (as applicable), then repartitions these storage devices and re-creates the file systems for each of their partitions. In other words, the partition table of each storage device is the exact same as before this secure erase procedure (MBR gets destroyed during a secure erase); each partition will have the same file system type and partition label, and be mounted to the same mount point with the same options. This makes it possible to boot the EOS again by installing a new boot-config and EOS SWI, then rebooting (which can be done using Aboot/fullrecover).

All secure erasing is best effort; we use firmware-based secure erase when available, and a software-based mechanism when the firmware mechanism might fail or be insufficient (e.g., writing random data even after sending an ATA Secure Erase command) or does not exist (e.g., eUSB). Unfortunately, no non-physically destructive mechanism can completely guarantee the destruction of all data on a storage device.



Note: Certain Arista switches have a dedicated storage device for serial console logging. While we do consider console output to be sensitive data, we do not secure erase this storage device. Platform support and usage information regarding serial console logging can be found here: https://www.arista.com/en/support/toi/eos-4-21-0f/14038-reload-console-logs.

Preparing for Secure Erase

Always connect to the switch/supervisor via serial console prior to executing the CLI command described below. Executing the CLI command will leave the switch in Aboot; since the Aboot shell is only available from the serial console, a switch will only be accessible via its serial port after executing this command.

If a system has two supervisors, the redundancy state of the supervisor to be secure erased should be `standby`.

Performing Secure Erase

To securely erase the flash and optional SSD storage device(s) on supported platforms, use the **reset system storage secure** command.

Examples

The following commands check the redundancy status of the supervisor to be erased, then perform a switchover to change its status to `standby` preparatory to initiating the secure erase:

```
switch#show redundancy status
  my state = active
peer state = standby switch#config
switch(config)#redundancy manual switchover
This supervisor will be restarted.
```

The following command securely erases data stored on the switch, excluding dedicated console-logging storage:

```
switch#reset system storage secure
WARNING! This will destroy all
data and will NOT be recoverable.
Device will reboot into Aboot, and
execution may take up to one hour.
Would you like to proceed? [y/N]y
```

3.4 Configure Session

The command **`configure session`** creates a configuration session in which CLI commands can be issued that do not take effect until the session is committed . Each **`configure session`** is saved with a unique name. A session can be entered, modified and exited at any time without impacting the currently running system configuration.

When a session is committed, the configuration that was modified during the session is copied into ***running-config***, overwriting any other configuration changes made since the session was created. A session can be aborted or removed, thereby removing the session completely and freeing up memory used by the session. The user must explicitly request that the changes in a deferred session be applied to the configuration of the router by entering a **`commit`** command and exiting the mode. Alternately, the user may abandon the changes by entering an **`abort`** command. An uncommitted configuration session will be discarded if the switch reboots and will time out after 24 hours.

Configuration sessions are used to make sets of changes, after verifying that there are no CLI errors. Configuration sessions allow the administrator to pre-provision a group of CLI commands in a named session, then execute each configuration session at specified times.

3.4.1 Configuration Session

The **`configure session`** command allows users to make a series of configuration changes in a temporary location and commit them to ***running-config*** at once by issuing the **`commit`** command.

- **`configure session <name of session>`** and ***running-config***: The user enters a session (versus **configure terminal** in the case where configuration sessions are not used). If a session name is not specified, a system named session is created. A snapshot of the current ***running-config*** is copied into the session's data structure as the basis of further configuration changes.
- CLI configuration commands: User can run any configuration commands inside the session.
- **`rollback clean-config`**: User can run **rollback** command to revert the sessions configuration to the factory-default configuration (or clean configuration).
- **`show session-config`**: User can run **`show session-config`** to show the sessions configuration, which will be the future ***running-config*** once committed.
- **`commit`**: User issues **`commit`** to commit the changes, which will replace the current ***running-config***.
- **`abort`**: To abort the session and throw away all changes.

- **`exit`**: User can **exit** from the session, and later return to the same session by running **`configure session`** again.
- For named session: More than one CLI instance can enter the same session and make changes to the session configuration. Once the session is committed in any of the CLIs, no other CLI can commit or make any other changes in that session.



Note: committing a configuration session replaces ***running-config*** with the session configuration, which consists of the running configuration at the time the session was initiated plus the commands that were entered as part of the session. Any changes that were made to ***running-config*** since the session was initiated will be overwritten when the session is committed.

3.4.2 Configure Replace

The command **`configure replace <URL>`** replaces the current ***running-config*** with the configuration saved in **<URL>**. By default, **`configure replace <URL>`** will replace ***running-config*** only if the configuration in **<URL>** loads without errors. The command **`configure replace <URL> ignore-errors`** forces the operation even if there are errors.



Note: The command **`copy <URL> running-config`** was typically used to apply a saved configuration file to the system, and append that configuration to the current ***running-config*** (in lieu of replacing it). However, Arista recommends using the CLI command **`configure replace <URL>`** to streamline the process of deterministically restoring the system back to a known good configuration.

The normal workflow internally uses a configuration session to perform the replace.

3.4.3 Configuration CLI

In the CLI, execute the following configuration steps to create a configuration session.

1. **`configure session`** [*name of session*]

   Create or enter a session. If a name is not specified, it is automatically generated. The user is put in the session configuration mode and the prompt will change to show the first six characters of the session name. Designating the name of a session is optional. When *name of session* is not specified, a unique name is assigned.

   ```
   no configure session  name of session
   ```

   Delete the specified configuration session. Designating the name of a session is required.
2. **`commit`**

   Commit the changes made in the session. This command must be issued from within the session configuration mode.

   ```
   abort
   ```

   Abort the session, which is the same as deleting it. This command must be issued from within the session configuration mode.
3. **`rollback clean-config`**

   Revert configuration in the session to the clean, factory-default configuration. This command must be issued from within the session configuration mode.
4. **`service configuration session max completed`** *number*

   Sets a limit on the maximum number of committed sessions that are saved.
5. **`service configuration session max pending`** *number*

   Sets a limit on the maximum number of uncommitted sessions that can be outstanding.

3.5 Recovery Procedures

These sections describe switch recovery procedures:

- Removing the Enable Password from the Startup Configuration
- Reverting the Switch to the Factory Default Startup Configuration
- Restoring the Factory Default EOS Image and Startup Configuration
- USB Support for ZeroTouch Provisioning
- Restoring the Configuration and Image from a USB Flash Drive

The first three procedures require Aboot Shell access through the console port. If the console port is not accessible, use the last procedure in the list to replace the configuration file through the USB Flash Drive.

1 describes the switch booting process and includes descriptions of the Aboot shell, Aboot boot loader, and required configuration files.

3.5.1 Removing the Enable Password from the Startup Configuration

The **enable password** controls access to Privileged EXEC mode. To prevent unauthorized disclosure, the switch stores the **enable password** as an encrypted string that it generates from the clear-text password. When the switch authentication mode is local and an **enable password** is configured, the CLI prompts the user to enter the clear-text password after the user types enable at the EXEC prompt.

The ***startup-config*** file stores the encrypted **enable password** to ensure that the switch loads it when rebooting. If the text version of the **enable password** is lost or forgotten, access to enable mode is restored by removing the encrypted **enable password** from the startup configuration file.



Note: During the recovery process, in a system containing more than one supervisor, the secondary supervisor must be physically removed from the system. This ensures the previous configuration is not recovered from the secondary supervisor upon reboot during the recovery process.

This procedure restores access to enable mode without changing any other configuration settings.

1. Access the Aboot shell:
   a. Power cycle the switch by successively removing and restoring access to its power source.
   b. Type **Ctrl-C** when prompted, early in the boot process.
   c. Enter the Aboot password, if prompted. If the Aboot password is unknown, refer to Restoring the Factory Default EOS Image and Startup Configuration for instructions on reverting all flash directory contents to the factory default, including the startup configuration and EOS image.
2. Change the active directory to /mnt/flash directory.

```
Aboot#cd /mnt/flash
```

3. Open the startup-config file in vi.

```
Aboot#vi startup-config
```

4. Remove the enable password line.
   This is an example of an enable password line:

```
enable password 5 $1$dBXo2KpF$Pd4XYLpI0ap1ZaU7glG1w/
```

5. Save the changes and exit vi.
6. Exit Aboot. This boots the switch.

```
Aboot#exit
```

3.5.2 Reverting the Switch to the Factory Default Startup Configuration

The ***startup-config*** file contains configuration parameters that the switch uses during a boot. Parameters that do not appear in ***startup-config*** are set to their factory defaults when the switch reloads. The process requires the Aboot password if Aboot is password protected.

This procedure reverts EOS configuration settings to the default state through bypassing the ***startup-config*** file during a switch boot.

1. Access the Aboot shell through the console port:
   a. Type reload at the Privileged EXEC prompt.
   b. Type **Ctrl-C** when prompted, early in the boot process.
   c. Enter the Aboot password, if prompted. If the Aboot password is unknown, refer to Restoring the Factory Default EOS Image and Startup Configuration for instructions on reverting all flash directory contents to the factory default, including ***startup-config*** and EOS image.
2. Change the active directory to **/mnt/flash** directory.

```
Aboot#cd /mnt/flash
```

3. Rename the startup configuration file.

```
Aboot#mv startup-config startup-config.old
```

4. Exit Aboot. This boots the switch.

```
Aboot#exit
```

5. Cancel Zero Touch Provisioning (ZTP). Refer to Canceling Zero Touch Provisioning for instructions.
   If ZTP is not canceled, the switch either:
   - boots, using the ***startup-config*** file or boot script that it obtains from the network, or
   - remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.
6. Configure the **admin** and **enable** passwords.

```
switch>enable
switch#configure terminal
switch(config)#enable password xyz1
switch(config)#username admin secret abc41
```

7. Save the new ***running-config*** to the startup configuration file.

```
switch#write
```

8. (Optional) Delete the old startup configuration file.

```
switch#delete startup-config.old
```

   After ZTP is canceled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

3.5.3 Restoring the Factory Default EOS Image and Startup Configuration

A **fullrecover** command removes all internal flash contents (including configuration files, EOS image files, and user files), then restores the factory default EOS image and ***startup-config***. A subsequent installation of the current EOS image may be required if the default image is outdated. This process requires Aboot shell access through the console port.



Note: For hardware that is purchased after June 2017, the factory default partition will not have the backup EOS software image. This is done to increase the flash size on smaller flash size disks and

also since other options are available in the **fullrecover** command functionality to restore factory default EOS image. This is applicable to both fixed system and modular system hardware.

This procedure restores the factory default EOS image and startup configuration.

1. Access the Aboot shell through the console port:
   a. Type reload at the Privileged EXEC prompt.
   b. Type **Ctrl-C** when prompted, early in the boot process.
   c. Enter the Aboot password, if prompted. If the Aboot password is not known, enter an empty password three times, after which the CLI displays:

```
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
 state, or just press Enter to reboot:
```

   d. Type **fullrecover** and go to 4 .
2. Type **fullrecover** at the Aboot prompt.

```
Aboot#fullrecover
```

   Aboot displays this warning:

```
All data on /mnt/flash will be erased; type "yes" and press Enter to
 proceed, or just press Enter to cancel:
```

3. Type **yes** and press **Enter**.
   The switch performs these actions:
   - erases the contents of /mnt/flash
   - writes new boot-config, startup-config, and EOS.swi files to /mnt/flash
   - returns to the Aboot prompt
4. Exit Aboot. This boots the switch.

```
Aboot#exit
```

   The serial console settings are restored to their default values (9600/N/8/1/N).
5. Reconfigure the console port if non-default settings are required.
6. Cancel Zero Touch Provisioning (ZTP). Refer to Canceling Zero Touch Provisioning for instructions.
   If ZTP is not canceled, the switch either:
   - boots, using the ***startup-config*** file or boot script that it obtains from the network, or
   - remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

   After ZTP is canceled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

3.5.4 USB Support for ZeroTouch Provisioning

Use **Arista's Zero Touch Provisioning** to configure a switch without user intervention. The USB adds another way to provide the bootstrap-name and to verify the authenticity of the file-server.

3.5.4.1 USB Deployment

By using a USB drive during ZTP, the following features are possible:

1. Specify the location of the bootstrap file instead of using DHCP Option 67.
2. Provide the ***x509*** root of trust for verifying the bootstrap download location.
3. Provide the enrollment token for CloudVision Service customers.

3.5.4.2 Configuration

A USB containing a *yaml configuration* file is plugged into the Arista EOS switch before powering on the switch.

The configuration (**<USB-ROOT>/ztp/ztpConfig.yaml**) should look like this:

```
"bootstrapUrl"
 "serverCaCertificate"
 "enrollmentToken"
"version": "1.0"
```

bootstrapUrl: URL for bootstrap file, such as **https://cvp/config.py**.

```
"bootstrapUrl"
```

serverCaCertificate: path for ***x509*** root of trust for the remote file server on the USB, such as **"ca.crt"**.

```
"serverCaCertificate"
```

enrollmentToken: path for enrollment token on the USB, such as **"token.tok"**

```
"enrollmentToken"
```

All ZTP related files, **serverCaCertificate** and **enrollmentToken**, should be present in (**<USB-ROOT>/ztp/*),** and the location is to be specified in the ztpConfiguration **yaml w.r.t** to this folder.

```
"version": "1.0"
```

All the fields are optional. For example, this is a valid configuration. It will act as though there is no USB in place.

```
"bootstrapUrl"
 "serverCaCertificate"
 "enrollmentToken"
"version": "1.0"
```

- The following is a sample of configuration which is fully filled out. The structure of the USB drive is:
- USB Drive Roo
  - **ca.crt**
  - **token.tok**

```
"bootstrapUrl"
 "serverCaCertificate"
 "enrollmentToken"
"version": "1.0"
```

3.5.4.3 Advantages

- DHCP Server no longer needs to have Option 67 configured.
- The boot script location can now undergo additional checks such as validating the endpoint prior to downloading and running the boot script.
- Customers wishing to enroll their devices in the CloudVision Service have an easy means to do so.

3.5.5 Restoring the Configuration and Image from a USB Flash Drive

The USB flash drive port can be used to restore an original configuration when you cannot establish a connection to the console port. This process removes the contents of the internal flash drive, restores the factory default configuration, and installs a new EOS image from the USB flash drive.

This procedure restores the factory default configuration and installs an EOS image stored on a USB flash drive.

1. Prepare the USB flash drive:
   a. Verify the drive is formatted with MS-DOS or FAT file system. Most USB drives are pre-formatted with a compatible file system.
   b. Create a text file named **fullrecover** on the USB flash drive. The filename does not have an extension. The file may be empty.
   c. Create a text file named **boot-config**. The last modified timestamp of the **boot-config** file on the USB flash must differ from the timestamp of the **boot-config** file on the switch.
   d. Enter this line in the new **boot-config** file on the USB flash:

      ```
      SWI=flash:EOS.swi
      ```

   e. Copy an EOS image file to the flash drive. Rename it **EOS.swi** if it has a different file name. For best results, the flash drive should contain only these three files, because the procedure copies all files and directories on the USB flash drive to the switch.
      - fullrecover
      - boot-config
      - EOS.swi
2. Insert the USB flash drive into the USB flash port on the switch, as shown in Figure 2: Switch Ports.
3. Connect a terminal to the console port and configure it with the default terminal settings (9600/N/8/1) to monitor progress messages on the console.
4. Power up or reload the switch.

   The switch erases internal flash contents and copies the files from the USB flash drive to internal flash. The switch then boots automatically.
5. Cancel Zero Touch Provisioning (ZTP). Refer to Canceling Zero Touch Provisioning for instructions.

   If ZTP is not canceled, the switch either:
   - boots, using the startup-config file or boot script that it obtains from the network, or
   - remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

   After ZTP is canceled, the switch reboots using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

3.6 Session Management Commands

Global Configuration Commands

- configure replace
- configure session
- management api http-commands
- management console
- management ssh
- management telnet
- management xmpp
- reset system storage secure

Management Configuration Commands

- domain (XMPP Management)
- idle-timeout (Console Management)
- idle-timeout (SSH Management)
- idle-timeout (Telnet Management)
- protocol http (API Management)
- protocol https (API Management)
- protocol https certificate (API Management)
- server (XMPP Management)
- session privilege (XMPP Management)
- shutdown (API Management)
- shutdown (Telnet Management)
- shutdown (XMPP Management)
- switch-group (XMPP Management)
- username (XMPP Management)
- vrf (API Management)
- vrf (XMPP Management)
- xmpp send
- xmpp session

Display Commands

- show inventory
- show xmpp neighbors
- show xmpp status
- show xmpp switch-group

3.6.1 configure replace

The **`configure replace`** command replaces the current configuration with the new configuration from the specified source.

Command Mode

Privileged EXEC

Command Syntax

`configure replace` {***source_file_path:source_file_name*** | boot-extensions | clean-config | installed-extensions | running-config | startup-config}[ignore-errors] [md5 *md5sum*] [skip-checkpoint]

Parameters

- ***source_file_path:source_file_name*** replaces current configuration with the configuration from the specified source file.
- **boot-extensions** replaces current configuration with the boot extensions configuration.
- **clean-config** replaces current configuration with clean and default configurations.
- **installed-extensions** replaces current configuration with the installed extensions configuration.
- **running-config** obsolete.
- **starup-config** replaces current configuration with the startup configuration.
- **ignore-errors** ignores errors while loading the new configuration.
- **md5** *md5sum* performs a checksum to validate data integrity with the specified MD5 hashing algorithm.
- **skip-checkpoint** skips creating the checkpoint file of ***running-config***.

Example

This command replaces the current configuration state with the startup configuration.

```
switch(config)# configure replace start-config
! Preserving static routes. Use 'no ip routing delete-static-routes'
 to clear
them.
switch#
```

3.6.2 configure session

The **configure session** command allows a series of configuration changes to be made in a temporary location, then commited to ***running-config*** later by issuing the **commit** command. An uncommitted configuration session will be discarded if the switch reboots and will time out after 24 hours.

Note: committing a configuration session replaces ***running-config*** with the session configuration, which consists of the running configuration at the time the session was initiated plus the commands that were entered as part of the session. Any changes that were made to ***running-config*** since the session was initiated will be overwritten when the session is committed.

The **no configure session** *session_name* and **default configure session** ***session_name*** commands delete the specified configuration session.

Command Mode

Privileged EXEC

Command Syntax

configure session [***session_name***]

no configure session ***session_name***

default configure session ***session_name***

Parameter

session_name session name.

Guidelines

- If a session name is not specified, a session is created with a name generated by the systems.
- The switch permits up to five uncommitted sessions.
- Any uncommitted sessions are discarded when the switch reboots.
- Uncommitted sessions time out after 24 hours.

Example

This command creates a session (automatically named ***sess-1***) and adds commands to it. Issuing the **commit** command would cause all of the commands to be executed and would overwrite any configuration performed since the session was created.

```
switch(config)# configure session
switch(config-s-sess-1)# no username kevin
switch(config-s-sess-1)# aaa authentication dot1x default group radius
switch(config-s-sess-1)# dot1x system-auth-control
switch(config-s-sess-1)# copy running-config startup-config
switch(config-s-sess-1)# reload all now
switch(config-s-sess-1)#
```

3.6.3 domain (XMPP Management)

The **domain** command configures the switchs XMPP domain name. Only messages using a domain matching the locally configured one are accepted by the XMPP client. The switch's domain name is used if none is specified.

Management over XMPP is disabled by default. To enable it, you must provide the location of the server along with the domain, username and password for the switch.

Arista recommends configuring the XMPP domain before the username, because it will provide shortcuts for the **switch-group** and **username** so they can be configured without the domain attached to it (e.g., USERNAME instead of USERNAME@DOMAIN).

The **no domain** and **default domain** commands delete the domain name by removing the **domain** command from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

domain ***string***

no domain

default domain

Parameters

string domain name (text string).

Examples

- This command configures *test.aristanetworks.com* as the switchs domain name.

```
switch(config)# management xmpp
test1(config-mgmt-xmpp)# server arista-xmpp
test1(config-mgmt-xmpp)# domain test.aristanetworks.com
test1(config-mgmt-xmpp)# username test1@test.aristanetworks.com
 password 0 arista
test1(config-mgmt-xmpp) #no shutdown
```

- This command removes the domain name from the XMPP *configuration*.

```
switch(config-mgmt-xmpp)# no domain
switch(config-mgmt-xmpp)#
```

3.6.4 idle-timeout (Console Management)

The **idle-timeout (Console Management)** command configures the idle-timeout period for console connection sessions. The idle timeout is the interval that the connection waits after a users most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode

Mgmt-console

Command Syntax

idle-timeout ***idle_period***

no idle-timeout

default idle-timeout

Parameters

idle_period session idle-timeout length. Options include:

- ***0*** Automatic connection timeout is disabled.
- ***1 to 86400*** Automatic timeout period (minutes).

Example

- These commands configure a console idle-timeout period of three hours, then return the switch to global configuration mode.

```
switch(config)# management console
switch(config-mgmt-console)# idle-timeout 180
switch(config-mgmt-console)# exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)# management console
switch(config-mgmt-console)# idle-timeout 0
switch(config-mgmt-console)#
```

3.6.5 idle-timeout (SSH Management)

The **idle-timeout (SSH Management)** command configures the idle-timeout period for SSH connection sessions. The idle timeout is the interval that the connection waits after a users most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode

Mgmt-ssh Configuration

Command Syntax

idle-timeout ***idle_period***

no idle-timeout

default idle-timeout

Parameters

idle_period session idle-timeout length. Options include:

- ***0*** Automatic connection timeout is disabled.
- ***1 to 86400*** Automatic timeout period (minutes).

Example

- These commands configure an ssh idle-timeout period of three hours, then return the switch to global configuration mode.

```
switch(config)# management ssh
switch(config-mgmt-ssh)# idle-timeout 180
switch(config-mgmt-ssh)# exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)# management ssh
switch(config-mgmt-ssh)# idle-timeout 0
switch(config-mgmt-ssh)#
```

3.6.6 idle-timeout (Telnet Management)

The **idle-timeout (Telnet Management)** command configures the idle-timeout period for Telnet connection sessions. The idle timeout is the interval that the connection waits after a users most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode

Mgmt-telnet

Command Syntax

idle-timeout ***idle_period***

no idle-timeout

default idle-timeout

Parameters

idle_period session idle-timeout length. Options include:

- ***0*** Automatic connection timeout is disabled.
- ***1 to 86400*** Automatic timeout period (minutes).

Examples

- These commands configure a telnet idle-timeout period of three hours, then return the switch to ***global configuration mode***.

```
switch(config)# management telnet
switch(config-mgmt-telnet)# idle-timeout 180
switch(config-mgmt-telnet)# exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)# management telnet
switch(config-mgmt-telnet)# idle-timeout 0
switch(config-mgmt-telnet)#
```

3.6.7 management api http-commands

The **management api http-commands** command places the switch in mgmt-API configuration mode.

The **no management api http-commands** and **default management api http-commands** commands deletes the mgmt-api-http-command configuration mode statements from ***running-config***.

The ***mgmt-api-http-cmds configuration mode*** is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***mgmt-api-http-cmds configuration mode*** does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

management api http-commands

no management api http-commands

default management api http-commands

Commands Available in Mgmt-API Configuration Mode

- protocol http (API Management)
- protocol https (API Management)
- protocol https certificate (API Management)
- shutdown (API Management)
- vrf (API Management)

Examples

- This command places the switch in ***mgmt-api-http-cmds*** configuration mode.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)#
```

- This command returns the switch to global management mode.

```
switch(config-mgmt-api-http-cmds)# exit
switch(config)#
```

3.6.8 management console

The **management console** command places the switch in mgmt-console configuration mode to adjust the idle-timeout period for console connection sessions. The idle-timeout period determines the inactivity interval that terminates a connection session.

The **no management console** and **default management console** commands delete mgmt-console configuration mode statements from ***running-config***.

The ***mgmt-console*** configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***mgmt-console*** configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

management console

no management console

default management console

Commands Available in mgmt-console Configuration Mode

idle-timeout (Console Management)

Examples

- This command places the switch in mgmt-console configuration mode:

```
switch(config)# management console
switch(config-mgmt-console)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-console)# exit
switch(config)#
```

3.6.9 management ssh

The **management ssh** command places the switch in mgmt-ssh configuration mode to adjust SSH session connection parameters.

The **no management ssh** and **default management ssh** commands delete the ***mgmt-ssh*** configuration mode statements from ***running-config***.

The ***mgmt-ssh*** configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***mgmt-ssh*** configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

management ssh

no management ssh

default management ssh

Commands Available in Mgmt-ssh Configuration Mode

- **authentication mode (Management-SSH)**
- **cipher (Management-SSH)**
- **fips restrictions (Management-SSH)**
- **hostkey (Management-SSH)**
- **idle-timeout (Management-SSH)**
- **ip access group (Management-SSH)**
- **ipv6 access group (Management-SSH)**
- **login timeout (Management-SSH)**
- **key-exchange (Management-SSH)**
- **mac hmac (Management-SSH)**
- **server-port (Management-SSH)**
- **shutdown (Management-SSH)**
- **vrf** (Management-SSH)

Examples

- This command places the switch in mgmt-ssh configuration mode:

```
switch(config)# management ssh
switch(config-mgmt-ssh)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-ssh)# exit
switch(config)#
```

3.6.10 management telnet

The **management telnet** command places the switch in mgmt-telnet configuration mode to adjust telnet session connection parameters.

The **no management telnet** and **default management telnet** commands delete the mgmt-telnet configuration mode statements from ***running-config***.

The **mgmt-telnet configuration mode** is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***mgmt-telnet configuration mode*** does not affect ***running-config***. The **exit** command returns the switch to ***global configuration mode***.

Command Mode

Global Configuration

Command Syntax

management telnet

no management telnet

default management telnet

Commands Available in mgmt-telnet Configuration Mode

- **idle-timeout (Management-Telnet)**
- **ip access group (Management-Telnet)**
- **ipv6 access group (Management-Telnet)**
- **shutdown (Management-Telnet)**
- **vrf (Management-Telnet)**

Examples

- This command places the switch in mgmt-telnet configuration mode:

```
switch(config)# management telnet
switch(config-mgmt-telnet)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-telnet)# exit
switch(config)#
```

3.6.11 management xmpp

The **management xmpp** command places the switch in mgmt-xmpp configuration mode. Management over XMPP is disabled by default. To enable XMPP, you must provide the location of the XMPP server along with the username and password for the switch.

The **no management xmpp** and **default management xmpp** commands delete the mgmt-xmpp configuration mode statements from ***running-config***.

The ***mgmt-xmpp*** configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***mgmt-xmpp*** configuration mode does not affect ***running-config***. The **exit** command returns the switch to ***global configuration mode***.

Command Mode

Global Configuration

Command Syntax

management xmpp

no management xmpp

default management xmpp

Commands Available in Mgmt-xmpp Configuration Mode

- **domain (Management-xmpp)**
- **server (Management-xmpp)**
- **session (Management-xmpp)**
- **shutdown (Management-xmpp)**
- **switch-group (Management-xmpp)**
- **username (Management-xmpp)**
- **vrf (Management-xmpp)**

Examples

- This command places the switch in ***mgmt-xmpp configuration mode***:

```
switch(config)# management xmpp
switch(config-mgmt-xmpp)#
```

- This command returns the switch to ***global*** management mode:

```
switch(config-mgmt-xmpp)#exit
switch(config-mgmt-xmpp)#
```

3.6.12 protocol http (API Management)

The **protocol http** command enables the hypertext transfer protocol (HTTP) server.

The **no protocol http** and **default protocol http** commands disable the HTTP server by removing the **protocol http** statement from ***running-config***.

Command Mode

Mgmt-API Configuration

Command Syntax

protocol http [**TCP_PORT**]

no protocol http

default protocol http

Parameters

- **TCP_PORT** Port number to be used for the HTTP server. Options include:
  - ***no parameter*** Specifies default ***port number 80***.
  - **port *1 to 65535*** Specifies HTTP server port number. Value ranges from ***1*** to ***65535***.
- **localhost** The name of the server bound on the localhost.
- **port** The number of the TCP port to serve on.

Related Commands

management api http-commands places the switch in ***mgmt-api configuration mode***.

Example

These commands enables the management API for the HTTP server.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)#
```

3.6.13 protocol https (API Management)

The **protocol https** command enables the HTTP secure server. The HTTP secure server is active by default.

The **default protocol https** command restores the default setting by removing the **no protocol https** statement from ***running-config***. The **no protocol https** command disables the HTTP secure server.

Command Mode

Mgmt-API Configuration

Command Syntax

protocol https [**TCP_PORT**]

no protocol https

default protocol https

Parameters

- **TCP_PORT** Port number to be used for the HTTPS server. Options include:
  - ***no parameter*** Specifies default ***port number 443***.
  - **port *1 to 65535*** Specifies HTTP server port number. Value ranges from ***1*** to ***65535***.
- **certificate** The HTTPS key and certificate to use.
- **cipher** Exclusive list of cryptographic ciphers.
- **key-exchange** Exclusive list of key-exchange algorithms.
- **mac** Exclusive list of MAC algorithms.
- **port** The number of the TCP port to serve on.
- **ssl** Configure SSL options.

Related Commands

management api http-commands places the switch in ***mgmt-api configuration mode***.

Examples

- These commands enables service to the HTTP server. The **no shutdown** command allows access to the service.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)# protocol https
switch(config-mgmt-api-http-cmds)# no shutdown
```

- These commands specifies the port number that should be used for the HTTPS server. The **no shutdown** command allows access to the service.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)# protocol https port 52
switch(config-mgmt-api-http-cmds)# no shutdown
```

3.6.14 protocol https certificate (API Management)

The **protocol https certificate** command configures the HTTP secure server to request an X.509 certificate from the client. The client then authenticates the certificate with a public key.

The **no protocol https certificate** and **default protocol https certificate** commands restore default behavior by removing the **protocol https certificate** statement from ***running-config***.

Command Mode

Mgmt-API Configuration

Command Syntax

protocol https certificate

no protocol https certificate

default protocol https certificate

Related Command

management api http-commands places the switch in ***mgmt-api configuration mode***.

Example

These commands configure the HTTP secure server to request an X.509 certificate from the client for authentication.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)# protocol https certificate
switch(config-mgmt-api-http-cmds)#
```

3.6.15 reset system storage secure

Use the **reset system storage secure** command to trigger the secure erase mechanism. A secure erase is generally defined as a command that deliberately, permanently, and irreversibly removes/destroys the data stored on a storage device, rendering that data unrecoverable.

Command Mode

EXEC

Command Syntax

`reset system storage secure`

Examples

- To trigger the secure erase mechanism, use the **`reset system storage secure`** command.

```
switch# reset system storage secure
WARNING! This will destroy all
data and will NOT be recoverable.
Device will reboot into Aboot, and
execution may take up to one hour.
Would you like to proceed? [y/N]
```

- If a particular platform does not support the **`reset system storage secure`** command , the following message will appear:

```
switch# reset system storage secure
% Unavailable command (not supported on this hardware platform)
```

3.6.16 server (XMPP Management)

The **`server`** command adds a XMPP server to ***running-config***. Multiple XMPP servers can be set up for redundancy. For redundant configurations, the XMPP server location should be a DNS name and not a raw IP address. The DNS server is responsible for returning the list of available XMPP servers, which the client can go through until an accessible server is found.

User authentication is provided by the XMPP server. Command authorization can be provided by EOS local configuration or TACACS+. The XMPP server should use the same authentication source as the switches. RADIUS is not supported as an XMPP authorization mechanism.

The **`no server`** and **`default server`** commands remove the specified XMPP server from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

`server` **SERVER_NAME** [**SERVER_PORT**]

`no server`

`default server`

Parameters

- **SERVER_NAME** XMPP server location. Options include:
  - ***IP address*** in dotted decimal notation.
  - a host name for the XMPP server.
- **SERVER_PORT** Server port. Options include:
  - **port *1 to 65535*** where *number* ranges from ***1*** to ***65535***. If no port is specified, the default ***port 5222*** is used.

Examples

- This command configures the server hostname **`arista-xmpp to server port 1`**.

```
switch(config)# management xmpp
switch(config-mgmt-xmpp)# server arista-xmpp port 1
```

- This command removes the XMPP server.

```
switch(config-mgmt-xmpp)# no server
```

3.6.17 session privilege (XMPP Management)

The **session privilege** command will place the user in EXEC mode. The initial privilege level is meaningless by default. However, with the configuration of roles, users can add meaning to the different privilege levels. By default, XMPP does not limit access to any command.

Level 1-15: Commands accessible from EXEC Mode.

If AAA is not configured and the switch is configured to connect to the XMPP client, any message received is executed with privilege level 1 by default.

The **no session privilege** and **default session privilege** commands revert the list contents to **none** for the specified privilege levels.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

session privilege PRIV_LEVEL

no session privilege

default session privilege

Parameter

PRIV_LEVEL Privilege levels of the commands. Value ranges from ***0*** and ***15*.**

Examples

- These commands authorizes configuration commands (privilege level config 5) for XMPP.

```
switch(config)#(config)# management xmpp
switch(config-mgmt-xmpp)# session privilege 5
switch(config-mgmt-xmpp)#
```

- This command removes the privilege levels set for the XMPP session.

```
switch(config)# management xmpp
switch(config-mgmt-xmpp)# no session privilege
```

3.6.18 show inventory

The **show inventory** command displays the hardware components installed in the switch. Serial numbers and a description is also provided for each component.

Command Mode

EXEC

Command Syntax

show inventory

Example

This command displays the hardware installed in a DCS-7150S-52 switch.

```
switch> show inventory
System information
  Model                     Description
  ------------------------- ----------------------------------
  DCS-7150S-52-CL           52-port SFP+ 10GigE 1RU + Clock

  HW Version  Serial Number  Mfg Date
  ----------- -------------- ----------
  02.00       JPE13120702    2013-03-27

System has 2 power supply slots
  Slot Model             Serial Number
  ---- ----------------- -----------------
  1    PWR-460AC-F       K192KU00241CZ
  2    PWR-460AC-F       K192L200751CZ

System has 4 fan modules
  Module  Number of Fans  Model             Serial Number
  ------- --------------- ----------------- -----------------
  1       1               FAN-7000-F        N/A
  2       1               FAN-7000-F        N/A
  3       1               FAN-7000-F        N/A
  4       1               FAN-7000-F        N/A

System has 53 ports
  Type              Count
  ----------------- ----
  Management        1
  Switched          52

System has 52 transceiver slots
  Port Manufacturer     Model             Serial Number    Rev
  ---- ---------------- ----------------- ---------------- ----
  1    Arista Networks  SFP-10G-SR        XCW1225FD753     0002
  2    Arista Networks  SFP-10G-SR        XCW1225FD753     0002

  51   Arista Networks  SFP-10G-SR        XCW1225FD753     0002
  52   Arista Networks  SFP-10G-SR        XCW1225FD753     0002 switch>
```

3.6.19 show xmpp neighbors

The **show xmpp neighbors** command displays all neighbors and their connection status. The XMPP server keeps track of all relationships between its users.

Command Mode

EXEC

Command Syntax

show xmpp neighbors

Example

This command displays all the XMPP neighbors and their connection status.

```
switch# show xmpp neighbors
```

```
Neighbor                          State           Last Seen Login Time
------------------------------    --------------- --------------------
admin@test.aristanetworks.com     present         0:01:40 ago
test1@test.aristanetworks.com     present         20:29:39 ago

Neighbor                          Status Message
------------------------------    ------------------------------------
admin@test.aristanetworks.com
test1@test.aristanetworks.com   Arista Networks DCS-7048T-4S switch#
```

3.6.20 show xmpp status

The **show xmpp status** command displays the current XMPP connection status to the server.

The XMPP server keeps track of all relationships between its users. In order for two users to directly communicate, this relationship must first be established and confirmed by the other party.

Switches automatically confirm requests from outside parties as long as they are a user from the same domain name, for example when you chat with your switch from your own XMPP chat client.

Command Mode

EXEC

Command Syntax

show xmpp status

Example

This command displays the current XMPP connection status to the server.

```
switch# show xmpp status
XMPP Server:  port 5222
Client username: test@test.aristanetworks.com
Default domain: test.aristanetworks.com
Connection status: connected switch#
```

3.6.21 show xmpp switch-group

The **show xmpp switch-group** command displays the configured and active switch groups for the switch.

Command Mode

EXEC

Command Syntax

show xmpp switch-group

Example

This command displays the configured and active switch groups.

```
switch# show xmpp switch-group
testroom@conference.test.aristanetworks.com switch#
```

3.6.22 shutdown (API Management)

The **shutdown** command, in mgmt-api configuration mode, disables management over API on the switch. API is disabled by default.

The **no shutdown** command, in mgmt-api configuration mode, enables the management API access.

The **default shutdown** command, in mgmt-api configuration mode, disables the management API access.

Command Mode

Mgmt-API Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Related Command

management api http-commands places the switch in ***mgmt-api*** configuration mode.

Example

- These commands disables API access to the HTTP server.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)# shutdown
switch(config-mgmt-api-http-cmds)#
```

- These commands enables API access to the HTTP server.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)# no shutdown
switch(config-mgmt-api-http-cmds)#
```

3.6.23 shutdown (Telnet Management)

The **shutdown** command, in management-telnet mode, disables or enables Telnet on the switch. Telnet is disabled by default. The management telnet command places the switch in management-telnet mode.

- To enable Telnet, enter **no shutdown** at the management-telnet prompt.
- To disable Telnet, enter **shutdown** at the management-telnet prompt.

Command Mode

Management-Telnet Configuration

Command Syntax

shutdown

no shutdown

Examples

- These commands enable Telnet, then return the switch to ***global*** configuration mode.

```
switch(config)# management telnet
switch(config-mgmt-telnet)# no shutdown
```

```
switch(config-mgmt-telnet)# exit
switch(config)#
```

- This command disables Telnet.

```
switch(config-mgmt-telnet)# shutdown
```

3.6.24 shutdown (XMPP Management)

The **shutdown** command, in mgmt-xmpp mode, disables or enables management over XMPP on the switch. XMPP is disabled by default.

The **no shutdown** and **default shutdown** commands re-enable XMPP by removing the **shutdown** command from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- These commands enable management over XMPP, then return the switch to ***global*** configuration mode.

```
switch(config-mgmt-xmpp)# no shutdown
switch(config-mgmt-xmpp)# exit
switch(config)#
```

- This command disables management over XMPP.

```
switch(config-mgmt-xmpp)# shutdown
switch(config-mgmt-xmpp)#
```

3.6.25 switch-group (XMPP Management)

The **switch-group** command allows you to configure each switch to join specified chat rooms on startup. In order for the switch to participate in a chat group, the switch has to be configured to belong to the specified chat room.

The **no switch-group** and **default switch-group** commands delete the specified switch-group configuration (or all switch-group configurations if no name is specified) by removing the corresponding **switch-group** statement from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

switch-group ***name*** **SECURITY**

no switch-group

default switch-group

Parameters

- ***name*** Group name text that the user enters at the login prompt to access the CLI. To enter multiple names in a single command, separate them with spaces.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  @ # $ % ^ & * - _ = + ; < > ,. ~ |
- **SECURITY** password assignment.
  - **password** *pwd_txt name* is protected by specified password. *pwd_txt* is a clear-text string.
  - **password 0** *pwd_txt name* is protected by specified password. *pwd_txt* is a clear-text string.
  - **password 7** *pwd_txt name* is protected by specified password. *pwd_txt* is encrypted string.

Guidelines

- A switch group is an arbitrary grouping of switches within the network which belong to one chat group.
- In order to belong to one or more switch groups, the switch has to be manually assigned to it.
- Switch groups are defined dynamically based on the configuration of all of the switches in the network.
- As per the multi-user chat XMPP standard (XEP-0045), switch groups have a full name of GROUPNAME@conference.DOMAIN.
- All CLI commands allow either the full group name or the short name, which are appended the @conference.DOMAIN.
- If the switch belongs to multiple chat rooms, you must configure each group with a separate command.

Examples

- These commands configure the switch-group to be part of the chat room.

```
switch(config)# management xmpp
switch(config-mgmt-xmpp)# switch-group
testroom@conference.test.aristanetworks.com password 0 arista
```

- Use the **show xmpp switch-group** to verify the active switch-group for the switch.

```
switch# show xmpp switch-group
testroom@conference.test.aristanetworks.com
```

3.6.26 username (XMPP Management)

The **username** command configures the switch's username and password on the XMPP server.

The **no username** and **default username** commands delete the specified username by removing the corresponding **username** statement from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

username ***name*** **SECURITY**

no username

default username

Parameters

- ***name*** username text that defines the XMPP username and password.

Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

@ # $ % ^ & * ( ) - _ = + { } [ ] ; < > ,. ~ |

- **SECURITY** password assignment.
  - **password** *pwd_txt name* specifies and unencrypted shared key. *pwd_txt* is a clear-text string.
  - **password 0** *pwd_txt name* specifies and unencrypted key. *pwd_txt* is a clear-text string.
  - **password 7** *pwd_txt name* specifies a hidden key. *pwd_txt* is encrypted string.

Guidelines

Encrypted strings entered through this parameter are generated elsewhere. The **password 7** option (**SECURITY**) is typically used to enter a list of username-passwords from a script.

Examples

- These commands create the username and assigns it a password. The password is entered in clear text because the parameter is set to ***0***.

```
switch(config)# management xmpp
switch(config-mgmt-xmpp)# server arista-xmpp
switch(config-mgmt-xmpp)# domain test.aristanetworks.com
switch(config-mgmt-xmpp)# username test1@test.aristanetworks.com
 password 0
arista
switch(config-mgmt-xmpp)# no shutdown
```

- This command removes all usernames from the XMPP server.

```
switch(config-mgmt-xmpp)# no username
switch(config-mgmt-xmpp)#
```

3.6.27 vrf (API Management)

The **vrf** command places the switch in VRF configuration mode for the server. If the named VRF does not already exist, this command creates it.

Command Mode

Mgmt-API Configuration

Command Syntax

vrf VRF_INSTANCE

Parameters

VRF_INSTANCE specifies the VRF instance.

- **default** Instance is created in the default VRF.
- ***vrf_name*** Instance is created in the specified user-defined VRF.

Related Command

management api http-commands places the switch in ***mgmt-api*** configuration mode.

Example

This command creates a VRF named management-vrf and places the switch in ***VRF*** configuration mode for the new VRF.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)# vrf management-vrf
switch(config-mgmt-api-http-cmds-vrf-management-vrf)#
```

3.6.28 vrf (XMPP Management)

The **vrf** command places the switch in VRF configuration mode for the XMPP server. If the named VRF does not already exist, this command creates it.

The VRF configuration for the client is for the entire XMPP service, rather than per server. All servers resolving on a particular hostname must be reachable in the same VRF.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

vrf [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE specifies the VRF instance.

- **default** Instance is created in the default VRF.
- ***vrf_name*** Instance is created in the specified user-defined VRF.

Example

This command creates a VRF named management-vrf and places the switch in ***VRF*** configuration mode for the server.

```
switch(config)# management xmpp
switch(config-mgmt-xmpp)# vrf management-vrf
switch(config-mgmt-xmpp)
```

3.6.29 xmpp send

The **xmpp send** command can be used to connect to the XMPP server and send messages to switches or switch groups within the network.

Before switches can send messages to each other, they must friend each other. An easy way to have them auto friend each other is to have them join the same chat room. The friendship between switches can be verified by using the **show xmpp neighbor** command.

Command Mode

Privileged EXEC

Command Syntax

xmpp send to ***neighbor*** **XMIT_TYPE** ***content***

Parameters

- ***neighbor*** Options include switches or switch groups within the network that are connected as friends in a chat room.

- **XMIT_TYPE** Transmission type. Valid options include:
  - **command** Sends an XMPP command.
  - **message** Sends an XMPP message.
- ***content*** The command you want the friends within the chat room to display or execute.

Configuration Restrictions

- Only enable-mode commands are allowed within the multi-switch CLI.
- Changing into a different CLI mode and running several commands in that mode is not supported (e.g., into configuration mode).
- An external XMPP client (for example Adium) can be used to send multiple lines within a single message. By sending multiple lines, it is possible to change into another CLI mode. After the message is processed, the switch automatically return to the enable mode.
- Commands that prompt for a response (like reload) are not supported.
- Long commands, such as image file copies, may cause the switch XMPP client to momentarily stop responding and disconnect. The switch should reconnect and the long command should complete.
- Many command outputs display in a specific table format. To achieve the same visual feel as through a terminal, use a monospaced font, such as Courier, for the incoming messages.

Example

This command sends the switch in the chat room the request to execute the show version command.

```
switch# xmpp send test2 command show version
message from user: test2@test.aristanetworks.com
-----------------------------------------------
Hardware version:     04.40
Serial number:        JFL08432083
System MAC address:   001c.7301.7d69
Software image version:  4.12.3
Architecture:            i386
Internal build version:  4.12.3
Internal build ID:       f5ab5f57-9c26-4fe4-acaa-fb60fa55d01d
Uptime:                  2 hours and 38 minutes
Total memory:            1197548 kB
Free memory:             182452 kB
```

3.6.30 xmpp session

The **xmpp session** command is similar to running SSH from the switch. The user is required to input their username (default is to USER@DEFAULTDOMAIN) and password in order to connect to the XMPP server. This command allows you to interact in the enable mode with a switch or switch group over XMPP using the standard CLI, with access to help and tab completion. All commands are then executed remotely and only the non-empty results are displayed on the screen.

Command Mode

Privileged EXEC

Command Syntax

xmpp session ***switchgroup***

Parameters

switchgroup The option includes the switch group within the network that is connected as friends in a chat room.

Configuration Restrictions

- Only enable-mode commands are allowed within the multi-switch CLI.
- Changing into a different CLI mode and running several commands in that mode is not supported (e.g., into configuration mode).
- An external XMPP client (for example Adium) can be used to send multiple lines within a single message. By sending multiple lines, it is possible to change into another CLI mode. After the message is processed, the switch automatically return to the enable mode.
- Commands that prompt for a response (like reload) are not supported.
- Long commands, such as image file copies, may cause the switch XMPP client to momentarily stop responding and disconnect. The switch should reconnect and the long command should complete.
- Many command outputs display in a specific table format. To achieve the same visual feel as through a terminal, use a monospaced font, such as Courier, for the incoming messages.

Example

This command displays the status of ***Ethernet 3*** from ***test1***, which is a member of the switch group chat room.

```
switch# xmpp session all@test.aristanetworks.com
xmpp-all# show int Eth3 status

response from: test1@test.aristanetworks.com
--------------------------------------------------
Port  Name  Status     Vlan     Duplex  Speed   Type
Et3   bs3   connected  in Po3   a-full  a-1000  10GBASE-SR switch#
```

Chapter 4

Administering the Switch

This chapter describes administrative tasks that are typically performed only after initially configuring the switch or after recovery procedures.

This chapter includes these sections:

- Managing the Switch Name
- System Clock and Time Protocols
- Managing Display Attributes
- Logging of Event Notifications
- Event Monitor
- Managing EOS Extensions
- Switch Administration Commands

4.1 Managing the Switch Name

These sections describe how to configure the switchs domain and host name.

- Assigning a Name to the Switch describes the assigning of an FQDN to the switch.
- Specifying DNS Addresses describes the adding of name servers to the configuration.

4.1.1 Assigning a Name to the Switch

A Fully Qualified Domain Name (FQDN) labels the switch and defines its organization ID in the Domain Name System hierarchy. The switchs FQDN consists of a host name and domain name.

The host name is uniquely associated with one device within an IP-domain. The default host name is ***localhost***. You can configure the prompt to display the host name, as described in prompt.

- To assign a host name to the switch, use the **`hostname`** command. To return the switchs host name to the default value of ***localhost***, use the **`no hostname`** command.
- To specify the domain location of the switch, use the **`dns domain`** command.

Examples

- This command assigns the string *main-host* as the switchs host name.

```
switch(config)# hostname main-host
main-host(config)#
```

- This command configures *aristanetworks.com* as the switchs domain name.

```
switch(config)# dns domain aristanetworks.com
switch(config)#
```

- This procedure configures ***sales1.samplecorp.org*** as the switchs FQDN.

```
switch(config)# dns domain samplecorp.org
switch(config)#
```

- This ***running-config*** extract contains the switchs host name and IP-domain name.

```
switch# show running-config
! Command: show running-config
! device: switch (DCS-7150S-64-CL, EOS-4.13.2F)
!

vlan 3-4
!
username john secret 5 $1$a7Hjept9$TIKRX6ytkg8o.ENja.na50
!
hostname sales1
ip name-server vrf default 172.17.0.22
dns domain samplecorp.org
!

end switch#
```

4.1.2 Specifying DNS Addresses

The Domain Name Server (DNS) maps FQDN labels to IP addresses and provides addresses for network devices. Each network requires at least one server to resolve addresses. The configuration file can list a maximum of three server addresses.

To add name servers to the configuration, use the **ip name-server** command. Each command can add multiple servers. All server addresses support multiple VRFs and a priority may be specified for each name server. If all name servers have the default priority (***0***) the default DNSmasq behavior is followed for the configuration. It queries all name servers simultaneously and forwards the requests to the first name server for 50 queries or 20 seconds, whichever expires sooner for answering. If any priorities are non-zero, queries are issued in order with a five second timeout between unresponsive name servers.

Note: DNSmasq does not fulfill in-flight requests to unresposive name servers.

Example

This code displays the configuration file.

```
switch(config)# show ip name-server
IP Address VRF     Priority
---------- ------- --------
10.0.0.1   default        0
10.0.0.2   default        1
10.0.0.1   vrf1           2
10.0.0.2   vrf1           3
fc00::1    default        4
```

The switch assigns source IP addresses to outgoing DNS requests. To force the switch to use a single, user-defined source interface for all requests, use the **ip domain lookup** command.

Examples

- This command forces the switch to use ***VLAN 5*** as the source interface for DNS requests originating from the default VRF.

```
switch(config)# ip domain lookup source-interface Vlan5
switch(config)#
```

- This command forces the switch to use ***VLAN 10*** as the source interface for DNS requests originating from VRF purple.

```
switch(config)# ip domain lookup vrf purple source-interface Vlan10
switch(config)#
```

- This command configures up to four name servers on VRF purple.

```
switch(config)# ip name-server vrf purple 10.1.1.24 priority 4
switch(config)#
```

- This command removes the name servers.

```
switch(config)# no ip name-server vrf purple 10.1.1.24
switch(config)#
```

- This command removes all configured name in all VRFs.

```
switch(config)# no ip name-server
switch(config)#
```



Note:

NXDOMAIN is considered a valid reply for the query.

4.2 System Clock and Time Protocols

The switch uses the system clock for displaying the time and for time-stamping messages. The system clock is set to Coordinated Universal Time (UTC); the switch calculates local time based on the time zone setting. Time-stamps and time displays are in local time. The system clock can be set either manually or via Network Time Protocol (NTP); any NTP servers properly configured on the switch override time that is manually entered.

The following sections deal with the configuration of the system clock and the use of NTP and PTP.

- Configuring the Time Zone
- Setting the System Clock Manually
- Displaying the Time
- Network Time Protocol (NTP)

4.2.1 Configuring the Time Zone

The time zone setting is used by the switch to convert the system time (UTC) to local time. To specify the time zone, use the **clock timezone** command.

Examples

- These commands configure the switch for the United States Central Time Zone.

```
switch(config)#clock timezone US/Central
switch(config)#show clock
Mon Jan 14 18:42:49 2013
timezone is US/Central
switch(config)#
```

- To view the predefined time zone labels, enter **clock timezone** with a question mark.

```
switch(config)#clock timezone ?
 Africa/Abidjan                    Africa/Accra
```

```
  WET                                   WET timezone
  Zulu                                  Zulu timezone

switch(config)#clock timezone
```

- This command displays all time zone labels that start with **America**.

```
switch(config)#clock timezone AMERICA?
America/Adak                    America/Anchorage

America/Yellowknife
switch(config)#clock timezone AMERICA
```

4.2.2 Setting the System Clock Manually

The **clock set** command manually configures the system clock time and date, in local time. Any NTP servers properly configured on the switch override time that is manually entered.

Example

This command manually sets the switch time.

```
switch#clock set 08:15:24 14 Jan 2013
Mon Jan 14 08:15:25 2013
timezone is US/Central
```

4.2.3 Displaying the Time

To display the local time and configured time zone, enter the **show clock** command.

Example

This command displays the switch time.

```
switch(config)#show clock
Mon Jan 14 16:32:46 2013
timezone is America/Los_Angeles
```

4.2.4 Network Time Protocol (NTP)

Network Time Protocol (NTP) is enabled on the switch by default, and time settings from any properly configured NTP server will override manual setting of the system clock.

NTP servers synchronize time settings of systems running an NTP client. The switch supports NTP versions 1 through 4. The default is version 4. After configuring the switch to synchronize with an NTP server, it may take up to ten minutes for the switch to set its clock. The ***running-config*** lists NTP servers that the switch is configured to use.

The following NTP sections deal with NTP on the switch:

- Configuring the NTP Server
- Configuring the NTP Source
- Configuring the Switch as an NTP Server
- Configuring NTP Authentication
- Viewing NTP Settings and Status

4.2.4.1 Configuring the NTP Server

The **ntp server** command adds a server to the list or modifies the parameters of a previously listed address. When the system contains multiple NTP servers, the prefer keyword can be used to specify a preferred NTP server, which will be used as the NTP server if not discarded by NTP.

Note that all NTP servers must be in the same VRF, and that they are added in the default VRF if no VRF is specified.

The system clock is set via NTP if NTP is enabled and there is at least one NTP server properly configured on the switch, and NTP overrides manual setting of the system clock. NTP is enabled by default. To disable NTP, use the **no ntp** command.

Example

These commands add three NTP servers, designating the second server as preferred.

```
switch(config)#ntp server local-NTP
switch(config)#ntp server 172.16.0.23 Prefer
switch(config)#ntp server 172.16.0.25
```

4.2.4.2 Configuring the NTP Source

To control the address to which NTP responses to the switch are sent, a local interface can be specified as the source in outgoing NTP packets using the **ntp local-interface** command. The IP address of that interface is then used as the source address in all outgoing NTP packets unless the switch is acting as an NTP server and a server-specific source is configured using the **source** option of the **ntp server** command.

Example

This command configures the IP address of ***VLAN interface 25*** as the source specified in all outgoing NTP packets.

```
switch(config)#ntp local-interface vlan 25
switch(config)#
```

4.2.4.3 Configuring the Switch as an NTP Server

To configure the switch to accept NTP requests on all interfaces, use the **ntp serve all** command to enable NTP server mode globally on the switch. To configure an individual interface to accept or deny NTP requests, use the **ntp serve** command. Interface level settings override the global settings, and changing the settings at either the global or interface level also causes the switch to re-synchronize with its upstream NTP server. NTP server mode is disabled by default.

Examples

- This command configures the switch to act as an NTP server, accepting NTP requests.

```
switch(config)# ntp serve all
switch(config)#
```

- These commands configure ***interface ethernet 5*** to accept NTP requests regardless of global settings.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)#ntp serve
switch(config-if-Et5)#
```

4.2.4.4 Configuring NTP Authentication

The switch can be configured to accept NTP packets only from an authenticated server or client. NTP authentication is disabled by default.

To configure the switch to authenticate NTP packets, create one or more authentication keys using the **ntp authentication-key** command, specify which keys are trusted by using the **ntp trusted-key** command, use the **ntp authenticate** command to enable NTP authentication, and specify to use the trusted-key for a specific server. The NTP server must be configured to use the same authentication key and key ID number.



Note: When NTP authentication is enabled on a switch, all NTP servers upstream of the switch, as well as all NTP clients of the switch, should have matching keys configured, and clients must have NTP authentication enabled.

Example

These commands configure the switch to authenticate NTP packets using key 328 with the plaintext password timeSync.

```
switch(config)#ntp authentication-key 328 md5 timeSync
switch(config)#ntp trusted key 328
switch(config)#ntp authenticate
switch(config)#
```

4.2.4.5 Viewing NTP Settings and Status

To display the status of Network Time Protocol (NTP) on the switch, use the **show ntp status** command. To display the status of connections to NTP servers, use the **show ntp associations** command.



Note: In the output for **show ntp associations**, the reference ID (which identifies the time source of the NTP server) is either the IPv4 address of the time source or, if that source has an IPv6 address, the first four octets of the MD5 hash of that IPv6 address. In EOS releases prior to *4.23.2*, the **show ntp status** command identified the system peer by its reference ID as described above, but in later releases it shows the IP address (whether IPv4 or IPv6).

Examples

- This command displays the status of the switch's NTP connection.

```
switch# show ntp status
synchronised to NTP server (192.168.78.62) at stratum 3
   time correct to within 66 ms
   polling server every 1024 s
switch #
```

- This command displays data about the NTP servers in the configuration.

```
switch# show ntp associations
remote           refid          st t when  poll reach   delay   offset  jitter
==============================================================================
+1.ntp.arista.co 125.157.10.11   2 u  539  1024  377  121.748   -0.345   0.893
-3.ntp.arista.co 127.31.152.34   2 u  868  1024  377  101.671    2.434   1.529
+2.ntp.arista.co 176.131.12.185  2 u  676  1024  377  116.505     0.03   0.768
*4.ntp.arista.co 120.181.192.192 2 u  696  1024  377   48.431   -0.416    0.15 switch#
```

4.3 Managing Display Attributes

Display commands control the content of the banner and the command line prompt.

4.3.1 Banners

The switch can display two banners:

- **Login banner**: The login banner precedes the login prompt. One common use for a login banner is to warn against unauthorized network access attempts.
- **motd banner**: The message of the day (motd) banner is displayed after a user logs into the switch.

This output displays both banners in bold:

```
This is a login banner
switch login: john
Password:
Last login: Mon Jan 14 09:24:36 2013 from adobe-wrks.aristanetworks.com
This is an motd banner
switch>
```

These commands create the login and motd banner shown earlier in this section.

```
switch(config)#banner login
Enter TEXT message. Type 'EOF' on its own line to end.
This is a login banner
EOF
switch(config)#banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd banner
EOF
switch(config)#
```

To create a banner:

1. Enter global configuration mode.

   ```
   switch#config
   switch(config)#
   ```

2. Enter banner edit mode by typing the desired command:
   - To create a login banner, type **banner login**.
   - To create a motd banner, type **banner motd**.

   The switch responds with instructions on entering the banner text.

   ```
   switch(config)#banner login
   Enter TEXT message. Type 'EOF' on its own line to end.
   ```

3. Enter the banner text.

   ```
   This is the first line of banner text.
   This is the second line of banner text.
   ```

4. Press **Enter** to place the cursor on a blank line after completing the banner text.
5. Exit banner edit mode by typing EOF.

   ```
   EOF
   switch(config)#
   ```

4.3.2 Configuring prompt

The prompt provides an entry point for EOS commands. The `prompt` command configures the contents of the prompt. The **`no prompt`** command returns the prompt to the default of %H%P.

Characters allowed in the prompt include A-Z, a-z, 0-9, and these punctuation marks:

! @ # $ % Âˆ & * ( ) - = + f g [ ] ; : < > , . ? / Â˜ n

The prompt supports these control sequences:

- %s “ space character
- %t “ tab character
- %% “ percent character
- %H “ host name
- %D “ time and date
- %D{*f_char*} “ time and date, format specified by the BSD **strftime** (f_char) time conversion function.
- %h “ host name up to the first."
- %P “ extended command mode
- %p “ command mode
- %r “ redundancy status on modular systems (has no effect on a fixed system)
- %R “ extended redundancy status on modular systems â€“ includes status and slot number (has no effect on a fixed system)

Examples

- This command creates a prompt that displays **system 1** and the command mode.

```
switch(config)# prompt system%s1%P
system 1(config)#
```

- This command creates a prompt that displays the command mode.

```
switch(config)# prompt %p
(config)#
```

- These equivalent commands create the default prompt.

```
% prompt %H%P
host-name.dut103(config)#

%no prompt
host-name.dut103(config)#
```

4.4 Logging of Event Notifications

Arista switches log event notifications using the Syslog protocol. By default, event notifications are logged internally to /var/log/messages, but they can also be displayed on the console or logged to an external server. Severity levels and log message destinations can be configured via the CLI, and individual processes and protocols can also be configured to adjust or limit the messages that they log. Details of the current logging configuration may be viewed using the **`show logging`** command.

For a full list of Syslog messages, visit the Arista website.

4.4.1 Managing TCAM Capacity Warnings

Strata chipsets (present in the 7010, 7050X, 7060X, 7250X, 7260X, and 7300X series) provide event logging for the hardware capacity of TCAM tables on a per-slice basis, triggering a capacity warning by default

whenever any TCAM slice exceeds 90% capacity. As a result, default TCAM logging can generate high levels of syslog messages on these platforms. If this presents a problem, the **hardware capacity alert table** command can be used to adjust the capacity levels at which warnings occur to above the 90% default; this adjustment can be made per TCAM resource and per slice. The command can also be used to disable TCAM hardware capacity messages of level “Warning” and below entirely for a given slice. To disable messages, set the threshold to 0 or use the **no** version of the command.

Examples

- This command reduces hardware capacity Syslog warnings by increasing the capacity threshold to 99% for EFP table monitoring in slice 2.

```
switch(config)#hardware capacity alert table EFP feature Slice-2 threshold
 99
```

- This command reduces messages by disabling hardware capacity Syslog warnings entirely for the IFP table in slice 5.

```
switch(config)#hardware capacity alert table IFP feature Slice-5 threshold 0
```

- This command reduces messages by disabling hardware capacity Syslog warnings entirely for the VFP table in all slices.

```
switch(config)#no hardware capacity alert table VFP
```

Note: Hardware capacity messages are user-configurable only at or below the “Warning” level. The TCAM management software always sends “Error” messages to Syslog and to affected features when all TCAM resources are depleted.

4.5 Event Monitor

The event monitor writes system event records to local files for access by SQLite database commands.

Note:

Beginning with release *EOS-4.20.5F*, **event-monitor** is not enabled by default. Use the **event-monitor** command to explicitly enable event-monitor.

4.5.1 Description

The event monitor receives notifications for important events or changes to the enabled event monitor tables. These changes are logged to a fixed-size circular buffer. The size of this buffer is configurable, but it does not grow dynamically. Buffer contents can be stored to permanent files to increase the event monitor effective capacity. The permanent file size and the number of permanent files is configurable. The buffer is stored at a fixed location on the switch.

Specific event monitor queries are available through CLI commands. For queries not available through specific commands, manual queries are supported through other CLI commands. When the user issues a query command, the relevant events from the circular buffer and permanent files are written to and accessed from a temporary SQLite database file. The database keeps a separate table for each logging type (such as MAC, ARP, route, and others). When the monitor receives notification of a new event, the database file is deleted, then recreated.

4.5.2 Configuring the Event Monitor

Enabling the Event Monitor

The **event-monitor** command enables the event monitor and specifies the types of events that are logged. The event monitor is an event logging service that records system events to a local database. The event monitor records these events:

- all changes to all events.
- ARP changes to the ARP table (IPv4 address to MAC address mappings).
- Neighbor changes to the neighbor table (IPv6 address to MAC address mappings).
- backup backed up log files.
- buffer changes to the local buffer settings.
- IGMP snoopingchanges to the IGMP snooping table.
- LACP changes to the LACP table events.
- MAC changes to the MAC address table (MAC address to port mappings).
- mroute changes to the IP multicast routing table.
- neighbor changes to the neighbor routing table.
- route changes to the IPv4 routing table.
- route6 changes to the IPv6 routing table.
- stpunstable events that cause STP instability.

Beginning with Release *EOS-4.20.5F*, **event-monitor** is not enabled by default. Use the **event-monitor** command to explicitly enable event-monitor.The **no event-monitor all** disables the event monitor. The **no event-monitor** command, followed by a log type parameter, disables event recording for the specified type.

Example

- This command disables the event monitor for all types of events.

```
switch(config)#no event-monitor all
```

- This command enables the event monitor for routing table changes.

```
switch(config)#event-monitor route
```

The **event-monitor clear** command removes the contents of the event monitor buffer. If event monitor backup is enabled, this command removes the contents from all event monitor backup files.

Example

This command clears the contents of the event monitor buffer.

```
switch#event-monitor clear
switch(config)#
```

Configuring the Buffer

The **event-monitor buffer max-size** command specifies the size of the event monitor buffer. The event monitor buffer is a fixed-size circular data structure that receives event records from the event monitor. When event monitor backup is enabled, the buffer is copied to a backup file before each rollover. Buffer size ranges from 6 Kb to 50 Kb. The default size is 32 Kb.

Example

This command configures a buffer size of ***48 Kb***.

```
switch(config)#event-monitor buffer max-size 48
switch(config)#
```

Configuring Permanent Files

The **event-monitor backup path** command enables storage of the event monitor buffer to permanent switch files and specifies the path/name of these files. The command references file location either from the flash drive root directory where the CLI operates (**/mnt/flash**) or from the switch root directory (/).

The event monitor buffer is circular after the buffer is filled, new data replaces older data at the beginning of the buffer. The buffer is copied into a new backup file after each buffer writing cycle before the switch starts re-writing the buffer.

Example

These commands configure the switch to store the event monitor buffer in sw-event.log, then display the new file in the flash directory.

```
switch(config)#event-monitor backup path eventmon_backup_dir/event.log
switch(config)#
bash-4.3# ls /mnt/flash/eventmon_backup_dir/

arpevent.log.1  lacpevent.log.1  neighborevent.log.1  routeevent.log.1
igmpsnoopingevent.log.1  macevent.log.1   route6event.log.1
stpunstableevent.log.1
```

The **event-monitor backup max-size** command specifies the quantity of event monitor backup files the switch maintains. The switch appends an extension number to the file name when it creates a new file. After every 500 events, the switch deletes the oldest backup file if the file limit is exceeded.

Example

These commands configure the switch to back up the event buffer to a series of files named sw-event.log. The switch can store a maximum of four files.

```
switch(config)#event-monitor backup path sw-event.log
switch(config)#event-monitor backup max-size 4
switch(config)#
```

The first five files that the switch creates to store event monitor buffer contents are:

sw-event.log.0

sw-event.log.1

sw-event.log.2

sw-event.log.3

sw-event.log.4

The switch deletes **sw-event.log.0** the first time it verifies the number of existing backup files after the creation of **sw-event.log.4**.

4.5.3 Querying the Event Monitor

These CLI commands perform SQL-style queries on the event monitor database:

- The **show event-monitor arp** command displays ARP table events.
- The **show event-monitor mac** command displays MAC address table events.
- The **show event-monitor route** command displays routing table events.

Example

This command displays all events triggered by MAC address table events.

```
switch#show event-monitor mac
% Writing 0 Arp, 0 Route, 1 Mac events to the database
2012-01-19 13:57:55|1|0808.0808.0808|Ethernet1|configuredStaticMac|added|0
```

For other database queries, the **show event-monitor sqlite** command performs an SQL-style query on the database, using the statement specified in the command.

Example

This command displays all entries from the route table.

```
switch#show event-monitor sqlite select * from route;
2019-09-30 14:01:21.659428|16.16.16.255/32|default|receiveBcast|0|0|updated|20
2019-09-30 14:01:21.659464|192.168.201.12/30|default|connected|1|0|updated|21
2019-09-30 14:01:21.659497|192.168.1.255/32|default|receiveBcast|0|0|updated|22
2019-09-30 14:01:21.659503|192.168.201.8/32|default|receiveBcast|0|0|updated|23
2019-09-30 14:01:21.659512|16.16.16.0/32|default|receiveBcast|0|0|updated|24
2019-09-30
14:01:21.659517|192.168.201.12/32|default|receiveBcast|0|0|updated|25
2019-09-30
14:01:21.659524|192.168.201.15/32|default|receiveBcast|0|0|updated|26
2019-09-30 14:01:21.659541|192.168.201.8/30|default|connected|1|0|updated|27
2019-09-30 14:01:21.659564|16.16.16.0/24|default|connected|1|0|updated|28
2019-09-30 14:01:21.659578|192.168.201.9/32|default|receive|0|0|updated|29
```

4.5.4 Accessing Event Monitor Database Records

The **event-monitor interact** command replaces the CLI prompt with an SQLite prompt. The event monitor buffer and all backup logs are synchronized into a single SQLite file and loaded for access from the prompt.

- To access help from the SQLite prompt, enter **.help**
- To exit SQLite and return to the CLI prompt, enter **.quit** or **.exit**.

The **event-monitor sync** command combines the event monitor buffer and all backup logs and synchronizes them into a single SQLite file. The data can be accessed through SQLite or by using the **show event-monitor** commands described above.

Examples

- This command replaces the EOS CLI prompt with an SQLite prompt.

  ```
  switch#event-monitor interact
  sqlite>
  ```

- This command exits SQLite and returns to the EOS CLI prompt.

  ```
  sqlite> .quit switch#
  ```

- This command synchronizes the buffer and backup logs into a single SQLite file.

  ```
  switch(config)#event-monitor sync
  ```

```
switch(config)#
```

4.6 Managing EOS Extensions

The most simple and efficient way to make the most of the extensibility on which EOS is built is through the use of extensions. An extension is a pre-packaged optional feature or a set of scripts in an RPM Package Manager (RPM) or Software image extension (SWIX) format. A variety of extensions are available from the EOS Central page at https://www.arista.com/en/.

These sections describe basic EOS extension tasks:

- Installing EOS Extensions
- Installing EOS Extensions on a Dual-Supervisor Switch
- Verifying EOS Extensions Installation
- Uninstalling an EOS Extension

4.6.1 Installing EOS Extensions

Complete the following steps to install an EOS extension.

1. Download the desired extension and copy it onto the devices flash storage.

```
switch# dir
Directory of flash:/
-rwx 479183792 Jun 23 09:46 EOS-4.13.3F.swi
-rwx 21280296 Feb 6 16:48 arista-splunk-extension.swix
-rwx 27 Jun 23 10:08 boot-config  drwx 4096 Sep 26 2012 schedule
-rwx 1481 Jun 27 05:54 startup-config
```

2. Copy the file from the flash storage to the extensions partition.

```
switch# copy flash:arista-splunk-extension.swix extension:
Copy completed successfully.
```

3. Install the EOS extension.

```
switch# extension arista-splunk-extension.swix
If this extension modifiers the behavior of the Cli, any running Cli
 sessions will
need to be reset in order for the Cli modifications to take effect.
```

4. If extension persistence across reboots is required, the extension should also be copied into the boot-extensions partition.

```
switch# copy installed-extensions boot-extensions
```

5. Run the extension. As the CloudVision extension adds additional CLI commands to EOS, the CLI session must be restarted so that the additional commands are available. To achieve this, close the SSH or the telnet session and open a new session.

4.6.2 Installing EOS Extensions on a Dual-Supervisor Switch

Complete the following steps to install an EOS extension on a dual-supervisor switch.

1. Copy the extension from the primary supervisor to the standby supervisors flash directory.

```
switch(s1)#copy flash:<filename>.swixsupervisor-peer://mnt/flash/
```

2. Establish a session to the standby supervisor from the primary.

```
switch(s1)#session peer-supervisor
Warning: Permanently added '[127.1.0.2]:3601' (RSA) to the list of known
 hosts.
Last login: Mon Aug 27 17:32:00 2018 from supervisor1
```

```
WARNING - you are currently logged in to the standby supervisor.
Not all cli commands are available or supported. Any configuration
done from this cli will not be reflected in the active supervisor's
running config and will be lost when the active supervisor writes
its startup config.
switch(s2)#
```

3. Repeat the steps listed in the Installing EOS Extensions to install the EOS extension.
4. Repeat the steps listed in the Verifying EOS Extensions Installation to verify the extension installation.
5. Exit standby supervisor.

```
switch(s2)#exit
Connection to 127.1.0.1 closed.
switch(s1)#
```

4.6.3 Verifying EOS Extensions Installation

Complete the steps to verify that the EOS extensions are installed correctly.

1. Run the **`show extensions`** command to verify that the EOS extensions are available and installed correctly.

```
switch#show extensions
Name Version/Release Status Extension
---------------------------------------- -------------------------
EosSdk-1.2.1-fl.boca-1943435.i686.rpm 1.2.1/1943435.flbocaeossd A, NI 1
arista-splunk-extension.swix 0.95/1498976.2013ltdsplun A, I 2
fping-2.4b2-10.fc12.i686.rpm 2.4b2/10.fc12 A, I 1
gnuplot.swix 1.10.0/1.fc14 A, I 16
splunkforwarder-5.0.9-213964.i386.rpm 5.0.9/213964 A, NI 1
```

```
A: available | NA: not available | I: installed | NI: not installed | F:
 forced
```

2. Run the **`show boot-extensions`** command to verify that the EOS extensions are enabled for boot persistence.

```
switch#show boot-extensions
arista-splunk-extension.swix
fping-2.4b2-10.fc12.i686.rpm
gnuplot.swix
```

4.6.4 Uninstalling an EOS Extension

Complete the following steps to uninstall an EOS extension.

1. Uninstall the existing EOS extension using the **`no extension`** command.

```
switch#no extension fping-2.4b2-10.fc12.i686.rpm
switch#show extensions
Name                                     Version/Release             Status extension
---------------------------------------- -------------------------    ----------
EosSdk-1.2.1-fl.boca-1943435.i686.rpm    1.2.1/1943435.flbocaeossd A,  NI 1
arista-splunk-extension.swix             0.95/1498976.2013ltdsplun A,  I 2
```

```
fping-2.4b2-10.fc12.i686.rpm              2.4b2/10.fc12 A,             NI 1
gnuplot.swix                              1.10.0/1.fc14 A,             I 16
splunkforwarder-5.0.9-213964.i386.rpm     5.0.9/213964 A,              NI 1
```

```
A: available | NA: not available | I: installed | NI: not installed | F:
 forced
```

2. Remove the extension from the boot-extension using the **copy installed-extensions boot-extensions** command.

```
switch#copy installed-extensions boot-extensions
```

Note: If your system is a Dual-Supervisor Switch, connect to the secondary supervisor using the session peer-supervisor command, repeat steps 1 and 2, and finally exit from the secondary supervisor.

4.7 Switch Administration Commands

Switch Name Configuration Commands

- clear ptp interface counters
- dns domain
- hostname
- ip domain-list
- ip domain lookup
- ip host
- ip name-server
- ipv6 host
- show hostname
- show hosts
- show ip domain-name
- show ip name-server
- show local-clock time-properties

Banner Configuration Commands

- banner login
- banner motd
- show banner

Prompt Configuration Command

- prompt

Event Manager Commands

- event-monitor
- event-monitor backup max-size
- event-monitor backup path
- event-monitor buffer max-size
- event-monitor clear
- event-monitor interact
- event-monitor sync
- no event-monitor

- show event-monitor arp
- show event-monitor igmpsnooping
- show event-monitor mac
- show event-monitor mroute
- show event-monitor neighbor
- show event-monitor route6
- show event-monitor route
- show event-monitor sqlite
- show event-monitor stpunstable

Email Configuration Command

- email

System Clock Commands

- clock set
- clock timezone
- show clock

NTP Commands

- ntp authenticate
- ntp authentication-key
- ntp local-interface
- ntp serve
- ntp serve all
- ntp server
- ntp trusted-key
- show ntp associations
- show ntp status

Syslog Configuration Commands

- logging format
- logging persistent
- logging repeat-messages

Power Configuration Commands

- power enable module

4.7.1 banner login

The **`banner login`** command configures a message that the switch displays before login and password prompts. The login banner is available on console, telnet, and ssh connections.

The **`no banner login`** and **`default banner login`** commands delete the login banner.

Command Mode

Global Configuration

Command Syntax

```
banner login
```

```
no banner login
```

default banner login

Parameters

- *banner_text* To configure the banner, enter a message when prompted. The message may span multiple lines. Banner text supports the following keywords:
  - **$(hostname)** displays the switchs host name.
- **EOF** To end the banner editing session, type EOF on its own line and press **enter**.

Examples

- These commands create a two-line login banner.

```
switch(config)# banner login
Enter TEXT message. Type 'EOF' on its own line to end.
This is a login banner for $(hostname).
Enter your login name at the prompt.
EOF
switch(config)#
```

- This output displays the login banner.

```
This is a login banner for switch.
Enter your login name at the prompt.
switch login:john
Password:
Last login: Mon Jan 14 09:05:23 2013 from adobe-wrks.aristanetworks.com
 switch>
```

4.7.2 banner motd

The **banner motd** command configures a message of the day (motd) that the switch displays after a user logs in. The motd banner is available on console, telnet, and ssh connections.

The **no banner motd** and **default banner motd** commands delete the motd banner.

Command Mode

Global Configuration

Command Syntax

banner motd

no banner motd

default banner motd

Parameters

- *banner_text* To configure the banner, enter a message when prompted. The message may span multiple lines. Banner text supports this keyword:
  - **$(hostname)** displays the switchs host name.
- **EOF** To end the banner editing session, type EOF on its own line and press **enter**.

Examples

- These commands create an motd banner.

```
switch(config)#banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd banner for $(hostname)
EOF
```

```
switch(config)#
```

- This output displays the motd banner.

```
switch login:john
Password:
Last login: Mon Jan 14 09:17:09 2013 from adobe-wrks.aristanetworks.com
This is an motd banner for Switch switch>
```

4.7.3 clear ptp interface counters

The **clear ptp interface counters** command resets the Precision Time Protocol (PTP) packet counters.

Command Mode

Privileged EXEC

Command Syntax

clear ptp interface [**INTERFACE_NAME**] **counters**

Parameters

INTERFACE_NAME Interface type and numbers. Options include:

- ***no parameter*** Displays information for all interfaces.
  - **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
  - **loopback *l_range*** Loopback interface specified by ***l_range***.
  - **management *m_range*** Management interface range specified by ***m_range***.
  - **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
  - **vlan *v_range*** VLAN interface range specified by ***v_range***.
  - **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.

  Valid parameter formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command clears all PTP counters.

```
switch# clear ptp counters
switch#
```

4.7.4 clock set

The **clock set** command sets the system clock time and date. If the switch is configured with an NTP server, NTP time synchronizations override manually entered time settings.

Time entered by this command is local, as configured by the **clock timezone** command.

Command Mode

Privileged EXEC

Command Syntax

clock set ***hh:mm:ss date***

Parameters

- ***hh:mm:ss*** is the current time (24-hour notation).
- ***date*** is the current date. Date formats include:
  - **mm/dd/yy** example: 05/15/2012

- example: May 15 2012
- example: 15 May 2012

Example

This command manually sets the switch time.

```
switch# clock set 08:15:24 14 Jan 2013
Mon Jan 14 08:15:25 2013
timezone is US/Central
```

4.7.5 clock timezone

The **clock timezone** command specifies the UTC offset that converts system time to local time. The switch uses local time for time displays and to time-stamp system logs and messages.

The **no clock timezone** and **default clock timezone** commands delete the **timezone** statement from ***running-config***, setting local time to UTC.

Command Mode

Global Configuration

Command Syntax

clock timezone ***zone_name***

no clock timezone

default clock timezone

Parameters

zone_name the time zone. Settings include a list of predefined time zone labels.

Examples

- This command configures the switch for the United States Central Time Zone.

```
switch(config)# clock timezone US/Central
switch(config)# show clock
Fri Jan 11 18:42:49 2013
timezone is US/Central
switch(config)#
```

- To view the predefined time zone labels, enter **clock timezone** with a question mark.

```
switch(config)# clock timezone ?
  Africa/Abidjan                    Africa/Accra
  Africa/Addis_Ababa                Africa/Algiers
  Africa/Asmara                     Africa/Asmera
  Africa/Bamako                     Africa/Bangui

  W-SU                              W-SU timezone
  WET                               WET timezone
  Zulu                              Zulu timezone

switch(config)#clock timezone
```

- This command displays all time zone labels that start with **America**.

```
switch(config)# clock timezone AMERICA?
America/Adak                    America/Anchorage
America/Anguilla                America/Antigua
America/Araguaina               America/Argentina/Buenos_Aires
```

```
America/Virgin                  America/Whitehorse
America/Winnipeg                America/Yakutat
America/Yellowknife

switch(config)#clock timezone AMERICA
```

4.7.6 dns domain

The **dns domain** command configures the switchs domain name. The switch uses this name to complete unqualified host names.

The **no dns domain** and **default dns domain** commands delete the domain name by removing the **dns domain** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

dns domain ***string***

no dns domain

default dns domain

Parameters

string domain name (text string).

Example

This command configures ***aristanetworks.com*** as the switchs domain name.

```
switch(config)# dns domain aristanetworks.com
switch(config)#
```

4.7.7 email

The **email** command places the switch in email client configuration mode. If you configure a from-user and an outgoing SMTP server on the switch, you can then use an email address as an output modifier to a **show** command and receive the output as email.

Command Mode

Global Configuration

Command Syntax

email

Example

This command places the switch in email client configuration mode.

```
switch(config)# email
switch(config)#
```

4.7.8 event-monitor backup max-size

The **event-monitor backup max-size** command specifies the quantity of event monitor backup files the switch maintains. Values range from 1 to 200 files with a default of ten files.

The **event-monitor backup path** command specifies the path/name of these files. The switch appends an extension to the file name that tracks the creation order of backup files. When the quantity of files exceeds the configured limit, the switch deletes the oldest file.

The **no event-monitor backup max-size** and **default event-monitor backup max-size** command restores the default maximum number of backup files the switch can store to ten by removing the corresponding **event-monitor backup max-size** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

event-monitor backup max-size ***file_quantity***

no event-monitor backup max-size

default event-monitor backup max-size

Parameters

file_quantity maximum number of backup files. Value ranges from ***1*** to ***200***. Default is ***10***.

Example

These commands configure the switch to back up the event buffer to a series of files named sw-event.log. The switch can store a maximum of four files.

```
switch(config)# event-monitor backup path sw-event.log
switch(config)# event-monitor backup max-size 4
switch(config)#
```

The first five files that the switch creates to store event monitor buffer contents are:

sw-event.log.0

sw-event.log.1

sw-event.log.2

sw-event.log.3

sw-event.log.4

The switch deletes **sw-event.log.0** the first time it verifies the number of existing backup files after the creation of **sw-event.log.4**.

4.7.9 event-monitor backup path

The **event-monitor backup path** command enables the storage of the event monitor buffer to switch files and specifies the path/name of these files. The command references the file location either from the flash drive root directory (/mnt/flash) where the CLI operates or from the switch root directory (/).

The event monitor buffer is circular after the buffer is filled, new data is written to the beginning of the buffer, replacing old data. At the conclusion of each buffer writing cycle, it is copied into a new backup file before the switch starts re-writing the buffer. The switch appends a extension number to the file name when it creates a new file. After every 500 events, the switch deletes the oldest backup file if the file limit specified by the **event-monitor backup max-size** command is exceeded.

running-config can contain a maximum of one **event-monitor backup path** statement. Subsequent **event-monitor backup path** commands replace the existing statement in ***running-config***, changing the name of the file where event monitor backup files are stored.

The **no event-monitor backup path** and **default event-monitor backup path** commands disable the storage of the event monitor buffer to switch files by deleting the **event-monitor backup path** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

event-monitor backup path **URL_FILE**

no event-monitor backup path

default event-monitor backup path

Parameters

URL_FILE path and file name of the backup file:

- ***path_string*** specified path is appended to **/mnt/flash/**.
- **file:** ***path_string*** specified path is appended to /.
- **flash:** ***path_string*** specified path is appended to **/mnt/flash/**.

Example

These commands configure the switch to store the event monitor buffer in sw-event.log, then display the new file in the flash directory.

```
switch(config)# event-monitor backup path eventmon_backup_dir/event.log
switch(config)#
bash-4.3# ls /mnt/flash/eventmon_backup_dir/

arpevent.log.1  lacpevent.log.1  neighborevent.log.1  routeevent.log.1
igmpsnoopingevent.log.1  macevent.log.1   route6event.log.1
stpunstableevent.log.1
```

4.7.10 event-monitor buffer max-size

The **event-monitor buffer max-size** command specifies the size of the event monitor buffer. The event monitor buffer is a fixed-size circular data structure that receives event records from the event monitor. When event monitor backup is enabled (**event-monitor backup path**), the buffer is copied to a backup file before each rollover.

Buffer size ranges from ***6 Kb*** to ***50 Kb***. The default size is ***32 Kb***.

The **no event-monitor buffer max-size** and **default event-monitor buffer max-size** commands restore the default buffer size of 32 Kb by removing the **event-monitor buffer max-size** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

event-monitor buffer max-size ***buffer_size***

no event-monitor buffer max-size

default event-monitor buffer max-size

Parameters

buffer_size buffer capacity (Kb). Values range from ***6*** to ***50***. Default value is ***32***.

Example

This command configures a buffer size of ***48 Kb***.

```
switch(config)# event-monitor buffer max-size 48
switch(config)#
```

4.7.11 event-monitor clear

The **event-monitor clear** command removes the contents of the event monitor buffer. If event monitor backup is enabled, this command removes the contents from all event monitor backup files.

Command Mode

Privileged EXEC

Command Syntax

event-monitor clear

Example

This command clears the contents of the event monitor buffer.

```
switch# event-monitor clear
switch#
```

4.7.12 event-monitor interact

The **event-monitor interact** command replaces the CLI prompt with an SQLite prompt. The event monitor buffer and all backup logs are synchronized into a single SQLite file and loaded for access from the prompt.

- To access help from the SQLite prompt, enter **.help**.
- To exit SQLite and return to the CLI prompt, enter **.quit** or **.exit**.

Command Mode

Privileged EXEC

Command Syntax

event-monitor interact

Examples

- This command replaces the EOS CLI prompt with an SQLite prompt.

```
switch# event-monitor interact
sqlite>
```

- This command exits SQLite and returns to the EOS CLI prompt.

```
sqlite> .quit
switch#
```

4.7.13 event-monitor sync

The **event-monitor buffer sync** command combines the event monitor buffer and all backup logs and synchronizes them into a single SQLite file, which is stored at **/var/log/eventMon.db**.

Command Mode

Privileged EXEC

Command Syntax

event-monitor sync

Example

This command synchronizes the buffer and backup logs into a single SQLite file.

```
switch(config)# event-monitor sync
switch(config)#
```

4.7.14 event-monitor

The **event-monitor** command enables the event monitor and specifies the types of events that are logged. The event monitor is an event logging service that records system events to a local database.

The database maintains a separate table for each event type.

The event monitor is disabled by default.

- The **no event-monitor all** command disables the event monitor.
- The **no event-monitor** command, followed by a log type parameter, disables event recording for the specified type.
- The **event-monitor** command enables the specified event logging type by removing the corresponding **no event-monitor** command from ***running-config***.

The **no event-monitor** and **default event-monitor** commands, without a **LOG_TYPE** parameter, restore the default event monitor settings by deleting all event monitor related commands from ***running-config***.

Command Mode

Global Configuration

Command Syntax

event-monitor LOG_TYPE

no event-monitor LOG_TYPE

default event-monitor LOG_TYPE

Parameters

LOG_TYPE specifies the event logging type. Options include:

- **all** all event logging types.
- **arp** changes to ARP table.
- **backup** backed up log files.
- **buffer** changes to the local buffer settings.
- **igmpsnooping** changes to IGMP snooping table.
- **lacp** changes to the LACP table events.
- **mac** changes to MAC address table.
- **mroute** changes to multicast routing table.
- **neighbor** changes to the neighbor routing table.
- **route** changes to IP routing table.
- **route6** changes to IP route6 table.
- **stpunstable** events that cause STP instability.

Related Command

no event-monitor

Examples

- This command disables the event monitor for all types of events.

```
switch(config)# no event-monitor all
switch(config)#
```

- This command enables the event monitor for routing table changes.

```
switch(config)# event-monitor route
switch(config)#
```

4.7.15 hostname

The **hostname** command assigns a text string as the switchs host name. The default host name is ***localhost***.

The prompt displays the host name when appropriately configured through the **prompt** command.

The **no hostname** and **default hostname** commands return the switchs host name to the default value of ***localhost***.

Command Mode

Global Configuration

Command Syntax

hostname *string*

no hostname

default hostname

Parameter

string host name assigned to the switch.

Example

This command assigns the string ***main-host*** as the switchs host name.

```
switch(config)# hostname main-host
main-host(config)#
```

The prompt was previously configured to display the host name.

4.7.16 ip domain lookup

The **ip domain lookup** command specifies the source interface for all DNS requests sent from the specified VRF.

The **no ip domain lookup** and **default ip domain lookup** commands return the switch to its default state, in which the switch selects source IP addresses for each DNS request from the specified VRF.

Command Mode

Global Configuration

Command Syntax

ip domain lookup [VRF_INSTANCE] source-interface INTF_NAME

no ip domain lookup [**VRF_INSTANCE**] **source-interface**

default ip domain lookup [**VRF_INSTANCE**] **source-interface**

Parameters

- **VRF_INSTANCE** specifies the VRF instance being modified.

- ***no parameter*** changes are made to the default VRF.
- **vrf *vrf_name*** changes are made to the specified VRF.

- **INTF_NAME** name of source interface to be used for DNS requests. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - management ***m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - vlan ***v_num*** VLAN interface specified by ***v_num***.

Examples

- This command specifies ***VLAN 5*** as the source interface for DNS requests originating from the default VRF.

```
switch(config)# ip domain lookup source-interface Vlan5
switch(config)#
```

- This command specifies ***VLAN 10*** as the source interface for DNS requests originating from VRF purple.

```
switch(config)# ip domain lookup vrf purple source-interface Vlan10
switch(config)#
```

4.7.17 ip domain-list

The **`ip domain-list`** command specifies a domain name to add to the IP domain list.

The **`no ip domain-list`** and **`default ip domain-list`** commands return the IP domain list to its default state, in which the switch selects source IP addresses for each DNS request from the specified VRF.

Command Mode

Global Configuration

Command Syntax

`ip domain-list` [**IP_DOMAIN_NAME**]

`no ip domain-list` [**IP_DOMAIN_NAME**]

`default ip domain-list` [**IP_DOMAIN_NAME**]

Parameter

IP_DOMAIN_NAME specifies the IP domain name.

Examples

- This command specifies foo.com as the IP domain name to add to the IP domain list.

```
switch(config)# ip domain-list foo.com
switch(config)#
```

- This command removes foo.com and returns the IP domain list to its default state.

```
switch(config)# no ip domain-list foo.com
switch(config)#
```

4.7.18 ip host

The **`ip host`** command associates a hostname to an IPv4 address. This command supports local hostname resolution based on local hostname-IP address maps. Multiple hostnames can be mapped to an IP address.

IPv4 and IPv6 addresses can be mapped to the same hostname (to map an IPv6 address to a hostname, use the **ipv6 host** command). The **show hosts** command displays the local hostname-IP address mappings.

The **no ip host** and **default ip host** commands removes hostname-IP address maps by deleting the corresponding **ip host** command from ***running-config***, as specified by command parameters:

- ***no parameters***: command removes all hostname-IP address maps.
- ***hostname*** parameter: command removes all IP address maps for the specified hostname.
- ***hostname*** and ***IP address*** parameters: command removes specified hostname-IP address maps.

Command Mode

Global Configuration

Command Syntax

ip host ***hostname hostadd_1*** [***hostadd_2***] ...[***hostadd_X***]

no ip host [***hostname***] [***hostadd_1*** [***hostadd_2***] [***hostadd_X***]

default ip host [***hostname***] [***hostadd_1*** [***hostadd_2***] [***hostadd_X***]

Parameters

- ***hostname*** hostname (text).
- ***hostadd_N*** IPv4 address associated with hostname (dotted decimal notation).

Related Commands

- **ipv6 host**
- **show hosts**

Examples

- This command associates the hostname ***test_lab*** with the IP addresses ***10.24.18.5*** and ***10.24.16.3***.

```
switch(config)# ip host test_lab 10.24.18.5 10.24.16.3
```

- This command removes all IP address maps for the hostname ***production_lab***.

```
switch(config)# no ip host production_lab
switch(config)#
```

4.7.19 ip name-server

The **ip name-server** command adds name server addresses to ***running_config***. The switch uses name servers for name and address resolution. The switch can be configured with up to three name servers. Although a command can specify multiple name server addresses, ***running_config*** stores each address in a separate statement. Name server addresses can be IPv4 and IPv6; each command can specify both address types.

Attempts to add a fourth server generate an error message. All name server addresses must be configured in the same VRF. When name servers were previously configured in a VRF, they must all be removed before adding new name server entries.

The **no ip name-server** and **default ip name-server** commands remove specified name servers from ***running_config***. Commands that do not list an address remove all name servers.

Command Mode

Global Configuration

Command Syntax

ip name-server [**VRF_INSTANCE**] [***SERVER_1***] [***SERVER_2***] [***SERVER_3***]

no ip name-server [**VRF_INSTANCE**] [***SERVER_1***] [***SERVER_2***] [***SERVER_3***]

default ip name-server [**VRF_INSTANCE**] [***SERVER_1***] [***SERVER_2***] [***SERVER_3***]

Parameters

- **VRF_INSTANCE** specifies the VRF instance containing the addresses.
  - ***no parameter*** default VRF.
  - **vrf *vrf_name*** a user-defined VRF.
- **SERVER_X** IP address of the name server (dotted decimal notation). Options include:
  - ***ipv4_addr*** (A.B.C.D)
  - ***ipv6_addr*** (A:B:C:D:E:F:G:H)

  A command can contain both (IPv4 and IPv6) address types.

Guidelines

All configured name server addresses must come from the same VRF. To use a user defined VRF for connection to a name server, first remove any name servers configured in the default VRF.

Examples

- This command adds two name servers to the configuration.

```
switch(config)# ip name-server 172.0.14.21 3:4F21:1902::
switch(config)#
```

- This command attempts to add a name server when the configuration already lists three servers.

```
switch(config)# ip name-server 172.1.10.22
% Maximum number of nameservers reached. '172.1.10.22' not added
switch(config)#
```

4.7.20 ipv6 host

The **ipv6 host** command associates a hostname to an IPv6 address. This command supports local hostname resolution based on local hostname-IP address maps. Multiple hostnames can be mapped to an IPv6 address. IPv4 and IPv6 addresses can be mapped to the same hostname (to map IPv4 addresses to a hostname, use the **ip host** command). The **show hosts** command displays the local hostname-IP address mappings.

The **no ipv6 host** and **default ipv6 host** commands remove hostname-IP address maps by deleting the corresponding **ipv6 host** command from ***running-config***, as specified by command parameters:

- ***no parameters***: command removes all hostname-IPv6 address maps.
- ***hostname*** parameter: command removes all IPv6 address maps for the specified hostname.
- ***hostname*** and ***IP address*** parameters: command removes specified hostname-IP address maps.

Command Mode

Global Configuration

Command Syntax

ipv6 host ***hostname hostadd_1*** [***hostadd_2***] ...[***hostadd_X***]

no ipv6 host [***hostname***] [***hostadd_1*** [***hostadd_2***] [***hostadd_X***]

default ipv6 host [***hostname***] [***hostadd_1*** [***hostadd_2***] [***hostadd_X***]

Parameters

- ***hostname*** hostname (text).
- ***hostadd_N*** IPv6 addresses associated with hostname (dotted decimal notation).

Related Commands

- **ip host**
- **show hosts**

Example

This command associates the hostname ***support_lab*** with the IPv6 address ***2001:0DB8:73:ff:ff:26:fd:90***.

```
switch(config)# ipv6 host support_lab 2001:0DB8:73:ff:ff:26:fd:90
switch(config)#
```

4.7.21 logging format

The **logging format** command configures formatting options for syslog messages.

The **no logging format** and **default logging format** commands remove the corresponding **logging format** command from ***running-config*** and restore the specified formatting to its default setting.

Command Mode

Global Configuration

Command Syntax

`logging format` {hostname{fqdn|ipv4}|rfc5424|sequence-numbers|timestamp{high-resolution|traditional[timezone][year]}}

`no logging format` {hostname|rfc5424|sequence-numbers|timestamp}

`default logging format` {hostname|rfc5424|sequence-numbers|timestamp}

Parameters

hostname {fqdn\ipv4} specifies the formatting for the hostname in syslog messages as either **fqdn** (fully qualified domain name) or **ipvr** (IPv4 address).

rfc5424 causes syslogs generated locally to include high-resolution timestamps, and syslogs forwarded to remote servers to be sent in RFC5424 format.

sequence-numbers causes the sequence numbers of syslog messages to be visible when the messages are displayed.

timestamp {high-resolution|traditional[timezone][year]} specifies the formatting for syslog timestamps as either **high-resolution** (high-resolution RFC3339 timestamps) or **traditional** (traditional syslog timestamps as specified in RFC3164). When using the traditional timestamp format, **timezone** and **year** can also be included.

Examples

- This command enables sequence numbering that can been seen when Syslog messages are displayed.

```
switch(config)# logging format sequence-numbers
switch(config)#
```

- To display the sequence numbers, issue the **show logging** command.

```
switch# show logging
    Syslog logging: enabled
    Buffer logging: level debugging
    Console logging: level informational
    Synchronous logging: disabled
    Trap logging: level informational
    Sequence numbers: enabled
    Syslog facility: local4
    Hostname format: Hostname only
    Repeat logging interval: disabled

Log Buffer:
```

```
Nov 12 14:03:34 switch1 SuperServer: 1: %SYS-7-CLI_SCHEDULER_LOG_STORED: Logfile for scheduled
 CLI execution job 'tech-support' is stored in
flash:/schedule/tech-support/tech-support_2012-11-12.1402.log.gz
Nov 12 14:06:52 switch1 Cli: 2: %SYS-5-CONFIG_I: Configured from console by admin on con0
 (0.0.0.0)
Nov 12 14:07:26 switch1 Cli: 3: %SYS-5-CONFIG_E: Enter configuration mode from console by admin
 on con0 (0.0.0.0)
Nov 12 14:14:29 switch1 Cli: 4: %SYS-5-CONFIG_I: Configured from console by admin on con0
 (0.0.0.0)
Nov 12 14:15:55 switch1 Cli: 5: %SYS-5-CONFIG_E: Enter configuration mode from console by admin
 on con0 (0.0.0.0)
Nov 12 14:33:05 switch1 Cli: 6: %SYS-5-CONFIG_I: Configured from console by admin on con0
 (0.0.0.0)
Nov 12 14:45:13 switch1 Cli: 7: %SYS-5-CONFIG_E: Enter configuration mode from console by admin
 on con0 (0.0.0.0)
switch#
```

4.7.22 logging persistent

The **`logging persistent`** command logs the files stored on the flash disk.

The **`no logging persistent`** command disables the logging from the ***running-config***.

Command Mode

Global Configuration Mode

Command Syntax

`logging persistent` ***logging file size***

`no logging persistent` ***logging file size***

Parameter

- ***logging file size*** The maximum size (in bytes) of logging file stored on flash disk. The value ranges from 1024 to 2147483647.

Example

- This command configures logging persistent on the switch.

```
switch# config
switch(config)# logging persistent 1024
! Note: writing system log message on non-volatile flash will affect the
 life
expectancy of the flash drive due to heavy writing. Please disable
 persistent logging unless needed.
```

4.7.23 logging repeat-messages

The **`logging repeat-messages`** command configures repetition of syslog messages instead of summarizing the count of repeats.

The **`no logging repeat-messages`** and **`default logging repeat-messages`** commands disable the functionality to repeat logging messages in ***running-config***.

Command Mode

Global Configuration

Command Syntax

`logging repeat-messages`

`no logging repeat-messages`

`default logging repeat-messages`

Examples

- This command repeats syslog messages instead of summarizing the count of repeats.

```
switch(config)# logging repeat-messages
switch(config)#
```

- This command displays the status of logging repeat messages command.

```
switch(config)# show logging
Syslog logging: enabled
    Buffer logging: level debugging
    Console logging: level debugging
    Monitor logging: level debugging
    Synchronous logging: disabled
    Trap logging: level informational
    Sequence numbers: disabled
    Syslog facility: local4
    Hostname format: Hostname only
    Repeat logging interval: disabled
    Repeat messages: enabled

Facility             Severity            Effective Severity
--------------       -------------       ------------------
aaa                  debugging           debugging
accounting           debugging           debugging

switch(config)#
```

4.7.24 no event-monitor

The **`no event-monitor`** and **`default event-monitor`** commands remove the specified **event-monitor** configuration statements from ***running-config***, returning the switch to the specified default state.

- **`no event-monitor`** with ***no parameters***, restores all default setting states:
  - event monitor is enabled.
  - buffer backup is disabled.
- The **`no event-monitor backup`** disables the backup.

To disable the event monitor, enter the **`no event-monitor all`** command (**`event-monitor`**).

Command Mode

Global Configuration

Command Syntax

`no event-monitor` [**PARAMETER**]

`default event-monitor` [**PARAMETER**]

Parameters

PARAMETER the event monitor property that is returned to the default state.

- ***no parameter*** all event monitor properties.
- **backup** event monitor buffer backup is disabled.

Example

This command removes all event monitor configuration statements from ***running-config***.

```
switch(config)# no event-monitor
switch(config)#
```

4.7.25 ntp authenticate

The **ntp authenticate** command enables the authentication of incoming NTP packets. When authentication is enabled, NTP packets will be used to synchronize time on the switch only if they include a trusted authentication key. Authentication keys are created on the switch using the **ntp authentication-key** command, and the **ntp trusted-key** command is used to specify which keys are trusted. NTP authentication is disabled by default.

The **no ntp authenticate** and **default ntp authenticate** commands disable NTP authentication on the switch by removing the corresponding **ntp authenticate** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ntp authenticate

no ntp authenticate

default ntp authenticate

Examples

- This command enables NTP authentication on the switch.

```
switch(config)# ntp authenticate
switch(config)#
```

- This command disables NTP authentication on the switch.

```
switch(config)# no ntp authenticate
switch(config)#
```

4.7.26 ntp authentication-key

The **ntp authentication-key** command creates an authentication key for use in authenticating incoming NTP packets. For the key to be used in authentication:

- It must be configured as a trusted key using the **ntp trusted-key** command.
- NTP authentication must be enabled on the switch using the **ntp authenticate** command.
- The same key must be configured on the NTP server.

The **no ntp authentication-key** and **default ntp authentication-key** commands remove the specified authentication key by removing the corresponding **ntp authentication-key** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ntp authentication-key*key_id* **ENCRYPT_TYPE** ***password_text***

no ntp authentication-key ***key_id***

default ntp authentication-key ***key_id***

Parameters

- ***key_id*** key ID number. Value ranges from ***1*** to ***65534***.
- **ENCRYPT_TYPE** encryption method. Values include:
  - **md5** ***key_text*** is MD5 encrypted.

- **sha1** ***key_text*** is SHA-1 encrypted.
- ***password_text*** the authentication-key password.

Examples

- This command creates an NTP authentication key with ***ID 234*** and password ***timeSync*** using MD5 encryption.

```
switch(config)# ntp authentication-key 234 md5 timeSync
```

  Running-config stores the password as plain text.
- This command removes ***NTP authentication key 234***.

```
switch(config)# no ntp authentication-key 234
```

4.7.27 ntp local-interface

The **ntp local-interface** command configures an interface as the local NTP source. The IP address of that interface will then be used as the source address in NTP packets sent by the switch. If the switch is acting as an NTP server and a server-specific source interface has been configured using the **source** option of the **ntp server** command, the server-specific source address will take precedence.

The **no ntp local-interface** and **default ntp local-interface** commands remove the **ntp local-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ntp local-interface [**VRF_INSTANCE**] **INT_PORT**

no ntp local-interface

default ntp local-interface

Parameters

- **VRF_INSTANCE** the VRF instance to be used for connection to the specified server. Options include:
  - ***no parameter*** connects using the default VRF.
  - **vrf** ***vrf_name*** connects using the specified user-defined VRF.
- **INT_PORT** the interface port that specifies the NTP local interface. Settings include:
  - **ethernet** ***e_range*** Ethernet interface list.
  - **loopback** ***l_range*** loopback interface list.
  - **management** ***m_range*** management interface list.
  - **port-channel** ***c_range*** port channel interface list.
  - **vlan** ***v_range*** VLAN interface list.

Examples

- This command configures ***ntp local-interface vlan 25*** as the local NTP source. NTP packets exiting the switch use the IP address of ***VLAN interface 25*** as their source address.

```
switch(config)# ntp local-interface vlan 25
switch(config)#
```

- This command removes the **ntp local-interface** command from the configuration.

```
switch(config)# no ntp local-interface
```

```
switch(config)#
```

4.7.28 ntp serve all

The **ntp serve all** command configures the switch to act as an NTP server by accepting incoming NTP requests.

Using this command also causes the switch to re-synchronize with its upstream NTP server.

Individual interfaces can be configured separately to accept or deny NTP requests by using the **ntp serve** command, and these settings override the global setting.

Command Mode

Global Configuration

Command Syntax

ntp serve all

no ntp serve all

default ntp serve all

Example

- This command configures the switch to accept incoming NTP requests.

```
switch(config)# ntp serve all
switch(config)#
```

- This command configures the switch to deny incoming NTP requests.

```
switch(config)# no ntp serve all
switch(config)#
```

4.7.29 ntp serve

The **ntp serve** command configures the command mode interface to accept incoming NTP requests regardless of the global setting.

The **no ntp serve** command configures the command mode interface to refuse incoming NTP requests regardless of the global setting. The **default ntp serve** command configures the command mode interface to follow the global setting.

Using this command also causes the switch to re-synchronize with its upstream NTP server.

Command Modes

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Interface-VXLAN Configuration

Command Syntax

ntp serve

no ntp serve

`default ntp serve`

Examples

- These commands configure ***interface ethernet 5*** to accept incoming NTP requests regardless of global settings.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)#ntp serve
switch(config-if-Et5)#
```

- These commands configure ***interface ethernet 5*** to deny incoming NTP requests regardless of global settings.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)#no ntp serve
switch(config-if-Et5)#
```

- These commands configure ***interface ethernet 5*** to use global settings in responding to incoming NTP requests.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)#default ntp serve
switch(config-if-Et5)#
```

4.7.30 ntp server

The **`ntp server`** command adds a Network Time Protocol (NTP) server to ***running-config***. If the command specifies a server that already exists in ***running-config***, it will modify the server settings. The switch synchronizes the system clock with an NTP server when ***running-config*** contains at least one valid NTP server.

The switch supports NTP versions 1 through 4. The default is version 4.

The **prefer** option specifies a preferred NTP server, which is used as the NTP server if not discarded by NTP.

The **`no ntp server`** and **`default ntp server`** commands remove the specified NTP server from ***running-config***. To remove an NTP server configured in a user-defined VRF, include the VRF name in the **`no ntp server`** command.

Command Mode

Global Configuration

Command Syntax

`ntp server` [**VRF_INSTANCE**] **SERVER_NAME** [**PREFERENCE**] [**NTP_VERSION**] [**IP_SOURCE**] [**burst**] [**iburst**] [**AUTH_KEY**][**MAX_POLL_INT**][**MIN_POLL_INT**]

`no ntp server` [**VRF_INSTANCE**] **SERVER_NAME**

`default ntp server` [**VRF_INSTANCE**] **SERVER_NAME**

All parameters except **VRF_INSTANCE** and **SERVER_NAME** can be placed in any order.

Parameters

- **VRF_INSTANCE** the VRF instance to be used for connection to the specified server.
  - ***no parameter*** connects using the default VRF.
  - **vrf *vrf_name*** connects using the specified user-defined VRF.
- **SERVER_NAME** NTP server location. Options include:
  - ***IP address*** in dotted decimal notation.
- **PREFERENCE** indicates priority of this server when the switch selects a synchronizing server.

  - ***no parameter*** server has no special priority.
  - **prefer** server has priority when the switch selects a synchronizing server.
- **NTP_VERSION** specifies the NTP version. Settings include:
  - ***no parameter*** sets NTP version to 4 (default).
  - **version *number***, where ***number*** ranges from ***1*** to ***4***.
- **IP_SOURCE** specifies the source interface for NTP updates for the specified NTP server. This option overrides global settings created by the **`ntp local-interface`** command. Options include:
  - ***no parameter*** sets the source interface to the global default.
  - **source ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **source loopback *l_num*** loopback interface specified by ***l_num***.
  - **source management *m_num*** management interface specified by ***m_num***.
  - **source port-channel *p_num*** port-channel interface specified by ***p_num***.
  - **source vlan *v_num*** VLAN interface specified by ***v_num***.
- **burst** indicates that when the NTP server is reached, the switch sends packets to the server in bursts of eight instead of the usual one. Recommended only for local servers. Off by default.
- **iburst** indicates that the switch sends packets to the server in bursts of eight instead of the usual one until the server is reached. Recommended for general use to speed synchronization. Off by default.
- **AUTH_KEY** the authentication key to use in authenticating NTP packets from the server.
  - ***no parameter*** no authentication key is specified.
  - **key *1 to 65534*** switch will use the specified key to authenticate NTP packets from the server.
- **MAX_POLL_INT** specifies the maximum polling interval for the server (as the base-2 logarithm of the interval in seconds). Settings include:
  - ***no parameter*** sets the maximum polling interval to ***10*** (***1,024*** seconds, the default).
  - **maxpoll *number***, where ***number*** is the base-2 logarithm of the interval in seconds. Values range from ***3*** (***8*** seconds) to ***17*** (***131,072*** seconds, approximately ***36*** hours).
- **MIN_POLL_INT** specifies the minimum polling interval for the server (as the base-2 logarithm of the interval in seconds). Settings include:
  - ***no parameter*** sets the minimum polling interval to ***6*** (***64*** seconds, the default).
  - **minpoll *number***, where ***number*** is the base-2 logarithm of the interval in seconds. Values range from ***3*** (***8*** seconds) to ***17*** (***131,072*** seconds, approximately ***36*** hours).

Guidelines

To configure multiple parameters for a single server, include them all in a single **`ntp server`** command. Using the command again for the same server overwrites parameters previously configured in ***running-config***.

All NTP servers must use the same VRF. If no VRF is specified, the server is configured in the default VRF. To use a user-defined VRF for connection to an NTP server, first use the **`no ntp server`** command to remove any NTP servers configured in the default VRF.

When specifying a source interface, choose an interface in the same VRF as the server. If the source interface is not in the same VRF, the source data will be included in ***running-config*** but will not be added to NTP packets.

An NTP server may be configured using an invalid or inactive VRF, but the status of the NTP server will remain inactive until the VRF is active.

Examples

- This command configures the switch to update its time with the NTP server at address ***172.16.0.23*** and designates it as a preferred NTP server.

```
switch(config)# ntp server 172.16.0.23 prefer
```

- This command configures the switch to update its time through an NTP server named ***local-nettime***.

```
switch(config)# ntp server local-nettime
```

- This command configures the switch to update its time through a ***version 3*** NTP server.

```
switch(config)# ntp server 171.18.1.22 version 3
```

- These commands reconfigure the switch to access the above NTP servers through VRF magenta.

```
switch(config)# no ntp server 172.16.0.23
switch(config)# no ntp server local-nettime
switch(config)# no ntp server 171.18.1.22
switch(config)# ntp server vrf magenta 172.16.0.23 prefer
switch(config)# ntp server vrf magenta local-nettime
switch(config)# ntp server vrf magenta 171.18.1.22 version 3
switch(config)#
```

4.7.31 ntp trusted-key

The **ntp trusted-key** command specifies which authentication keys will be trusted for authentication of NTP packets. A packet with a trusted key will be used to update the local time if authenticated.

The **no ntp trusted-key** and **default ntp trusted-key** commands remove the specified authentication keys from the trusted key list by removing the corresponding **ntp trusted-key** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ntp trusted-key **key_list**

no ntp trusted-key

default ntp trusted-key

Parameters

key_list specified one or more keys. Formats include a number (***1*** to ***65534***), number range, or comma-delimited list of numbers and ranges.

Example

This command configures the switch to trust authentication keys ***234*** and ***237*** for authentication of NTP packets.

```
switch(config)# ntp trusted-key 234,237
switch(config)#
```

4.7.32 power enable module

The **power enable module** command powers up the specified module. The **no power enable module** command powers down the specified module.

Command Mode

Global Configuration

Command Syntax

power enable module {**fabric**|**linecard**|**supervisor**|**switchcard**} ***module_number***

`no power enable module` {**fabric**|**linecard**|**supervisor**|**switchcard**} ***module_number***

`default power enable module` {**fabric**|**linecard**|**supervisor**|**switchcard**} ***module_number***

Parameters

fabric specifies a fabric card

linecard specifies a linecard

supervisor specifies a supervisor

switchcard specifies a switch card

module_number specifies the number of the module

Examples

- This command powers down linecard 3.

```
switch(config)# no power enable module linecard 3
switch(config)#
```

- These commands reload fabric module 2.

```
switch(config)# no power enable module fabric 2
switch(config)#power enable module fabric 2
switch(config)#
```

4.7.33 prompt

The **`prompt`** command specifies the contents of the CLI prompt. Characters allowed in the prompt include A-Z, a-z, 0-9, and these punctuation marks:

! @ # $ % # & * ( ) - = + f g [ ] ; : < > , . ? / # n

The prompt supports these control sequences:

- %s – space character
- %t – tab character
- %% – percent character
- %D – time and date
- %D{*f_char*} – time and date, format specified by the BSD **strftime** (f_char) time conversion function.
- %H – host name
- %h – host name up to the first '.'
- %P – extended command mode
- %p – command mode
- %r [1] – redundancy status on modular systems.
- %R [2] – extended redundancy status on modular systems – includes status and slot number.

[1] When logged into a fixed system or a supervisor on a modular system, this option has no effect.
[2] When logged into a fixed system, this option has no effect.

Table 4: Command Mode Prompt Examples

Command Mode	Command Mode Prompt	Extended Command Mode Prompt
Exec	>	>
Privileged Exec	#	#
Global Configuration	(config)#	(config)#
Ethernet Interface Configuration	(config-if)#	(config-if-ET15)#
VLAN Interface Configuration	(config-if)#	(config-if-Vl24)#
Port Channel Interface Configuration	(config-if)#	(config-if-Po4)#
Management Interface Configuration	(config-if)#	(config-if-Ma1)
Access List Configuration	(config-acl)#	(config-acl-listname)#
OSPF Configuration	(config-router)#	(config-router-ospf)#
BGP Configuration	(config-router)#	(config-router-bgp)#

The **`no prompt`** and **`default prompt`** commands return the prompt to the default of ***%H%R%P***.

Command Mode

Global Configuration

Command Syntax

`prompt` ***p_string***

`no prompt`

`default prompt`

Parameters

p_string prompt text (character string). Elements include letters, numbers, and control sequences.

Examples

- This command creates a prompt that displays ***system 1*** and the command mode.

```
host-name.dut103(config)# prompt system%s1%P
system 1(config)#
```

- This command creates a prompt that displays the command mode.

```
host-name.dut103(config)# prompt %p
(config)#
```

- These equivalent commands create the default prompt.

```
% prompt %H%P
host-name.dut103(config)#
```

```
% no prompt
host-name.dut103(config)#
```

4.7.34 show banner

The **show banner** command displays the specified banner.

Command Mode

Privileged EXEC

Command Syntax

show banner BANNER_TYPE [login | motd]

Parameters

BANNER_TYPE banner that the command displays. Options include:

- **login** command displays login banner.
- **motd** command displays message of the day banner.

Example

These commands configure and display the message of the day banner.

```
switch(config)# banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd banner for $(hostname)
EOF

switch(config)# show banner motd
This is an motd banner for $(hostname)
switch(config)#
```

4.7.35 show clock

The **show clock** command displays the current system clock time and configured time zone. The switch uses the system clock for system log messages and debugging traces.

Command Mode

EXEC

Command Syntax

show clock

Example

This command displays the current system clock time and configured time zone.

```
switch> show clock
Wed Nov  2 10:29:32 2011
timezone is America/Los_Angeles switch>
```

4.7.36 show event-monitor arp

The **`show event-monitor arp`** command performs an SQL-style query on the event monitor database and displays ARP table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

`show event-monitor arp` [**GROUP**] [**MESSAGES**] [**INTERFACE**] [**IP**] [**MAC**] [**TIME**] [**VRF**]

Optional parameters can be placed in any order.

Parameters

- **GROUP** used with aggregate functions to group results. Analogous to SQL **group by** command.
  - ***no parameter*** results are not grouped.
  - **group by interface** results are grouped by interface.
  - **group by ip** results are grouped by IP address.
  - **group by mac** results are grouped by MAC address.
  - **group by vrf** results are grouped by VRF.
- **MESSAGES** number of messages returned from query. Analogous to SQL **limit** command.
  - ***no parameter*** result-set size is not limited.
  - **limit *msg_quantity*** number of results that are displayed. Values range from ***1*** to ***15,000***.
- **INTERFACE** restricts result-set to events that include specified interface (SQL Like command).
  - ***no parameter*** result-set not restricted by interface.
  - **match-interface ethernet *e_range*** Ethernet interface list.
  - **match-interface loopback *l_range*** loopback interface list.
  - **match-interface management *m_range*** management interface list.
  - **match-interface port-channel *c_range*** port channel interface list.
  - **match-interface tunnel *t_range*** tunnel interface list.
  - **match-interface vxlan *vx_range*** VXLAN interface list.
  - **match-interface port-channel *c_range*** port channel interface list.
- **IP** restricts result-set to events that include specified IP address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific IP addresses.
  - **match-ip *ip_address_rex*** IP address, as represented by regular expression.
- **MAC** restricts result-set to events that include specified MAC address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific MAC addresses.
  - **match-mac *mac_address_rex*** MAC address, as represented by regular expression.
- **TIME** restricts result-set to events generated during specified period.
  - ***no parameter*** result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.
- **VRF** restricts result-set to events that include a specific VRF.
  - ***no parameter*** result-set not restricted by time of event.
  - **match-vrf *vrf_name*** the VRF name.

Example

This command displays ARP table events listed in the event monitor database.

```
switch# show event-monitor arp
% Writing 220017 Arp, 234204 Route, 1732559 Mac events to the database
2012-11-06 12:36:10|10.33.6.159|Vlan1417|0000.00dc.cc0d|0|added|2186271
2012-11-06 12:38:20|10.33.7.150|Vlan1417|0000.00f7.e25f|0|added|2186292
2012-11-06 12:38:34|10.33.6.62|Vlan1417|0000:00:01:c2:ac|0|added|2186295
2012-11-06 12:39:13|10.33.7.162|Vlan1417|00:00:00:45:c2:79|0|added|2186299
2012-11-06 12:39:50|10.33.12.54|Vlan1417|||removed|2186303
2012-11-06 12:39:51|10.33.6.218|Vlan1417|00:00:00:e9:36:46|0|added|2186305
2012-11-06 12:40:00|10.33.6.140|Vlan1417|00:00:00:4a:36:c3|0|added|2186308
2012-11-06 12:40:02|10.33.6.239|Vlan1417|00:00:00:5b:a7:21|0|added|2186312
2012-11-06 12:41:16|10.33.7.11|Vlan1417|00:00:00:3f:94:59|0|added|2186320
2012-11-06 12:41:50|10.33.7.60|Vlan1417|00:00:00:1f:3c:8e|0|added|2186346
2012-11-06 12:43:34|10.33.7.81|Vlan1417|00:00:00:e3:0d:9c|0|added|2186762
2012-11-06 12:43:42|10.33.6.214|Vlan1417|00:00:00:7b:09:7d|0|added|2186765
2012-11-06 12:43:59|10.33.7.149|Vlan1417|00:00:00:8d:a6:d8|0|added|2186768
 switch#
```

4.7.37 show event-monitor igmpsnooping

The **show event-monitor igmpsnooping** command performs an SQL-style query on the event-monitor database, using the statement specified in the command.

Command Mode

Privileged EXEC

Command Syntax

show event-monitor igmpsnooping [**GROUP**] [**MESSAGES**] [**MAC**] [**INTERFACE**] [**VLAN**] [**TIME**]

Parameters

- **GROUP** used with aggregate functions to group results. Analogous to SQL **group by** command.
  - ***no parameter*** results are not grouped.
  - **group-by interface** results are grouped by interface.
  - **group-by mac** results are grouped by MAC address.
  - **group-by vlan** results are grouped by VLAN.
- **MESSAGES** number of messages returned from query. Analogous to SQL **limit** command.
  - ***no parameter*** result-set size is not limited.
  - **limit *msg_quantity*** number of results that are displayed. Values range from ***1*** to ***15,000***.
- **MAC** restricts result-set to events that include specified MAC address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific MAC addresses.
  - **match-mac *mac_address_rex*** MAC address, as represented by regular expression.
- **INTERFACE** restricts result-set to events that include specified interface (SQL Like command).
  - ***no parameter*** result-set not restricted by interface.
  - **match-interface ethernet *e_range*** Ethernet interface list.
  - **match-interface loopback *l_range*** loopback interface list.
  - **match-interface management *m_range*** management interface list.
  - **match-interface port-channel *c_range*** port channel interface list.
  - **match-interface vlan *v_range*** VLAN interface list.
  - **match-interface tunnel *t_range*** tunnel interface list.
  - **match-interface vxlan *vx_range*** VXLAN interface list.
- **TIME** restricts result-set to events with specified period.
  - ***no parameter*** result-set not restricted by time of event.

- **match-time last-minute** includes events generated during last minute.
- **match-time last-day** includes events generated during last day.
- **match-time last-hour** includes events generated during last hour.
- **match-time last-week** includes events generated during last week.

- **VLAN** restricts result-set to events that include a specific VLAN (SQL Like command).
  - ***no parameter*** result-set not restricted by time of event.
  - **match-vlan vlan** VLAN interface number.

Example

```
switch# show event-monitor igmpsnooping
switch#
```

4.7.38 show event-monitor mac

The **show event-monitor mac** command performs an SQL-style query on the event monitor database and displays MAC address table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

show event-monitor mac [**GROUP**] [**MESSAGES**] [**INTERFACE**] [**MAC**] [**TIME**]

Optional parameters can be placed in any order.

Parameters

- **GROUP** used with aggregate functions to group results. Analogous to SQL **group by** command.
  - ***no parameter*** results are not grouped.
  - **group-by interface** results are grouped by interface.
  - **group-by mac** results are grouped by MAC address.
- **MESSAGES** number of messages returned from query. Analogous to SQL **limit** command.
  - ***no parameter*** result-set size is not limited.
  - **limit *msg_quantity*** number of results that are displayed. Values range from ***1*** to ***15,000***.
- **INTERFACE** restricts result-set to events that include specified interface (SQL Like command).
  - ***no parameter*** result-set not restricted by interface.
  - **match-interface ethernet *e_range*** Ethernet interface list.
  - **match-interface loopback *l_range*** loopback interface list.
  - **match-interface management *m_range*** management interface list.
  - **match-interface port-channel *c_range*** port channel interface list.
  - **match-interface vlan *v_range*** VLAN interface list.
  - **match-interface tunnel *t_range*** tunnel interface list.
  - **match-interface vxlan *vx_range*** VXLAN interface list.
- **MAC** restricts result-set to events that include specified MAC address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific MAC addresses.
  - **match-mac *mac_address_rex*** MAC address, as represented by regular expression.
- **TIME** restricts result-set to events with specified period.
  - ***no parameter*** result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.

- **match-time last-hour** includes events generated during last hour.
- **match-time last-week** includes events generated during last week.

Examples

- This command displays all events triggered by MAC address table events.

```
switch# show event-monitor mac
% Writing 0 Arp, 0 Route, 1 Mac events to the database
2012-01-19 13:57:55|1|0808.0808.0808|Ethernet1|configuredStaticMac|added|0
```

- This command displays events triggered by MAC address table changes.

```
switch# show event-monitor mac match-mac 08:08:08:%
2012-01-19 13:57:55|1|0808.0808.0808|Ethernet1|configuredStaticMac|added|0
```

4.7.39 show event-monitor mroute

The **show event-monitor mroute** command performs an SQL-style query on the event-monitor database, using the statement specified in the command.

Command Mode

Privileged EXEC

Command Syntax

show event-monitor mroute [**GROUP**] [**MESSAGES**] [**IP**] [**INTERFACE**] [**SRC_IP**] [**TIME**]

Optional parameters can be placed in any order.

Parameters

- **GROUP** used with aggregate functions to group results. Analogous to SQL **group by** command.
  - ***no parameter*** results are not grouped.
  - **group-by interface** results are grouped by interface.
  - **group-by ipv6** results are grouped by IPv6 address.
  - **group-by mac** results are grouped by MAC address.
  - **group-by vrf** results are grouped by VRF.
- **MESSAGES** number of messages returned from query. Analogous to SQL **limit** command.
  - ***no parameter*** result-set size is not limited.
  - **limit *msg_quantity*** number of results that are displayed. Values range from ***1*** to ***15,000***.
- **IP** restricts result-set to events that include specified IP address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific IP addresses.
  - **match-ipv6 *ip_address_rex*** IP address, as represented by regular expression.
- **INTERFACE** restricts result-set to events that include specified interface (SQL Like command).
  - ***no parameter*** result-set not restricted by interface.
  - **match-interface ethernet *e_range*** Ethernet interface list.
  - **match-interface loopback *l_range*** loopback interface list.
  - **match-interface management *m_range*** management interface list.
  - **match-interface port-channel *c_range*** port channel interface list.
  - **match-interface vlan *v_range*** VLAN interface list.
  - **match-interface tunnel *t_range*** tunnel interface list.
  - **match-interface vxlan *vx_range*** VXLAN interface list.
- **SRC_IP** restricts result-set to events that include specified Source IP address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific IP addresses.

- **match-ip *ip_address_rex*** IP address, as represented by regular expression.

- **TIME** restricts result-set to events with specified period.
  - ***no parameter*** result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.

Example

This command displays neighbor table events listed in the event monitor database.

```
switch# show event-monitor mroute
2011-07-28 12:33:28|default|16.17.18.19/32|225.0.0.1/32|||added|30
2011-07-28 12:33:28|default|16.17.18.19/32|225.0.0.1/32|Vlan2|iif|join|31
2011-07-28 12:33:28|default|16.17.18.19/32|225.0.0.1/32|Vlan3|oif|join|32
2011-07-28 12:33:28|default|16.17.18.19/32|225.0.0.1/32|Vlan4|oif|join|33
2011-07-28 12:33:28|default|10.11.12.13/32|225.0.0.2/32|||added|34
2011-07-28 12:33:28|default|10.11.12.13/32|225.0.0.2/32|Vlan3|iif|join|35
2011-07-28 12:33:28|default|10.11.12.13/32|225.0.0.2/32|Vlan2|oif|join|36
2011-07-28 12:33:28|default|16.17.18.19/32|225.0.0.1/32|Vlan4||leave|37
2011-07-28 12:33:28|default|16.17.18.19/32|225.0.0.1/32|||deleted|38
2011-07-28 12:33:28|default|10.11.12.13/32|225.0.0.2/32|||deleted|39
```

4.7.40 show event-monitor neighbor

The **`show event-monitor neighbor`** command performs an SQL-style query on the event monitor database and displays neighbor table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

`show event-monitor neighbor` [**GROUP**][**MESSAGES**][**INTERFACE**][**IP6**][**MAC**][**TIME**][**VRF**]

Optional parameters can be placed in any order.

Parameters

- **GROUP** used with aggregate functions to group results. Analogous to SQL **group by** command.
  - ***no parameter*** results are not grouped.
  - **group-by interface** results are grouped by interface.
  - **group-by ip6** results are grouped by IPv6 address.
  - **group-by mac** results are grouped by MAC address.
  - **group-by vrf** results are grouped by VRF.
- **MESSAGES** number of messages returned from query. Analogous to SQL **limit** command.
  - ***no parameter*** result-set size is not limited.
  - **limit *msg_quantity*** number of results that are displayed. Values range from ***1*** to ***15,000***.
- **INTERFACE** restricts result-set to events that include specified interface (SQL Like command).
  - ***no parameter*** result-set not restricted by interface.
  - **match-interface ethernet *e_range*** Ethernet interface list.
  - **match-interface loopback *l_range*** loopback interface list.
  - **match-interface management *m_range*** management interface list.
  - **match-interface port-channel *c_range*** port channel interface list.

  - **match-interface vlan *v_range*** VLAN interface list.
  - **match-interface tunnel *t_range*** tunnel interface list.
  - **match-interface vxlan *vx_range*** VXLAN interface list.
- **IP6** restricts result-set to events that include specified IP address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific IP addresses.
  - **match-ipv6 *ip6_address_rex*** IPv6 address, as represented by regular expression.
- **MAC** restricts result-set to events that include specified MAC address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific MAC addresses.
  - **match-mac *mac_address_rex*** MAC address, as represented by regular expression.
- **TIME** restricts result-set to events with specified period.
  - ***no parameter*** result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.
- **VRF** restricts result-set to events that include a specific VRF (SQL Like command).
  - ***no parameter*** result-set not restricted by time of event.
  - **match-vrf *vrf_name*** VRF name, as represented by a regular expression.

Example

This command displays neighbor table events listed in the event monitor database.

```
switch# show event-monitor neighbor
2019-09-30 14:37:32.894147|def0::1|Vlan1|default|0005.0005.0005|1|added|1
2019-09-30 14:37:32.894395|def0::2|Vlan1|default|0005.0005.0005|1|added|2
2019-09-30 14:37:32.894607|def0::3|Vlan1|default|0005.0005.0005|1|added|3
2019-09-30 14:37:32.894815|def0::4|Vlan1|default|0005.0005.0005|1|added|4
2019-09-30 14:37:32.895071|def0::5|Vlan1|default|0005.0005.0005|1|added|5
2019-09-30 14:37:32.895303|def0::6|Vlan1|default|0005.0005.0005|1|added|6
2019-09-30 14:37:32.895527|def0::7|Vlan1|default|0005.0005.0005|1|added|7
2019-09-30 14:37:32.895732|def0::8|Vlan1|default|0005.0005.0005|1|added|8
2019-09-30 14:37:32.895968|def0::9|Vlan1|default|0005.0005.0005|1|added|9
2019-09-30 14:37:32.896194|def0::a|Vlan1|default|0005.0005.0005|1|added|10
```

4.7.41 show event-monitor route6

The **`show event-monitor route6`** command performs an SQL-style query on the event monitor database and displays routing6 table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

`show event-monitor route6` [**GROUP**][**MESSAGES**][**IP6**][**TIME**]

Optional parameters can be placed in any order.

Parameters

- **GROUP** used with aggregate functions to group results. Analogous to SQL **group by** command.
  - ***no parameter*** results are not grouped.
  - **group by interface** results are grouped by interface.
  - **group by ip6** results are grouped by IPv6 address.

    - **group by mac** results are grouped by MAC address.
    - **group by vrf** results are grouped by VRF.
- **MESSAGES** number of messages returned from query. Analogous to SQL **limit** command.
    - ***no parameter*** result-set size is not limited.
    - **limit *msg_quantity*** number of results that are displayed. Values range from ***1*** to ***15,000***.
- **IP6** restricts result-set to events that include specified IP address (SQL Like command).
    - ***no parameter*** result-set not restricted to specific IP addresses.
    - **match-ipv6 *ip6_address_rex*** IPv6 address, as represented by regular expression.
- TIME restricts result-set to events with specified period.
    - ***no parameter*** result-set not restricted by time of event.
    - **match-time last-minute** includes events generated during last minute.
    - **match-time last-day** includes events generated during last day.
    - **match-time last-hour** includes events generated during last hour.
    - **match-time last-week** includes events generated during last week.

Example

This command displays neighbor table events listed in the event monitor database.

```
switch# show event-monitor route6
2019-09-30 14:59:30.660447|def1::1:0/128|default|receive|0|1|updated|41
2019-09-30 14:59:30.660720|def1::2:0/128|default|attached|0|1|updated|42
2019-09-30 14:59:30.660983|def1::3:0/128|default|staticConfig|0|1|updated|43
2019-09-30 14:59:30.661226|def1::4:0/128|default|kernel|0|1|updated|44
2019-09-30 14:59:30.661469|def1::5:0/128|default|rip|0|1|updated|45
2019-09-30 14:59:30.661706|def1::6:0/128|default|connected|0|1|updated|46
2019-09-30 14:59:30.661968|def1::7:0/128|default|redirect|0|1|updated|47
2019-09-30 14:59:30.662207|def1::8:0/128|default|bgpAggregate|0|1|updated|48
2019-09-30 14:59:30.662451|def1::9:0/128|default|ospfAggregate|0|1|updated|49
2019-09-30 14:59:30.662694|def1::a:0/128|default|ospf|0|1|updated|50
2019-09-30 14:59:30.662935|def1::b:0/128|default|bgp|0|1|updated|51
2019-09-30 14:59:30.663174|def1::c:0/128|default|unknown|0|1|updated|52 switch#
```

4.7.42 show event-monitor route

The **`show event-monitor route`** command performs an SQL-style query on the event monitor database and displays routing table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

`show event-monitor route` [**GROUP**][**MESSAGES**][**IP**][**TIME**]

Optional parameters can be placed in any order.

Parameters

- **GROUP** used with aggregate functions to group results. Analogous to SQL **group by** command.
    - ***no parameter*** results are not grouped.
    - **group-by ip** results are grouped by IPv4 address.
- **MESSAGES** number of messages returned from query. Analogous to SQL **limit** command.
    - ***no parameter*** result-set size is not limited.
    - **limit *msg_quantity*** number of results that are displayed. Values range from ***1*** to ***15,000***.

- **IP** restricts result-set to events that include specified IP address (SQL Like command).
  - ***no parameter*** result-set not restricted to specific IP addresses.
  - **match-ip *ip_address_rex*** IP address, as represented by regular expression.
- **TIME** restricts result-set to events with specified period.
  - ***no parameter*** result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.

Example

This command displays 10 routing table events listed in the event monitor database.

```
switch# show event-monitor route limit 10
2019-09-30 14:01:21.659428|16.16.16.255/32|default|receiveBcast|0|0|updated|20
2019-09-30 14:01:21.659464|192.168.201.12/30|default|connected|1|0|updated|21
2019-09-30 14:01:21.659497|192.168.1.255/32|default|receiveBcast|0|0|updated|22
2019-09-30 14:01:21.659503|192.168.201.8/32|default|receiveBcast|0|0|updated|23
2019-09-30 14:01:21.659512|16.16.16.0/32|default|receiveBcast|0|0|updated|24
2019-09-30
14:01:21.659517|192.168.201.12/32|default|receiveBcast|0|0|updated|25
2019-09-30
14:01:21.659524|192.168.201.15/32|default|receiveBcast|0|0|updated|26
2019-09-30 14:01:21.659541|192.168.201.8/30|default|connected|1|0|updated|27
2019-09-30 14:01:21.659564|16.16.16.0/24|default|connected|1|0|updated|28
2019-09-30 14:01:21.659578|192.168.201.9/32|default|receive|0|0|updated|29
 switch#
```

4.7.43 show event-monitor sqlite

The **`show event-monitor sqlite`** command performs an SQL-style query on the event monitor database, using the statement specified in the command.

Command Mode

Privileged EXEC

Command Syntax

`show event-monitor sqlite` ***statement***

Parameter

statement SQLite statement.

Example

This command displays all entries from the route table.

```
switch# show event-monitor sqlite select * from route;
2019-09-30 14:01:21.659428|16.16.16.255/32|default|receiveBcast|0|0|updated|20
2019-09-30 14:01:21.659464|192.168.201.12/30|default|connected|1|0|updated|21
2019-09-30 14:01:21.659497|192.168.1.255/32|default|receiveBcast|0|0|updated|22
2019-09-30 14:01:21.659503|192.168.201.8/32|default|receiveBcast|0|0|updated|23
2019-09-30 14:01:21.659512|16.16.16.0/32|default|receiveBcast|0|0|updated|24
2019-09-30
14:01:21.659517|192.168.201.12/32|default|receiveBcast|0|0|updated|25
2019-09-30
14:01:21.659524|192.168.201.15/32|default|receiveBcast|0|0|updated|26
2019-09-30 14:01:21.659541|192.168.201.8/30|default|connected|1|0|updated|27
2019-09-30 14:01:21.659564|16.16.16.0/24|default|connected|1|0|updated|28
```

```
2019-09-30 14:01:21.659578|192.168.201.9/32|default|receive|0|0|updated|29
 switch#
```

4.7.44 show event-monitor stpunstable

The **show event-monitor stpunstable** command performs an SQL-style query on the event-monitor database, using the statement specified in the command.

Command Mode

Privileged EXEC

Command Syntax

show event-monitor stpunstable [**MESSAGES**][**TIME**]

Optional parameters can be placed in any order.

Parameters

- **MESSAGES** number of messages returned from query. Analogous to SQL **limit** command.
  - ***no parameter*** result-set size is not limited.
  - **limit *msg_quantity*** number of results that are displayed. Values range from ***1*** to ***15,000***.
- **TIME** restricts result-set to events with specified period.
  - ***no parameter*** result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.

Example

```
switch# show event-monitor stpunstable limit 5
2019-02-07 07:22:10.286164|Cist|Ethernet5|forward-delay-while|1
2019-02-07 07:22:10.286651|Cist|Ethernet6|forward-delay-while|2
2019-02-07 07:22:10.286844|Cist|Ethernet8|forward-delay-while|3
2019-02-07 07:22:10.287030|Cist|Ethernet14|forward-delay-while|4
2019-02-07 07:22:10.287215|Cist|Ethernet21|forward-delay-while|5 switch#
```

4.7.45 show hostname

The **show hostname** command displays the hostname and the Fully Qualified Domain Name (FQDN) of the switch.

Command Mode

EXEC

Command Syntax

show hostname

Example

This command displays the hostname and FQDN of the switch.

```
switch> show hostname
Hostname: switch_1
FQDN:     switch_1.aristanetworks.com switch>
```

4.7.46 show hosts

The **show hosts** command displays the default domain name, name lookup service style, a list of name server hosts, and the static hostname-IP address maps.

Command Mode

EXEC

Command Syntax

show hosts

Example

This command displays the switchs IP domain name:

```
switch> show hosts
Default domain is: aristanetworks.com
Name/address lookup uses domain service
Name servers are: 172.22.22.40, 172.22.22.10
Static Mappings:

Hostname                         IP     Addresses

TEST_LAB                        IPV4    10.24.18.6
PRODUCTION_LAB                  IPV4    10.24.18.7
SUPPORT_LAB                     IPV6    2001:0DB8:73:ff:ff:26:fd:90 switch>
```

4.7.47 show ip domain-name

The **show ip domain-name** command displays the switchs IP domain name that is configured with the ip domain name command.

Command Mode

EXEC

Command Syntax

show ip domain-name

Example

This command displays the switchs IP domain name:

```
switch> show ip domain-name
aristanetworks.com switch>
```

4.7.48 show ip name-server

The **show ip name-server** command displays the ip addresses of name-servers in ***running-config***. The name servers are configured by the **ip name-server** command.

Command Mode

EXEC

Command Syntax

show ip name-server

Example

This command displays the IP address of name servers that the switch is configured to access.

```
switch>show ip name-server
172.22.22.10
172.22.22.40 switch>
```

4.7.49 show local-clock time-properties

The **show local-clock time-properties** command displays the Precision Time Protocol (PTP) clock properties.

Command Mode

Privileged EXEC

Command Syntax

show local-clock time-properties

Example

This command shows the PTP clock properties.

```
switch# show local-clock time-properties
Current UTC offset valid: False
Current UTC offset: 0
Leap 59: False
Leap 61: False
Time Traceable: False
Frequency Traceable: False
PTP Timescale: False
Time Source: 0x0 switch#
```

4.7.50 show ntp associations

The **show ntp associations** command displays the status of connections to NTP servers.

Command Mode

EXEC

Command Syntax

show ntp associations

Display Values

- **refid (reference ID)**: the reference ID of the configured NTP server's time source. The reference ID is either the IPv4 address of the source or (if the source has an IPv6 address) the first four octets of the MD5 hash of the IPv6 address.
- **st (stratum)**: number of steps between the switch and the reference clock.
- **t (transmission type)**: u unicast; b broadcast; l local.
- **when**: interval since reception of last packet (seconds unless unit is provided).
- **poll**: interval between NTP poll packets. Maximum (1024) reached as server and client syncs.
- **reach**: octal number that displays status of last eight NTP messages (377 - all messages received).
- **delay**: round-trip delay of packets to the NTP server.
- **offset**: difference between local clock and the server's clock.
- **jitter**: nominal offset estimation error.

Example

This command displays the status of the switch's NTP associations.

```
switch>show ntp associations
     remote           refid      st t when  poll reach   delay   offset  jitter
==============================================================================
+1.ntp.arista.co 125.157.10.11   2 u  539  1024  377  121.748   -0.345   0.893
-3.ntp.arista.co 127.31.152.34   2 u  868  1024  377  101.671    2.434   1.529
+2.ntp.arista.co 176.131.12.185  2 u  676  1024  377  116.505     0.03   0.768
*4.ntp.arista.co 120.181.192.192 2 u  696  1024  377   48.431   -0.416    0.15 switch#
```

4.7.51 show ntp status

The **show ntp status** command displays the status of NTP on the switch. If the switch clock is not synchronized to an NTP server, the status reads "unsynchronised" and shows the server polling interval. If the clock is synchronized to an NTP server, the status shows the IP address and stratum of the server, the precision of the synchronization, and the polling interval.

Note:

In EOS releases prior to *4.23.2*, this command identified system peers with IPv6 addresses by their reference IDs (the first four octets of the MD5 hash of the IPv6 address). In later releases, this command always shows the IP address of the system peer (whether IPv4 or IPv6).

Command Mode

EXEC

Command Syntax

show ntp status

Example

This command displays the switch's NTP status.

```
switch> show ntp status
synchronised to NTP server (192.168.78.62) at stratum 3
   time correct to within 66 ms
   polling server every 1024 s switch>
```

Chapter 5

7130 Layer 1 Configuration

The source for patching Layer 1 traffic between front panel ports, application FPGA ports, and Switch interfaces is configured using the **`l1 source`** command. Switch interfaces can be used instead of the Application interfaces for making connections between SwitchApp interfaces and front panel ports.

For more information refer to https://www.arista.com/en/support/toi/eos-4-27-1f/14877-switchapp-ultra-low-latency-packet-switch

7130 Layer 1 Commands

- l1 source
- show l1 source
- show l1 source capabilities
- show l1 destination
- show l1 matrix
- show l1 path

5.1 l1 source

The **`l1 source`** command configures the source for patching Layer 1 traffic between various front panel and Switch Application (SwitchApp) ports in the Ethernet Interface configuration mode.

The **`no l1 source`** and **`default l1 source`** commands remove the source configurations and return to the default platform configurations.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`l1 source` {interface | mac | none}

Parameters

- **interface** patches the Layer 1 traffic from the specified interface.
  - **application** Application FPGA connector interface
  - **cpu** CPU interface
  - **ethernet** Ethernet interface
  - **switch** Switch interface
- **mac** patches the Layer 1 traffic from the specified MAC address.
- **none** unpatches the Layer 1 traffic source.

Examples

- This command patches the Layer 1 traffic from the Ethernet interface ***et2***.

```
switch(config)# interface ethernet1
switch(config-if-Et1)# l1 source interface et2
switch(config-if-Et1)#
```

- This command unpatches the Layer 1 traffic source.

```
switch(config)# interface ethernet1
switch(config-if-Et1)# l1 source none
```

5.2 show l1 source

The **show l1 source** command displays the list of all source interfaces to connect for specific interfaces such as Ethernet ports, switch, FPGA ports, or CPU ports.

Command Mode

Privileged EXEC

Command Syntax

show l1 source [interface | capabilities]

Parameters

- **interface** lists the source interfaces available to connect for the specified interface.
  - **application** Application FPGA connector interface
  - **cpu** CPU interface
  - **ethernet** Ethernet interface
  - **switch** Switch interface
- **capabilities** lists the Layer 1 source capabilities for the specified switch.

Examples

- This command displays the list of all source interfaces to connect in the crosspoint and the specific interfaces.

```
switch# show l1 source
   Interface             Source Interface    Type
-------------------- ---------------------- -------
   Ethernet1             App1/1              dynamic
   Ethernet2
   Ethernet3             Ethernet2           dynamic
   Ethernet4             Ethernet2           dynamic
   Ethernet5             MAC                 dynamic
   App1/1                Ethernet1           dynamic
   App1/2                Sw17                static
   Sw1                   Ethernet1           dynamic
   Sw2
   Sw3
   Sw4
   Sw5
   Sw17                  App1/2              static
   Cpu1
   Cpu2
```

Note: Interfaces that do not have a source port are listed too but the Source Interface and Type columns appear as blanks.

- This command displays the list of source interfaces to connect for the Ethernet interface ***et1***.

```
switch# show l1 source interface et1
Interface             Source Interface    Type
--------------------- ------------------- -------
Ethernet1             App1/1              dynamic
```

5.3 show l1 source capabilities

The **`show l1 source capabilities`** command displays the list of all interfaces available to connect in the crosspoint and their possible source ports.

Command Mode

Privileged EXEC

Command Syntax

`show l1 source capabilities`

Example

- This command displays the list of all possible source interfaces available to connect in the crosspoint and the specific interfaces.

```
switch# show l1 source capabilities
Interface             Possible Source Interface(s)
--------------------- --------------------------------
Ethernet1             Ap1/1,Cpu1-2,Et1-16,Sw1-16
Ethernet2             Ap1/1,Cpu1-2,Et1-16,Sw1-16
Ethernet3             Ap1/1,Cpu1-2,Et1-16,Sw1-16
Ethernet4             Ap1/1,Cpu1-2,Et1-16,Sw1-16
Ethernet5             Ap1/1,Cpu1-2,Et1-16,Sw1-16
Application1/1        Ap1/1,Cpu1-2,Et1-16,Sw1-16
Application1/2        Sw17
Switch1               Ap1/1,Cpu1-2,Et1-16,Sw1-16
Switch2               Ap1/1,Cpu1-2,Et1-16,Sw1-16
Switch3               Ap1/1,Cpu1-2,Et1-16,Sw1-16
Switch4               Ap1/1,Cpu1-2,Et1-16,Sw1-16
Switch5               Ap1/1,Cpu1-2,Et1-16,Sw1-16
Switch17              Ap1/2
Cpu1                  Ap1/1,Cpu1-2,Et1-16,Sw1-16
Cpu2                  Ap1/1,Cpu1-2,Et1-16,Sw1-16
```

- This command displays the list of possible source interfaces available to connect for the Ethernet interface ***et1***.

```
switch# show l1 source interface et1 capabilities
Interface             Possible Source Interface(s)
--------------------- --------------------------------
Ethernet1             Ap1/1,Cpu1-2,Et1-16,Sw1-16
```

- This command displays the list of possible source interfaces available to connect for the FPGA port ***fpga***.

```
switch# show l1 source fpga capabilities
Interface             Possible Source Interface(s)
--------------------- ------------------------------
Application1/1        Ap1/1,Cpu1-2,Et1-16,Sw1-16
Application1/2        Sw17
```

5.4 show l1 destination

The **`show l1 destination`** command displays the list of all destinations for all interfaces.

Command Mode

Privileged EXEC

Command Syntax

`show l1 destination` [interface]

Parameters

interface lists the destination interfaces to connect for the specified interface.

- **application** Application FPGA connector interface.
- **cpu** CPU interface.
- **ethernet** Ethernet interface.
- **switch** Switch interface.

Examples

- This command displays the the list of all interfaces to connect in the crosspoint and their specific destination ports.

```
switch> show l1 destination
   Interface             Destination Interface(s) Type
   ------------------    ------------------------ ------
   Ethernet1             App1/1, Sw1              dynamic
   Ethernet2             Ethernet3, Ethernet4     dynamic
   Ethernet3
   Ethernet4
   Ethernet5
   App1/1                Ethernet1                dynamic
   App1/2                Sw17                     static
   Sw1
   Sw2
   Sw3
   Sw4
   Sw5
   Sw17                  App1/2                   static
   Cpu1
   Cpu2
```

Note: Interfaces that do not have a destination port are listed too but the Destination Interface and Type columns appear as blanks.

- This command displays the list of destination interfaces to connect for the Ethernet interface ***et1***.

```
switch# show l1 destination interface et1

Interface             Destination Interfaces   Type
--------------------- ------------------------ ------
Ethernet1             App1/1, Sw1              dynamic
```

5.5 show l1 matrix

The **`show l1 matrix`** command displays the matrix representation of Layer 1 paths for available interfaces in the privileged EXEC mode.

Command Mode

Privileged EXEC

Command Syntax

`show l1 matrix` [all | interface | detail | physical]

Parameters

- **all** displays the matrix of Layer 1 paths for possible interface types even if they do not have a connection.
- **interface** displays the matrix of Layer 1 paths for the specified interface.
  - **application** Application FPGA connector interface.
  - **cpu** CPU interface.
  - **ethernet** Ethernet interface.
  - **switch** Switch interface.
- **detail** displays the detailed matrix of Layer 1 paths.
- **physical** displays the Layer 1 matrix for physical interfaces only.

Examples

- This command displays the matrix of Layer 1 paths for all available interface types.

```
switch> show l1 matrix
              Ethernet   Switch     App
              1 3 4      1 17       1/1 1/2
              | | |      | |        |   |
      Et1 -> -|-|-|------+-|--------+   |
      Et2 -> -|-+-+        |            |
     Sw17 -> -|------------|------------+
   App1/1 -> -+            |
   App1/2 -> --------------+
```

- This command displays the matrix of Layer 1 paths for all available interface types even if the interfaces do not have a connection.

```
switch> show l1 matrix all
              Ethernet                                      Switch
              1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16   1 2 3 4 5 6 7 8 9 10

              |   | |                                   |
      Et1 -> -|---|-|-----------------------------------+
      Et2 -> -|---+-+
      Et3 ->  |
      Et4 ->  |
      Et5 ->  |
      Et6 ->  |
      Et7 ->  |
      Et8 ->  |
      Et9 ->  |
     Et10 ->  |
     Et11 ->  |
     Et12 ->  |
     Et13 ->  |
     Et14 ->  |
     Et15 ->  |
     Et16 ->  |
      Sw1 ->  |
      Sw2 ->  |
      Sw3 ->  |
      Sw4 ->  |
      Sw5 ->  |
      Sw6 ->  |
      Sw7 ->  |
      Sw8 ->  |
      Sw9 ->  |
     Sw10 ->  |
     Sw11 ->  |
     Sw12 ->  |
     Sw13 ->  |
     Sw14 ->  |
```

```
     Sw15 ->  |
     Sw16 ->  |
     Sw17 ->  |
   App1/1 -> -+
   App1/2 ->
     Cpu1 ->
     Cpu2 ->

              Switch                  App       Cpu
              11 12 13 14 15 16 17    1/1 1/2   1 2
                                      |   |     |
      Et1 -> --------------------|----+   |
      Et2 ->                     |        |
      Et3 ->                     |        |
      Et4 ->                     |        |
      Et5 ->                     |        |
      Et6 ->                     |        |
      Et7 ->                     |        |
      Et8 ->                     |        |
      Et9 ->                     |        |
     Et10 ->                     |        |
     Et11 ->                     |        |
     Et12 ->                     |        |
     Et13 ->                     |        |
     Et14 ->                     |        |
     Et15 ->                     |        |
     Et16 ->                     |        |
      Sw1 ->                     |        |
      Sw2 ->                     |        |
      Sw3 ->                     |        |
      Sw4 ->                     |        |
      Sw5 ->                     |        |
      Sw6 ->                     |        |
      Sw7 ->                     |        |
      Sw8 ->                     |        |
      Sw9 ->                     |        |
     Sw10 ->                     |        |
     Sw11 ->                     |        |
     Sw12 ->                     |        |
     Sw13 ->                     |        |
     Sw14 ->                     |        |
     Sw15 ->                     |        |
     Sw16 ->                     |        |
     Sw17 -> --------------------|--------+
   App1/1 ->                     |
   App1/2 -> --------------------+
     Cpu1 ->
     Cpu2 ->
```

- This command displays the matrix of Layer 1 paths for the specific Ethernet interfaces ***et1*** and ***et4***.

```
switch> show l1 matrix interface et1, et4
           Ethernet   Switch   App
           1 4        1        1/1
           | |        |        |
   Et1 -> -|-|--------+--------+
   Et2 -> -|-+
App1/1 -> -+
```

5.6 show l1 path

The **`show l1 path`** command displays the source and destination paths for all interfaces in the system.

Command Mode

Privileged EXEC

Command Syntax

`show l1 path` [interface]

Parameters

- **interface** lists the connections available for the specified interface.
  - **application** Application FPGA connector interface.
  - **cpu** CPU interface.
  - **ethernet** Ethernet interface.
  - **switch** Switch interface.

Examples

- This command displays the source and destination paths for all interfaces in the system.

```
switch# show l1 path
   Source               Destination          Type
   -------------------  -------------------- -------
   Ethernet1             App1/1              dynamic
   Ethernet1             Sw1                 dynamic
   Ethernet2             Ethernet3           dynamic
   Ethernet2             Ethernet4           dynamic
   App1/1                Ethernet1           dynamic
   App1/2                Sw17                static
   Sw17                  App1/2              static
```

- This command displays the list of all connections that include the Ethernet interfaces ***et1*** and ***et4***.

```
switch# show l1 path interface et1, et4
   Source               Destination          Type
--------------------- -------------------- -------
   Ethernet1             App1/1              dynamic
   Ethernet1             Sw1                 dynamic
   Ethernet2             Ethernet4           dynamic
   App1/1                Ethernet1           dynamic
```

Chapter 6

Timing Protocols

The Precision Time Protocol (PTP) provides a greater degree of clock accuracy for networked devices, allowing clocks to be synchronized locally in increments of less than a microsecond. PTP uses a master-slave hierarchy similar to that used by NTP. The most precise clock available is referred to as the master clock, and slave devices use the signal from the master to synchronize their own clocks.

The master clock sends out a **sync** message (referred to as an **announce** message in IEEE 1588-2008) at a regular interval. The slave clock responds with a time-stamped delay request message in order to measure and compensate for packet delays between the devices. The slave then receives a message from the master specifying when the delay message was received, which allows the slave to calculate final values for clock synchronization. Synchronization is maintained by the regular exchange of PTP packets between master and slave.



Note: Arista switches do not support setting of the system clock using PTP. System clock synchronization is best supported by the NTP service on the PTP grandmaster.

PTP is disabled globally by default. The following steps are required to enable PTP on an interface:

- Setting the PTP Mode
- Enabling PTP on an Interface

The following PTP global configurations are optional:

- Configuring the PTP Domain
- Configuring the Offset Hold Time
- Setting the PTP Priority
- Configuring the Source IP
- Configuring the TTL for PTP Packets
- Configuring PTP Monitoring

The following PTP interface-level configurations are optional:

- Setting the PTP Announce Interval
- Setting the PTP Timeout Interval
- Configuring the PTP Delay Mechanism
- Setting the Delay Request Interval
- Setting the Peer Delay Request Interval
- Setting the Peer Link Propagation Threshold
- Setting the Interval for Sending Synchronization Messages
- Setting the PTP Transport Type
- Viewing PTP Settings and Status
- Precision Time Protocol (PTP) Commands

6.1 Setting the PTP Mode

To allow PTP to be used on switch interfaces, first set the PTP mode using the **`ptp mode`** command. PTP mode options include:

- **boundary** The device acts as a boundary clock, and both runs and participates in the best master clock algorithm.
- **disabled** PTP is disabled, and the device forwards all PTP packets as normal traffic.

- **end-to-end transparent** The device acts as an end-to-end transparent clock, synchronizing all ports to a connected master clock and updating the time interval field of forwarded PTP packets using switch residence time.
- **peer-to-peer transparent** The device acts as a peer-to-peer transparent clock, synchronizing all ports to a connected master clock and updating the time interval field of forwarded PTP packets using switch residence time and inbound path delays.
- **generalized Precision Time Protocol (gPTP)** The device runs generalized Precision Time Protocol (gPTP), participating in the best master clock algorithm but also updating the interval field of forwarded PTP packets using switch residence time and inbound path delays.

To disable PTP globally on the switch, use the **no** or **default** forms of the **ptp mode** command.

Example

This command configures the device as a PTP boundary clock.

```
switch(config)# ptp mode boundary
switch(config)#
```

6.2 Enabling PTP on an Interface

To enable PTP on a specific interface on the device, use the **ptp enable** command.

Example

This command enables PTP on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)#ptp enable
```

6.3 Configuring the PTP Domain

To set the domain number to use for the clock, use the **ptp domain** command.

Example

This command configures the ***domain 1*** to use with a clock.

```
switch(config)# ptp domain 1
switch(config)#
```

6.4 Configuring the Offset Hold Time

To set the PTP offset hold time, use the **ptp hold-ptp-time** command.

Example

This command configures the PTP offset hold time to ***600*** seconds.

```
switch(config)# ptp hold-ptp-time 600
switch(config)#
```

6.5 Setting the PTP Priority

To set the priority 1 value, use the **ptp priority1** command. Lower values take precedence.

This command configures the priority 1 value of ***120*** to use when advertising the clock.

```
switch(config)# ptp priority1 120
switch(config)#
```

To set the priority 2 value for the clock, use the **ptp priority2** command.

This command configures the priority 2 value of ***128***.

```
switch(config)# ptp priority2 128
switch(config)#
```

6.6 Configuring the Source IP

To set the source IP address for all PTP packets, use the **ptp source ip** command.

Example

This command configures the source IP address of ***10.0.2.1*** for all PTP packets.

```
switch(config)# ptp source ip 10.0.2.1
switch(config)#
```

6.7 Configuring the TTL for PTP Packets

To set the Time To Live (TTL) of PTP packets, use the **ptp ttl** command. TTL is the maximum number of hops that a PTP packet may make.

Example

This command configures a TTL of ***64*** hops for PTP packets.

```
switch(config)# ptp ttl 64
switch(config)#
```

6.8 Configuring PTP Monitoring

PTP monitoring records PTP information including offset from master, mean path delay, and skew values, which can then be viewed using a **show** command. When this feature is enabled, PTP Syslog messages will also be generated for those metrics for which threshold values have been configured on the switch. PTP monitoring is enabled by default.

Enabling and Disabling PTP Monitoring

Use the **ptp monitor** command to enable PTP monitoring on the device (it is enabled by default). The no form of the command disables PTP monitoring and clears all the recorded PTP data.

Example

This command disables PTP monitoring and clears all recorded PTP data from the switch.

```
switch(config)# no ptp monitor
```

Configuring the Offset-from-master Threshold

The offset is the difference in nanoseconds between master and slave time. Use the **ptp monitor threshold offset-from-master** command to specify the offset-from-master threshold in nanoseconds. A Syslog message is generated if the most recently calculated time offset from the PTP master is outside of the range (-<threshold>, <threshold>). The maximum offset threshold is one second. The no form of the command clears the threshold value and prevents further Syslog messages from being generated for this parameter.

Example

This command sets an offset-from-master threshold value of ***500*** nanoseconds.

```
switch(config)# ptp monitor threshold offset-from-master 500
```

Configuring the Mean-path-delay Threshold

Mean path delay is the mean time in nanoseconds that PTP packets take to travel between master and slave. Use the **ptp monitor threshold mean-path-delay** command to specify the mean-path-delay threshold in nanoseconds. A Syslog message is generated if the value of the most recently calculated mean path delay is greater than or equal to this threshold. The maximum mean-path-delay threshold is one second. The no form of the command clears the threshold value and prevents further Syslog messages from being generated for this parameter.



Note: Mean path delay is always non-negative.

Example

This command sets a mean-path-delay threshold value of ***2000*** nanoseconds.

```
switch(config)# ptp monitor threshold mean-path-delay 2000
```

Configuring the Skew Threshold

PTP skew is the clock frequency difference between master and slave. Use the **ptp monitor threshold skew** command to configure the value of the skew-threshold percentage. A Syslog message is generated if the value of the most recently calculated skew is not in the range (1/(1+<threshold>), 1*(1+<threshold>)). Skew threshold percentage is represented a double precision (16 digit) real number ranging from ***0*** (0%) to ***10*** (1000%). The no form of the command clears the threshold value and prevents further Syslog messages from being generated for this parameter.

Example

This command sets a skew threshold value of ***5*** (500%).

```
switch(config)# ptp monitor threshold skew 5
```

6.9 Setting the PTP Announce Interval

To set the interval at which an interface sends PTP **announce** messages, use the **ptp announce interval** command. The interval is measured in log seconds. This value also affects the timeout interval.

This command configures the interval between PTP announcement messages on ***interface ethernet 5*** to ***4*** seconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp announce interval 2
switch(config-if-Et5)#
```

6.10 Setting the PTP Timeout Interval

To set the timeout multiplier for an interface, use the **ptp announce timeout** command. The timeout multiplier is the number of announcement intervals that the interface will wait without receiving a PTP announcement before a timeout occurs; values range from ***2*** to ***255***. The default multiplier is ***3***, which results in a ***6***-second timeout interval when the announcement interval is set to the default of ***2*** seconds.

This command sets timeout multiplier for the interface to ***5***; since the announcement interval has just been set to ***2*** (***4*** seconds), this means the interface will time out if it doesnt receive a PTP announcement for ***20*** seconds.

```
switch(config-if-Et5)# ptp announce timeout 5
switch(config-if-Et5)#
```

6.11 Configuring the PTP Delay Mechanism

To set the delay mechanism used in boundary-mode, use the **ptp delay-mechanism** command.

Example

This command sets the delay mechanism in boundary clock mode for the interface to peer-to-peer.

```
switch(config-if-Et5)# ptp delay-mechanism p2p
switch(config-if-Et5)#
```

6.12 Setting the Delay Request Interval

To set the time for the slave devices to send delay request messages, use the **ptp delay-req interval** command.

Example

This command sets the time the slave devices to send delay request messages to the master state to ***3*** for the interface.

```
switch(config-if-Et5)# ptp delay-request interval 3
switch(config-if-Et5)#
```

6.13 Setting the Peer Delay Request Interval

To set the minimum interval between the PTP peer delay-request messages, use the **ptp pdelay-req interval** command.

Example

This command sets the interval between PTP peer delay-request messages on the interface to ***3***.

```
switch(config-if-Et5)# ptp pdelay-request interval 3
switch(config-if-Et5)#
```

6.14 Setting the Peer Link Propagation Threshold

To set the delay threshold for which the peer will be considered unable to run generalized Precision Time Protocol (gPTP), use the **ptp pdelay-neighbor-threshold** command.

Example

This command sets the link propagation delay threshold on the interface to ***200000*** nanoseconds..

```
switch(config-if-Et5)# ptp pdelay-neighbor-threshold 200000
switch(config-if-Et5)#
```

6.15 Setting the Interval for Sending Synchronization Messages

To set the interval (in log seconds) for sending synchronization messages, use the **ptp sync-message interval** command. Value ranges and defaults vary based on the PTP mode of the switch.

Example

This command configures the interval for sending synchronization messages on the interface to ***3*** (***8*** seconds).

```
switch(config-if-Et5)# ptp sync-message interval 3
switch(config-if-Et5)#
```

6.16 Setting the PTP Transport Type

To set the PTP transport type, use the **ptp transport** command.

- This command configures the PTP transport type for the interface to IPv4.

```
switch(config-if-Et5)# ptp transport ipv4
switch(config-if-Et5)#
```

- This command configures the PTP transport type for the interface to IPv6.

```
switch(config-if)# ptp transport ipv6
switch(config-if)#
```

Setting the Local Priority of the Clock and Interfaces

To set the local priority of the clock and interfaces to control the topology, use the **ptp local-priority** command.

```
switch(config)# ptp local-priority 1
switch(config-if)# ptp local-priority 255
```

Setting up as a Slave to another PTP Device

Each interface may be configured with a candidate grantor IP address to send requests and potentially become slave to another PTP device. Once configured, the switch starts negotiating with the IP and depending on the Announce messages it receives, it may start requesting Sync and Delay Response to sync its clock. Each grantor may be associated with a unicast negotiation profile. If the profile is omitted, the default interval of one second and duration of 60 seconds for all message types is used. If the profile does not exist, the switch uses the default values until the profile gets added.

Use the **`ptp unicast-negotiation candidate-grantor`** command to set up the profile. The following commands set up the candidate grantor profile.

```
switch(config-if)# ptp unicast-negotiation candidate-grantor 10.0.0.1
switch(config-if)# ptp unicast-negotiation candidate-grantor 10.0.0.1 profile
 fastProfile
```

Setting up as a Master to another PTP Device

Each interface may be configured with a range of IP addresses of remote grantees to grant incoming requests and potentially become a master to another PTP device. By default, incoming requests outside the configured range of IP addresses will be denied. Each grantee may be associated with a unicast negotiation profile. If the profile is omitted, incoming requests with interval of ***0*** or longer for all message types are granted. If a profile is specified, it will compare with the configured interval. If the profile does not exist, the switch uses the default values until the profile gets added.

Use the **`ptp unicast-negotiation remote-grantee`** command to set up the profile. The following commands set up the remote grantee profile.

```
switch(config-if)# ptp unicast-negotiation remote-grantee 10.0.0.1/24
switch(config-if)# ptp unicast-negotiation remote-grantee 10.0.0.1/24 profile
 fastProfile
```

Setting the masterOnly Flag

For ports that should never be a slave regardless of other attributes, use the **`ptp role master`** command.

```
switch(config-if)# ptp role master
```

Configuring the Unicast Profile

A Unicast negotiation profile may be configured to change message rates and durations. The default value and the configurable range of each value are shown in the table below.

Field	Range	Default
Announce interval	[-3, 0]	0
Announce duration	[60, 1000]	60
Sync interval	[-7, 0]	0
Sync duration	[60, 1000]	60
Delay Response interval	[-7, 0]	0
Delay Response duration	[60, 1000]	60

When a profile is applied to a remote grantee on a Grant port, it will use the values to determine whether the given request should be granted or denied. If the requested interval is shorter than the profile, it will be

denied; otherwise, it will be granted. When a profile is applied to a candidate grantor on a Request port, it will be requested to the candidate grantor using the values in the profile.

The following configures a typical Unicast negotiation profile.

```
switch(config)# ptp unicast-negotiation profile fastProfile
switch(config-unicast-negotiation-profile-fastProfile)# announce interval -2
switch(config-unicast-negotiation-profile-fastProfile)# announce duration 500
switch(config-unicast-negotiation-profile-fastProfile)# sync interval -3
switch(config-unicast-negotiation-profile-fastProfile)# sync duration 300
switch(config-unicast-negotiation-profile-fastProfile)# delay-resp interval -3
switch(config-unicast-negotiation-profile-fastProfile)# delay-resp duration 300
```

Displaying Unicast Negotiation Configurations

The **show ptp unicast-negotiation profile** command displays all user configured profiles and their values.

```
(switch)# show ptp unicast-negotiation profile
Unicast Negotiation Profile fastProfile
Announce interval: 0.25 seconds
Announce duration: 500 seconds
Sync interval: 0.125 seconds
Sync duration: 300 seconds
Delay Response interval: 0.125 seconds
Delay Response duration: 300 seconds
```

The **show ptp unicast-negotiation candidate-grantor** command displays all configured candidate grantors, associated profile name and latest update.

```
(switch)# show ptp unicast-negotiation candidate-grantor
 Interface    Address      Profile       Grantor Status
---------- ---------- -------------- ------------------
 Ethernet1    4::1       fastProfile    Master
 Ethernet1    4::2       fastProfile    Candidate Master
 Ethernet2    4::2       fastProfile    Blacklisted
```

The **show ptp unicast-negotiation remote-grantee** command displays all configured configured remote grantees, associated profile name and latest update.

```
(switch)# show ptp unicast-negotiation remote-grantee
Interface    Address      Profile
------------ ---------- -------------
 Ethernet1    4::1/96    fastProfile
 Ethernet1    4::2/96    fastProfile
 Ethernet2    4::2/96    fastProfile
```

The **show ptp unicast-negotiation granted** command displays to which remote grantees each port has granted and some detail.

```
(switch)# show ptp unicast-negotiation granted
Interface    Address    Message     Interval        Duration       Expires In
---------- ---------- ------------ -------------- -------------- -------------
 Ethernet2    4::1       Announce    0.25 seconds    300 seconds    30 seconds
 Ethernet2    4::1       Sync        2.0 seconds     300 seconds    30 seconds
```

The **show ptp unicast-negotiation requested** command displays to which candidate grantors each port has requested and some detail.

```
(switch)# show ptp unicast-negotiation requested
 Interface    Address    Message     Interval        Duration       Expires In
------------ --------- ----------- --------------- -------------- -------------
```

```
 Ethernet2    4::2       Announce    0.25 seconds    600 seconds    250 seconds
 Ethernet2    4::2       Sync        2.0 seconds     300 seconds    denied
```

The **show ptp local-clock** command displays the clock local priority in G8275.2 mode.

```
(switch)# show ptp local-clock
 PTP Mode: Boundary Clock
Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Clock Domain: 44
Number of PTP ports: 64
Priority1: 128
Priority2: 128
Local Priority: 128
Clock Quality:
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
Offset From Master: -5
Mean Path Delay: 416 nanoseconds
Steps Removed: 1
Skew: 1.00000006399
Last Sync Time: 23:42:33 UTC Nov 01 2018
Current PTP System Time: 23:42:33 UTC Nov 01 2018
```

The **show ptp interface** command displays the interface local priority in G8275.2 mode.

Note: The Announce, Sync and Delay request message intervals are for multicast mode. Since G8275.2 operates in unicast mode, those values should be ignored.

```
(switch)# show ptp interface Ethernet 42 | nz  Interface Ethernet42
PTP: Enabled
Port state: Slave
Sync interval: 1.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 3
Delay mechanism: end to end
Delay request message interval: 0.25 seconds
Local Priority: 128
Transport mode: ipv4
Announce messages received: 2964
Sync messages received: 2558
Follow up messages received: 2558
Delay request messages sent: 2540
Delay response messages received: 2540
Signaling messages sent: 98
Signaling messages received: 101
```

6.17 Viewing PTP Settings and Status

The following commands display the status of the switch PTP server connections:

- Displaying General PTP Information
- Displaying PTP Local Clock and Offset
- Displaying PTP Masters Information
- Displaying PTP Clock Properties
- Displaying PTP Information for all Interfaces
- Displaying PTP Interface Counters
- Displaying PTP Foreign Master
- Displaying PTP Monitoring Information

- Displaying PTP Source IP

6.17.1 Displaying General PTP Information

To display general Precision Time Protocol (PTP) information, use the **show ptp** command.

```
switch# show ptp
PTP Mode: gptp - Generalized PTP Clock
Clock Identity: 2001:0DB8:73:ff:ff:26:fd:90
Grandmaster Clock Identity: 2001:0DB8:96:ff:fe:6c:ed:02
Number of slave ports: 1
Number of master ports: 6
Slave port: Ethernet33
Mean Path Delay (nanoseconds): 718
Steps Removed: 1
Neighbor Rate Ratio: 1.00000007883
Rate Ratio: 1.00000007883
Interface  State    ASTime  Since LastNeighbor Mean        Path        Residence
Capable    Changed  Rate    Ratio              Delay (ns)  Time (ms)
---------  -------- ------- ------------------ ----------- ----------- ---------
Et1        Disabled No      Never              1.0         00
Et2        Disabled No      Never              1.0         00
Et3        Disabled No      Never              1.0         00
Et4        Disabled No      Never              1.0         00
Et5        Disabled No      Never              1.0         00
Et6        Disabled No      Never              1.0         00
Et7        Master   Yes                        0:21:08     1.000000094200
```

6.17.2 Displaying PTP Clock Properties

To display PTP clock properties, use the Displaying PTP Clock Properties command.

```
switch# show local-clock time-properties
Current UTC offset valid: False
Current UTC offset: 0
Leap 59: False
Leap 61: False
Time Traceable: False
Frequency Traceable: False
PTP Timescale: False
Time Source: 0x0 switch#
```

6.17.3 Displaying PTP Foreign Master

To display information about foreign masters (PTP sources not designated as the switchs master from which the switch has received sync packets), use the **show ptp foreign-master-record** command.

```
switch# show ptp foreign-master-record
No Foreign Master Records switch#
```

6.17.4 Displaying PTP Information for all Interfaces

To display PTP information for specified interfaces, use the **show ptp interface** command.

```
switch# show ptp interface
Interface Ethernet1
PTP: Disabled
Port state: Disabled
Sync interval: 1.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 3
Delay mechanism: end to end
Delay request message interval: 32.0 seconds
```

```
Transport mode: ipv4

Interface Ethernet5
PTP: Disabled
Port state: Disabled
Sync interval: 8.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 5
Delay mechanism: peer to peer
Peer delay request message interval: 8.0 seconds
Peer Mean Path Delay: 0
Transport mode: ipv4 switch#
```

6.17.5 Displaying PTP Interface Counters

To display PTP interface counters for specified interfaces, use the **show ptp interface counters** command.

```
switch# show ptp interface ethernet 5 counters
Interface Ethernet5
Announce messages sent: 0
Announce messages received: 0
Sync messages sent: 0
Sync messages received: 0
Follow up messages sent: 0
Follow up messages received: 0
Delay request messages sent: 0
Delay request messages received: 0
Delay response messages sent: 0
Delay response messages received: 0
Peer delay request messages sent: 0
Peer delay request messages received: 0
Peer delay response messages sent: 0
Peer delay response messages received: 0
Peer delay response follow up messages sent: 0
Peer delay response follow up messages received: 0
switch#
```

6.17.6 Displaying PTP Local Clock and Offset

To display the local PTP clock and offset, use the **show ptp local-clock** command.

```
switch# show ptp local-clock
PTP Mode: Boundary Clock
Clock Identity: 0x00:1c:73:ff:ff:1e:83:24
Clock Domain: 1
Number of PTP ports: 24
Priority1: 128
Priority2: 128
Clock Quality:
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
Offset From Master: 0
Mean Path Delay: 0
Steps Removed: 0 switch#
```

6.17.7 Displaying PTP Masters Information

To display the PTP clocks master and grandmaster identity and configuration, use the **show ptp masters** command.

```
switch# show ptp masters
Parent Clock:
Parent Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Parent Port Number: 0
Parent IP Address: N/A
Observed Parent Offset (log variance): N/A
Observed Parent Clock Phase Change Rate: N/A

Grandmaster Clock:
Grandmaster Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Grandmaster Clock Quality:
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
   Priority1: 128
   Priority2: 128 switch#
```

6.17.8 Displaying PTP Monitoring Information

To display the list of up to 100 recorded entries of offset from master, mean path delay and skew values, the current PTP mode, whether or not the feature is enabled, the number of entries displayed, and the configured thresholds for each metric, use the **show ptp monitor** command. Entries are sorted by the system time at which the value was calculated, starting with the most recent data at the top.

```
switch# show ptp monitor
PTP Mode: Boundary Clock
Ptp monitoring: enabled
Number of entries: 5
Offset from master threshold: 1500
Mean path delay threshold: not configured
Skew threshold: 0.5
Interface  Time                           Offset from   Mean Path    Skew
                                          Master (ns)   Delay (ns)
---------- ------------------------------ ------------ -------------
 -----------
Et8        21:23:12.901 UTC Feb 22 2018  71            5849
 1.003159918
Et1        21:23:12.901 UTC Feb 22 2018  113           3672
 1.004990621
Et2        21:23:12.901 UTC Feb 22 2018  706           7799
 1.002744199
Et1        21:23:12.901 UTC Feb 22 2018  803           5861
 1.003432049
Et1        21:23:12.901 UTC Feb 22 2018  610           3415         0.998974658
```

6.17.9 Displaying PTP Source IP

To display PTP IP source information, use the **show ptp source ip** command.

```
switch# show ptp source ip
PTP source IP: 10.0.2.1 switch#
```

6.18 Precision Time Protocol (PTP) Commands

PTP Commands

- ptp announce interval
- ptp announce timeout
- ptp delay-mechanism
- ptp delay-req interval
- ptp domain
- ptp enable
- ptp forward-v1
- ptp hold-ptp-time
- ptp local-priority
- ptp mode
- ptp monitor
- ptp monitor threshold mean-path-delay
- ptp monitor threshold offset-from-master
- ptp monitor threshold skew
- ptp pdelay-neighbor-threshold
- ptp pdelay-req interval
- ptp priority1
- ptp priority2
- ptp role
- ptp source ip
- ptp sync timeout
- ptp sync-message interval
- ptp transport
- ptp ttl
- ptp unicast-negotiation

PTP Show Commands

- show ptp foreign-master-record
- show ptp interface
- show ptp interface counters
- show ptp local-clock
- show ptp masters
- show ptp monitor
- show ptp source ip
- show ptp unicast-negotiation
- show ptp unicast-negotiation profile
- show ptp

6.18.1 ptp announce interval

The **`ptp announce interval`** command configures the interval at which the configuration mode interface sends PTP **announce** messages. The **`no ptp announce interval`** command resets the announce interval to its default of 1 (2 seconds).

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

`ptp announce interval` ***log_interval***

`no ptp announce interval`

`default ptp announce interval`

Parameters

log_interval The number of log seconds between PTP **announce** messages (base ***2*** log (seconds)). Value ranges from ***-3*** (***1/8*** second) to ***4*** (***16*** seconds); default value is ***1*** (***2*** seconds).

Examples

- These commands set the interval between PTP **announce** messages sent by ***interface ethernet 5*** to ***4*** seconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp announce interval 2
switch(config-if-Et5)#
```

- These commands reset the PTP **announce** interval on ***interface ethernet 5*** to the default value of ***1*** (***2*** seconds).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp announce interval
switch(config-if-Et5)#
```

6.18.2 ptp announce timeout

The **`ptp announce timeout`** sets the timeout multiplier for the configuration-mode interface. The timeout multiplier is the number of announcement intervals that the interface will wait without receiving a PTP **announce** message before a timeout occurs; the range is from ***2*** to ***255***. The default multiplier is 3, which results in a 6-second timeout interval when the announce interval is set to the default of 2 seconds. To configure the announce interval, use the **`ptp announce interval`** command.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

`ptp announce timeout` ***multiplier***

`no ptp announce timeout`

`default ptp announce timeout`

Parameters

multiplier Number of announce intervals after which the interface will time out if it does not receive a PTP **announce** message. The range is from ***2*** to ***255***; default value is ***3***.

Examples

- This command sets the timeout multiplier for ***interface ethernet 5*** to ***5***. This means that the interface will time out if it doesnt receive a PTP **announce** message within five announce intervals.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp announce timeout 5
switch(config-if-Et5)#
```

- These commands reset the PTP timeout interval on ***interface ethernet 5*** to the default value of ***3***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp announce timeout
switch(config-if-Et5)#
```

6.18.3 ptp delay-mechanism

The **ptp delay-mechanism** command configures the delay mechanism in boundary clock mode. The **no ptp delay-mechanism** command disables the feature.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

ptp delay-mechanism ***mech_type***

no ptp delay-mechanism

default ptp delay-mechanism

Parameters

mech_type The delay mechanism. Options include:

- **e2e** end-to-end delay mechanism.
- **p2p** peer-to-peer mechanism.

Examples

- This command sets the delay mechanism to peer-to-peer in the boundary clock mode.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp delay-mechanism p2p
switch(config-if-Et5)#
```

- This command sets the delay mechanism to end-to-end in the boundary clock mode.
- This command removes the delay mechanism configuration from ***Ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp delay-mechanism e2e
switch(config-if-Et5)#
```

6.18.4 ptp delay-req interval

The **ptp delay-req interval** command specifies the time in log seconds recommended to the slave devices to send delay request messages. You must enable PTP on the switch first and configure the source IP address for PTP communication. The **no ptp delay-req interval** command resets the interval to its default of ***5*** (***32*** seconds).

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

ptp delay-req interval ***log_interval***

no ptp delay-req interval

default ptp delay-req interval

Parameters

log_interval The range is ***-1*** to ***8*** log seconds (base ***2*** log (seconds)). The default is ***5*** (***32*** seconds).

Examples

- These commands set the minimum interval allowed between PTP delay request messages on ***interface ethernet 5*** to ***3*** (***8*** seconds).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp delay-request interval 3
switch(config-if-Et5)#
```

- These commands reset the minimum interval allowed between PTP delay-request messages to the default of ***5*** (***32*** seconds).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp delay-request interval
switch(config-if-Et5)#
```

6.18.5 ptp domain

The **ptp domain** command sets the domain number to use for the clock. The **no ptp domain** command resets or restores the domain to the default number 0.

Command Mode

Global Configuration

Command Syntax

ptp domain ***domain_number***

no ptp domain

default ptp domain

Parameters

domain_number Value ranges from ***0*** to ***255***. Default number is 0.

Examples

- This command shows how to configure ***domain 1*** for use with a clock.

```
switch(config)# ptp domain 1
switch(config)#
```

- This command removes the configured ***domain 1*** for use with a clock.

```
switch(config)# no ptp domain 1
switch(config)#
```

6.18.6 ptp enable

The **ptp enable** command enables PTP on the interface. The **no ptp enable** command disables PTP on the interface.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

ptp enable

no ptp enable

default ptp enable

Examples

- This command enables PTP on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp enable
```

- This command disables PTP on ***Ethernet interface 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp enable
```

6.18.7 ptp forward-v1

The **ptp forward-v1** command configures the switch to forward Precision Time Protocol version 1 packets as regular multicast traffic. By default, PTP v1 packets are trapped by the CPU, logged and discarded.

The **no ptp forward-v1** and **default ptp forward-v1** commands restore the default forwarding behavior by removing the corresponding **ptp forward-v1** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ptp forward-v1

no ptp forward-v1

default ptp forward-v1

Examples

- This command configures the switch to forward PTP v1 packets as regular multicast traffic.

```
switch(config)# ptp forward-v1
switch(config)#
```

- This command configures the switch to log and discard PTP v1 packets.

```
switch(config)# no ptp forward-v1
switch(config)#
```

6.18.8 ptp hold-ptp-time

The **ptp hold-ptp-time** command configures the PTP offset hold time in seconds. The **no ptp hold-ptp-time** command resets or restores the PTP hold time to the default value.

Command Mode

Global Configuration

Command Syntax

ptp hold-ptp-time ***offset***

no ptp hold-ptp-time

default ptp hold-ptp-time

Parameters

offset Value ranges from ***0*** to ***86400***.

Examples

- This command shows how to configure the PTP offset hold time.

```
switch(config)# ptp hold-ptp-time 600
switch(config)#
```

- This command resets or restores the PTP offset hold time to the default value.

```
switch(config)# no ptp hold-ptp-time
switch(config)#
```

6.18.9 ptp local-priority

The **ptp local-priority** command configures the local priority of the clock and interfaces to control the topology.

Command Mode

Interface-Ethernet Configuration

Command Syntax

ptp local-priority **PRIORITY_NUM**

Parameters

PRIORITY_NUM The priority number from ***1*** to ***255***.

Example

This command sets the priority for the subinterface.

```
switch(config)# ptp local-priority 1
switch(config-if)# ptp local-priority 255
```

6.18.10 ptp mode

The **ptp mode** command configures the Precision Time Protocol (PTP) packet forwarding mode for the switch. By default, PTP is disabled globally; the mode must be changed to use PTP on switch interfaces.

The **no ptp mode** and **default ptp mode** commands return the forwarding mode to **disabled** by removing the **ptp mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ptp mode ***mode_name***

no ptp mode

default ptp mode

Parameters

mode_name Default mode is **disabled**. Options include:

- **boundary** The device acts as a boundary clock, and both runs and participates in the best master clock algorithm.
  - **disabled** The default mode. PTP is disabled, and the device forwards all PTP packets as normal traffic.
  - **e2etransparent** The device acts as an end-to-end transparent clock, synchronizing all ports to a connected master clock and updating the time interval field of forwarded PTP packets using switch residence time.
  - **p2ptransparent** The device acts as a peer-to-peer transparent clock, synchronizing all ports to a connected master clock and updating the time interval field of forwarded PTP packets using switch residence time and inbound path delays.
  - **gptp** The device runs generalized Precision Time Protocol (gPTP), participating in the best master clock algorithm but also updating the interval field of forwarded PTP packets using switch residence time and inbound path delays.

Examples

- This command configures the switch to act as a PTP boundary clock.

```
switch(config)# ptp mode boundary
switch(config)#
```

- This command restores PTP to disabled mode.

```
switch(config)# no ptp mode
switch(config)#
```

6.18.11 ptp monitor threshold mean-path-delay

Mean path delay is the mean time in nanoseconds that PTP packets take to travel between PTP master and slave. The **ptp monitor threshold mean-path-delay** command configures the mean-path-delay threshold in nanoseconds. When this threshold is configured, a Syslog message is generated if the value of the most recently calculated mean path delay is greater than or equal to this value.

The **no ptp monitor threshold mean-path-delay** and **default ptp monitor threshold mean-path-delay** commands clear the threshold value and prevent further Syslog messages from being generated for this parameter.

Command Mode

Global Configuration

Command Syntax

ptp monitor threshold mean-path-delay ***threshold***

no ptp monitor threshold mean-path-delay

default ptp monitor threshold mean-path-delay

Parameter

threshold threshold in nanoseconds. Values range from ***0*** to ***1000000000*** (***1*** second).

Example

This command sets a mean-path-delay threshold value of ***2000*** nanoseconds.

```
switch(config)# ptp monitor threshold mean-path-delay 2000
```

6.18.12 ptp monitor threshold offset-from-master

PTP offset is the difference in nanoseconds between master and slave time. The **ptp monitor threshold offset-from-master** command configures the offset-from-master threshold in nanoseconds. A Syslog message is generated if the most recently calculated time offset from the PTP master is outside of the range (-<threshold>, <threshold>). The maximum offset threshold is one second.

The **no ptp monitor threshold offset-from-master** and **no ptp monitor threshold offset-from-master** commands clear the threshold value and prevents further Syslog messages from being generated for this parameter.

Command Mode

Global Configuration

Command Syntax

ptp monitor threshold offset-from-master ***threshold***

no ptp monitor threshold offset-from-master

default ptp monitor threshold offset-from-master

Parameter

threshold Offset threshold value in nanoseconds. Values range from ***0*** to ***1000000000***.

Example

This command sets an offset-from-master threshold value of ***500*** nanoseconds.

```
switch(config)# ptp monitor threshold offset-from-master 500
switch(config)#
```

6.18.13 ptp monitor threshold skew

PTP skew is the clock frequency difference between master and slave. The **ptp monitor threshold skew** command configures the value of the skew-threshold percentage. A Syslog message is generated if the value of the most recently calculated skew is not in the range (1/(1+*threshold*), 1*(1+*threshold*)).

The **no ptp monitor threshold skew** and **default ptp monitor threshold skew** commands clear the threshold value and prevent further Syslog messages from being generated for this parameter.

Command Mode

Global Configuration

Command Syntax

ptp monitor threshold skew ***threshold***

no ptp monitor threshold skew

default ptp monitor threshold skew

Parameters

threshold skew percentage threshold represented as a double precision (16 digit) real number ranging from ***0*** (***0%***) to ***10*** (***1000%***).

Example

This command sets a skew threshold value of ***5*** (***500%***).

```
switch(config)# ptp monitor threshold skew 5
switch(config)#
```

6.18.14 ptp monitor

The **ptp monitor** command enables and disables PTP monitoring on the switch. When PTP monitoring is enabled, the switch records PTP status and configuration information (which can be viewed using the **show ptp monitor** command) and generates Syslog messages for metrics whose threshold values have been configured. PTP monitoring is enabled by default.

The **no ptp monitor** command disables the PTP monitoring and clears all the recorded data from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ptp monitor

no ptp monitor

Related Commands

- **ptp monitor threshold mean-path-delay**
- **ptp monitor threshold offset-from-master**
- **ptp monitor threshold skew**
- **show ptp monitor**

Example

This command disables ptp monitor PTP monitoring on the switch and clears all the recorded data.

```
switch(config)# ptp monitor
switch(config)#
```

6.18.15 ptp pdelay-neighbor-threshold

The **ptp pdelay-neighbor-threshold** command configures the propagation delay threshold above which the switch will consider the neighbor connected to this port to be incapable of participating in generalized Precision Time Protocol (gPTP).

The **no ptp pdelay-neighbor-threshold** and **default ptp pdelay-neighbor-threshold** commands restore the threshold to 100000 nanoseconds by removing the corresponding **ptp pdelay-neighbor-threshold** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

ptp pdelay-neighbor-threshold ***link_prop***

no ptp pdelay-neighbor-threshold

default ptp pdelay-neighbor-threshold

Parameters

link_prop Threshold in nanoseconds. Value ranges from ***0*** to ***10000000000*** (ten billion). Default is ***100000***.

Examples

- These commands set the link propagation delay threshold on ***interface ethernet 5*** to ***200000*** nanoseconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp pdelay-neighbor-threshold 200000
switch(config-if-Et5)#
```

- These commands restore the link propagation delay threshold on ***interface ethernet 5*** to its default value of ***100000*** nanoseconds.

```
switch(config)# interface ethernet 5s
witch(config-if-Et5)# no ptp pdelay-neighbor-threshold
switch(config-if-Et5)#
```

6.18.16 ptp pdelay-req interval

The **ptp pdelay-req interval** command configures the interval between Precision Time Protocol peer delay-request messages. The **no ptp pdelay-req interval** command removes the configuration.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

ptp pdelay-req interval ***log_interval***

no ptp pdelay-req interval

default ptp pdelay-req interval

Parameters

log_interval The log interval in seconds (base 2 log (seconds)). Value ranges from ***0*** to ***5***.

Examples

- This command shows how to configure the interval allowed between PTP peer delay request messages on ***interface Ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp pdelay-request interval 3
switch(config-if-Et5)#
```

- This command removes the configure the interval allowed between PTP peer delay request messages on ***interface Ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp pdelay-request interval
switch(config-if-Et5)#
```

6.18.17 ptp priority1

The **ptp priority1** command configures the priority 1 value for advertising the switchs PTP clock. Priority 1 is the most significant of the six factors used by devices in the selection of a master clock. Lower values indicate higher priority.

The **no ptp priority1** and **default ptp priority1** commands restore the priority 1 default setting of 128.

Command Mode

Global Configuration

Command Syntax

ptp priority1 ***priority_rate***

no ptp priority1

default ptp priority1

Parameters

priority_rate Value ranges from ***0*** to ***255***. Default is ***128***.

Examples

- This command sets the ***priority 1*** level for the switchs PTP clock to ***120***.

```
switch(config)# ptp priority1 120
switch(config)#
```

- This command restores the default ***priority 1*** level of ***128***.

```
switch(config)# no ptp priority1
switch(config)#
```

6.18.18 ptp priority2

The **ptp priority2** command sets the ***priority 2*** value for the clock. The range is from ***0*** to ***255***. ***Priority 2*** is the fifth most significant of the six factors used by devices in the selection of a master clock. Lower values indicate higher priority.

The **no ptp priority2** and **default ptp priority2** commands restore the ***priority 2*** default setting of ***128***.

Command Mode

Global Configuration

Command Syntax

ptp priority2 ***priority_rate***

no ptp priority2

default ptp priority2

Parameters

priority_rate Specifies the ***priority 2*** level for the PTP clock. Value ranges from ***0*** to ***255***; default value is ***128***.

Examples

- This command sets the ***priority 2*** level for the switchs PTP clock to ***120***.

```
switch(config)# ptp priority2 120
switch(config)#
```

- This command restores the default ***priority 2*** level of ***128***.

```
switch(config)# no ptp priority2
switch(config)#
```

6.18.19 ptp role

The **ptp role** command configures a port to operate either in the master mode or the dynamic mode when it is executed in the interface configuration mode.

The **no ptp role** command removes the master or dynamic mode if it was previously configured on an interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

ptp role [dynamic | master]

no ptp role

default ptp role

Related Commands

- **ptp enable**
- **ptp enable**
- **show ptp interface**

Parameters

- **dynamic** the dynamic mode.
- **master** the master clock mode that has the most precise time.

Examples

- This command configures a port to operate in the master mode for ***interface ethernet 1***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# ptp role master
```

- This command configures a port to operate in the dynamic mode for ***interface ethernet 1***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# ptp role dynamic
```

6.18.20 ptp source ip

The **ptp source ip** command configures the source IP address for all PTP packets. The IP address can be in IPv4 format. The **no ptp source ip** command removes this configuration.

Command Mode

Global Configuration

Command Syntax

ptp source ip ***ipv4_addr***

no ptp source ip

default ptp source ip

Parameters

ipv4_addr the IPv4 address.

Examples

- This command configures the source IP address ***10.0.2.1*** for all PTP packets.

```
switch(config)# ptp source ip 10.0.2.1
switch(config)#
```

- This command removes the source IP address **10.0.2.1** for all PTP packets.

```
switch(config)# no ptp source ip
switch(config)#
```

6.18.21 ptp sync timeout

A PTP synchronization timeout occurs if a sync message is not received for a specified period of time, calculated as a multiple of the PTP sync-message interval. The **`ptp sync timeout`** command configures the sync timeout multiplier. The range is ***2*** to ***255***, with a default of ***20*** (20 times the sync interval). To configure the sync interval, use the **`ptp sync-message interval`** command.

The **`no ptp sync timeout`** and **`default ptp sync timeout`** commands restore the PTP sync timeout multiplier to its default value of ***20***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`ptp sync timeout` ***interval_multiplier***

`no ptp sync timeout`

`default ptp sync timeout`

Parameters

interval_multiplier The number of sync intervals that must pass without the configuration mode interface receiving a PTP sync message before a timeout occurs. Value ranges from ***2*** to ***255***. Default value is ***20***.

Example

These commands configure the sync timeout on ***interface ethernet 5*** to ten times the configured sync interval.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp sync timeout 10
switch(config-if-Et5)#
```

6.18.22 ptp sync-message interval

The **`ptp sync-message interval`** command configures the time for sending synchronization messages by specifying its log2 value. Default values and ranges depend on the PTP mode, which is set using the **`ptp mode`** command.

The **`no ptp sync-message interval`** and **`default ptp sync-message interval`** commands restore the sync interval to its default (***1*** second in **boundary** mode, ***1/8*** second in **gptp** mode) by removing the corresponding **`ptp sync-message interval`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

`ptp sync-message interval` ***log_interval***

no ptp sync-message interval

default ptp sync-message interval

Parameters

log_interval The interval between PTP synchronization messages sent from the master to the slave (base 2 log (seconds)). Values vary according to PTP mode: in **boundary** mode, the range is from ***-7*** (***1/128*** second) to ***3*** (***8*** seconds) and the default value is ***0*** (***1*** second). In **gptp** mode, the range is from ***-3*** (***1/8*** second) to ***17*** (***131072*** seconds, approximately ***36*** hours) with a default of ***-3***.

Examples

- These commands set the interval for PTP synchronization messages on ***interface ethernet 5*** to ***3*** (***8*** seconds).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp sync-message interval 3
switch(config-if-Et5)#
```

- In **boundary** mode, these commands restore the interval for PTP synchronization messages on ***interface ethernet 5*** to its default of ***0*** (***1*** second).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp sync-message interval
switch(config-if-Et5)#
```

6.18.23 ptp transport

The **ptp transport** command configures the PTP transport type for a specific interface. Any values set in interface PTP configuration mode override the settings in the PTP configuration profile associated with the interface. The **no ptp transport** command removes the stetting from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

ptp transport **TRANSPORT_TYPE**

no ptp transport

default ptp transport

Parameters

TRANSPORT_TYPE The transport mode in boundary clock mode. Options include:

- **ipv4** The IPv4 address used as the transport type on the interface.
- **ipv6** The IPv6 address used as the transport type on the interface.
- **layer2** The Layer 2 protocol used as the transport type on the interface.

Examples

- This command overrides the transport type in the profile and sets it to be IPv4 for the interface.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp transport ipv4
switch(config-if-Et5)#
```

- This command removes the interval for PTP synchronization messages on ***interface Ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp transport
switch(config-if-Et5)#
```

6.18.24 ptp ttl

The **ptp ttl** command configures the Time To Live (TTL) value of the PTP packets. The **no ptp ttl** resets the TTL value to the default value of ***1*** hop by removing the **ptp ttl** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

ptp ttl ***hop_count***

no ptp ttl

default ptp ttl

Parameters

hop_count The TTL value measured in hops. Value ranges from ***1*** to ***255***, default is ***1***.

Examples

- This command sets the time to live of the PTP packets to ***60*** hops.

```
switch(config)# ptp ttl 60
switch(config)#
```

- This command resets the time to live of the PTP packets to the default value of ***1*** hop.

```
switch(config)# no ptp ttl
switch(config)#
```

6.18.25 ptp unicast-negotiation

The **ptp unicast-negotiation** command configures the master slave profiles for PTP devices.

Command Mode

Interface-Ethernet Configuration

Command Syntax

ptp unicast-negotiation

ptp unicast-negotiation remote-grantee

Parameters

- **candidate-grantor** become slave to another PTP device.
- **remote-grantee** become master to another PTP device.

Examples

- This command sets up an interface as a potential master.

```
switch(config-if)# ptp unicast-negotiation remote-grantee 10.0.0.1/24
```

```
switch(config-if)# ptp unicast-negotiation remote-grantee 10.0.0.1/24
 profile fastProfile
```

- This command sets up an interface as a potential slave.

```
switch(config-if)# ptp unicast-negotiation candidate-grantor 10.0.0.1
switch(config-if)# ptp unicast-negotiation candidate-grantor 10.0.0.1
 profile fastProfile
```

6.18.26 show ptp

The **show ptp** command displays summary Precision Time Protocol (PTP) information and PTP status of switch ports.

Command Mode

EXEC

Command Syntax

show ptp

Example

This command displays summary PTP information.

```
switch#show ptp
PTP Mode: gptp - Generalized PTP Clock
Clock Identity: 2001:0DB8:73:ff:ff:26:fd:90
Grandmaster Clock Identity: 2001:0DB8:96:ff:fe:6c:ed:02
Number of slave ports: 1
Number of master ports: 6
Slave port: Ethernet33
Mean Path Delay (nanoseconds): 718
Steps Removed: 1
Neighbor Rate Ratio: 1.00000007883
Rate Ratio: 1.00000007883
Interface StateASTime Since LastNeighborMean PathResidence
Capable   Changed  Rate    Ratio              Delay (ns)  Time (ms)
--------- -------- ------- ------------------ ----------- -----------
Et1       Disabled No      Never              1.0          00
Et2       Disabled No      Never              1.0          00
Et3       Disabled No      Never              1.0          00
Et4       Disabled No      Never              1.0          00
Et5       Disabled No      Never              1.0          00
Et6       Disabled No      Never              1.0          00
Et7       Master           Yes                0:21:08      1.000000094200
```

6.18.27 show ptp foreign-master-record

The **show ptp foreign-master-record** command displays information about foreign masters (PTP sources not designated as the switchs master from which the switch has received sync packets).

Command Mode

EXEC

Command Syntax

show ptp foreign-master-record

Example

This command displays information about PTP foreign masters.

```
switch# show ptp foreign-master-record
No Foreign Master Records switch#
```

6.18.28 show ptp interface counters

The **show ptp interface counters** command displays PTP interface counters for all interfaces.

Command Mode

EXEC

Command Syntax

show ptp[**INTERFACE_NAME**] **counters**

Parameters

INTERFACE_NAME Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **loopback *l_range*** Loopback interface specified by ***l_range***
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
- **vlan *v_range*** VLAN interface range specified by ***v_range***.
- **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.

Valid range formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the PTP interface counters.

```
switch# show ptp interface ethernet 5 counters
Interface Ethernet5
Announce messages sent: 0
Announce messages received: 0
Sync messages sent: 0
Sync messages received: 0
Follow up messages sent: 0
Follow up messages received: 0
Delay request messages sent: 0
Delay request messages received: 0
Delay response messages sent: 0
Delay response messages received: 0
Peer delay request messages sent: 0
Peer delay request messages received: 0
Peer delay response messages sent: 0
Peer delay response messages received: 0
Peer delay response follow up messages sent: 0
Peer delay response follow up messages received: 0
switch#
```

6.18.29 show ptp interface

The **show ptp interface** command displays PTP information for all the interfaces on the device.

Command Mode

EXEC

Command Syntax

show ptp [**INTERFACE_NAME**][**STATUS_FILTER**]

Parameters

- **INTERFACE_NAME** Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **loopback *l_range*** Loopback interface specified by ***l_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
- **vlan *v_range*** VLAN interface range specified by ***v_range***.

Valid range formats include number, number range, or comma-delimited list of numbers and ranges.

- **STATUS_FILTER** Filters interfaces by their configuration status. Options include:
  - ***no parameter*** all interfaces.
  - **enabled** PTP configured interfaces.

Example

This command displays PTP information for all the interfaces on the device.

```
switch# show ptp interface
Interface Ethernet1
PTP: Disabled
Port state: Disabled
Sync interval: 1.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 3
Delay mechanism: end to end
Delay request message interval: 32.0 seconds
Transport mode: ipv4

Interface Ethernet5
PTP: Disabled
Port state: Disabled
Sync interval: 8.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 5
Delay mechanism: peer to peer
Peer delay request message interval: 8.0 seconds
Peer Mean Path Delay: 0
Transport mode: ipv4 switch#
```

6.18.30 show ptp local-clock

The **`show ptp local-clock`** command displays the Precision Time Protocol (PTP) clock information.

Command Mode

EXEC

Command Syntax

`show ptp local-clock`

Example

This command shows how to display the PTP local clock and offset.

```
switch# show ptp local-clock
PTP Mode: Boundary Clock
Clock Identity: 0x00:1c:73:ff:ff:1e:83:24
Clock Domain: 1
Number of PTP ports: 24
Priority1: 128
Priority2: 128
Clock Quality:
```

```
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
Offset From Master: 0
Mean Path Delay: 0
Steps Removed: 0 switch#
```

6.18.31 show ptp masters

The **show ptp masters** command displays information about the switchs PTP master and grand master clocks.

Command Mode

Privileged EXEC

Command Syntax

show ptp masters

Example

This command displays information about the switchs PTP master and grand master clocks.

```
switch# show ptp masters
Parent Clock:
Parent Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Parent Port Number: 0
Parent IP Address: N/A
Observed Parent Offset (log variance): N/A
Observed Parent Clock Phase Change Rate: N/A

Grandmaster Clock:
Grandmaster Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Grandmaster Clock Quality:
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
   Priority1: 128
   Priority2: 128 switch#
```

6.18.32 show ptp monitor

The **show ptp monitor** command to displays the list of up to 100 recorded entries of offset from master, mean path delay and skew values, along with current PTP mode, whether or not the feature is enabled, number of entries displayed and the configured thresholds for each metric. Entries are sorted by the system time at which the value was calculated, starting with the most recent data at the top.

Command Mode

EXEC

Command Syntax

show ptp monitor

Example

This command displays the information recorded by PTP monitoring.

```
switch# show ptp monitor
PTP Mode: Boundary Clock
Ptp monitoring: enabled
Number of entries: 5
Offset from master threshold: 1500
```

```
Mean path delay threshold: not configured
Skew threshold: 0.5
Interface  Time                          Offset from   Mean Path     Skew
                                         Master (ns)   Delay (ns)
---------- ----------------------------- ------------  ------------- -----------
Et8        21:23:12.901 UTC Feb 22 2018  71            5849           1.003159918
Et1        21:23:12.901 UTC Feb 22 2018  113           3672           1.004990621
Et2        21:23:12.901 UTC Feb 22 2018  706           7799           1.002744199
Et1        21:23:12.901 UTC Feb 22 2018  803           5861           1.003432049
Et1        21:23:12.901 UTC Feb 22 2018  610           3415           0.998974658
```

6.18.33 show ptp source ip

The **show ptp source ip** command displays the PTP source IP for the device.

Command Mode

Privileged EXEC

Command Syntax

show ptp source ip

Example

This command shows the PTP source IP to be 10.0.2.1.

```
switch# show ptp source ip
PTP source IP: 10.0.2.1 switch#
```

6.18.34 show ptp unicast-negotiation

The **show ptp unicast-negotiation** command displays PTP unicast negotiation information for the switch.

Command Mode

EXEC

Command Syntax

show ptp unicast-negotiation profile

show ptp unicast-negotiation [**DETAILS**]

Parameters

DETAILS Details requested. Options include:

- **candidate-grantor** Display all configured candidate grantors, associated profile name and latest update.
- **remote-grantee** Display all configured remote grantees, associated profile name and latest update.
- **granted** Display to which remote grantees each port has granted and some detail.
- **requested** Display to which candidate grantors each port has requested and some detail.

Examples

- This command displays the unicast negotiation profiles configured on the switch.

```
switch# show ptp unicast-negotiation profile
Unicast Negotiation Profile fastProfile
Announce interval: 0.25 seconds
Announce duration: 500 seconds
Sync interval: 0.125 seconds
Sync duration: 300 seconds
Delay Response interval: 0.125 seconds
Delay Response duration: 300 seconds
```

- This command displays all configured candidate grantors, associated profile name and latest update.

```
switch# show ptp unicast-negotiation candidate-grantor
 Interface  Address    Profile        Grantor Status
---------- ---------- -------------- ------------------
 Ethernet1  4::1       fastProfile    Master
 Ethernet1  4::2       fastProfile    Candidate Master
 Ethernet2  4::2       fastProfile    Blacklisted
```

- This command displays all configured remote grantees, associated profile name and latest update.

```
switch# show ptp unicast-negotiation remote-grantee
 Interface    Address    Profile
------------ ---------- -------------
 Ethernet1    4::1/96    fastProfile
 Ethernet1    4::2/96    fastProfile
 Ethernet2    4::2/96    fastProfile
```

- This command displays to which remote grantees each port has granted and some detail.

```
switch# show ptp unicast-negotiation granted
Interface    Address    Message     Interval       Duration       Expires In
---------- ---------- ------------ -------------- -------------- -----------
 Ethernet2   4::1       Announce    0.25 seconds   300 seconds    30 seconds
 Ethernet2   4::1       Sync        2.0 seconds    300 seconds    30 seconds
```

- This command displays to which candidate grantors each port has requested and some detail.

```
switch# show ptp unicast-negotiation requested
Interface    Address   Message     Interval        Duration       Expires In
------------ --------- ----------- --------------- -------------- ------------
 Ethernet2   4::2      Announce    0.25 seconds    600 seconds    250 seconds
 Ethernet2   4::2      Sync        2.0 seconds     300 seconds    denied
```

6.18.35 show ptp unicast-negotiation profile

The **show ptp unicast-negotiation profile** command displays all user configured profiles and their values.

Command Mode

EXEC

Command Syntax

show ptp unicast-negotiation profile

Example

This command displays the PTP profiles.

```
switch# show ptp unicast-negotiation profile
Unicast Negotiation Profile fastProfile
Announce interval: 0.25 seconds
Announce duration: 500 seconds
Sync interval: 0.125 seconds
Sync duration: 300 seconds
Delay Response interval: 0.125 seconds
Delay Response duration: 300 seconds
```

Chapter 7

Switch Environment Control

The following sections describe the commands that display temperature, fan, and power supply status:

- Environment Control Introduction
- Environment Control Overview
- Configuring and Viewing Environment Settings
- Environment Commands

The switch chassis, fans, power supplies, line cards, and supervisors also provide LEDs that signal status and conditions that require attention. The Quick Start Guide for the individual switches provides information about their LEDs.

7.1 Environment Control Introduction

Arista Networks switching platforms are designed to work reliably in common data center environments.

To ensure their reliable operation and to monitor or diagnose the switch's health, Arista provides a set of monitoring capabilities available through the CLI or SNMP entity MIBs to monitor and diagnose potential problems with the switching platform.

7.2 Environment Control Overview

This section contains the following topics:

- Temperature
- Fans
- Power

7.2.1 Temperature

Arista switches include internal temperature sensors. The number and location of the sensors vary with each switch model. Each sensor is assigned temperature thresholds that denote alert and critical conditions. Temperatures that exceed the threshold trigger the following:

- **Alert Threshold**: All fans run at maximum speed and a warning message is logged.
- **Critical Threshold**: The component is shut down immediately and its Status LED flashes orange.

In modular systems, cards are shut down when their temperatures exceed the critical threshold. The switch is shut down if the temperature remains above the critical threshold for three minutes.

7.2.2 Fans

Arista switches include fan modules that maintain internal components at proper operating temperatures. The number and type of fans vary with switch chassis type:

- **Fixed configuration switches** contain hot-swappable independent fans. Fan models with different airflow directions are available. All fans within a switch must have the same airflow direction.
- **Modular switches** contain independent fans that circulate air from front-to-rear panel. Power supplies for modular switches also include fans that cool the power supply and supervisors.

The switch operates normally when one fan is not operating. Non-functioning modules should not be removed from the switch unless they are immediately replaced; adequate switch cooling requires the installation of all components, including a non-functional fan.

Two non-operational fans trigger an ***insufficient fan shutdown*** condition. Under normal operations, this condition initiates a switch power down procedure.

Fans are accessible from the rear panel.

7.2.3 Power

Arista switches contain power supplies which provide power to internal components.

- **Fixed configuration switches** contain two power supplies, providing 1+1 redundancy.
- **Modular switches** contain four power supplies, providing a minimum of 2+2 redundancy.

Power supply LED indicators are visible from the rear panel.

7.3 Configuring and Viewing Environment Settings

This section contain the following topics:

- Overriding Automatic Shutdown
- Viewing Environment Status
- Locating Components on the Switch

7.3.1 Overriding Automatic Shutdown

This section contains the following topics:

- Overheating
- Insufficient Fans
- Fan Speed

7.3.1.1 Overheating

The switch can be configured to continue operating during temperature shutdown conditions. Ignoring a temperature shutdown condition is strongly discouraged because operating at high temperatures can damage the switch and void the warranty.

Temperature shutdown condition actions are specified by the environment overheat action command. The switch displays this warning when configured to ignore shutdown temperature conditions.

```
Switch(config)# environment overheat action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
is overheating is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment overheat action shutdown' command.
======================================================================
Switch(config)#
```

The ***running-config*** contains the **`environment overheat action`** command when it is set to **ignore**. When the command is not in ***running-config***, the switch shuts down when an overheating condition exists.

The following ***running-config*** file lists the **`environment overheat action`** command.

```
switch# show running-config
! Command: show running-config
! device: switch (DCS-7150S-64-CL, EOS-4.13.2F)

ip route 0.0.0.0/0 10.255.255.1
!
environment overheat action ignore
!
!
end switch#
```

7.3.1.2 Insufficient Fans

The switch can be configured to ignore the ***insufficient fan shutdown*** condition. This is strongly discouraged because continued operation without sufficient cooling may lead to a critical temperature condition that can damage the switch and void the warranty.

Insufficient-fans shutdown override is configured by the environment insufficient-fans action command. The switch displays this warning when configured to ignore insufficient-fan conditions.

```
switch(config)# environment insufficient-fans action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
has insufficient fans inserted is unsupported and should only be done
under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment insufficient-fans action shutdown' command.
======================================================================
switch(config)#
```

The ***running-config*** contains the **`environment insufficient-fans action`** command when it is set to ***ignore***. When ***running-config*** does not contain this command, the switch shuts down when it detects an insufficient-fans condition.

7.3.1.3 Fan Speed

The switch can be configured to override the automatic fan speed. The switch normally controls the fan speed to maintain optimal operating temperatures. The fans can be configured to operate at a constant speed regardless of the switch temperature conditions.

Fan speed override is configured by the environment fan-speed command. The switch displays this warning when its control of fan speed is overridden.

```
switch(config)# environment fan-speed override 50
======================================================================
WARNING: Overriding the system fan speed is unsupported and should only
be done under the direction of an Arista Networks engineer.
You can risk damaging hardware by setting the fan speed too low
and doing so without direction from Arista Networks can be grounds
for voiding your warranty.
To set the fan speed back to automatic mode, use the
'environment fan-speed auto' command
======================================================================
switch(config)#
```

The ***running-config*** contains the **`environment fan-speed override`** command if it is set to override. When ***running-config*** does not contain this command, the switch controls the fan speed.

7.3.2 Viewing Environment Status

This section contains the following topics:

- Power Status
- Temperature Status
- Fan Status
- System Status

7.3.2.1 Power Status

The show environment power command displays the status of the power supplies.

Example

This command displays the status of the power supplies.

```
switch> show environment power
Power Input Output Output
Supply  Model         Capacity  Current  Current  Power    Status
------- ------------- --------- -------- -------- -------- -------
1       PWR-650AC      650W      0.44A    10.50A  124.0W   Ok Switch>
```

7.3.2.2 Temperature Status

To display internal temperature sensor status, enter show environment temperature.

```
switch> show environment temperature
System temperature status is: Ok
                                              Alert      Critical
Sensor  Description              Temperature  Threshold  Threshold
------- ------------------------ ------------ ---------- ----------
1       Front-panel temp sensor   22.000C       65C        75C
2       Fan controller 1 sensor   23.000C       75C        85C
3       Fan controller 2 sensor   28.000C       75C        85C
4       Switch chip 1 sensor      40.000C       105C       115C
5       VRM 1 temp sensor         48.000C       105C       110C switch>
```

System temperature status is the first line that the command displays. **System temperature status** values indicate the following:

- **Ok:** All sensors report temperatures below the alert threshold.
- **Overheating:** At least one sensor reports a temperature above its alert threshold.
- **Critical:** At least one sensor reports a temperature above its critical threshold.
- **Unknown:** The switch is initializing.
- **Sensor Failed:** At least one sensor is not functioning.

7.3.2.3 Fan Status

The show system environment cooling command displays the cooling and fan status.

Example

This command displays the fan and cooling status:

```
switch> show system environment cooling
System cooling status is: Ok
```

```
Ambient temperature: 22C
Airflow: port-side-intake
Fan Tray  Status          Speed
--------- --------------- ------
1 Ok 35%
2 Ok 35%
3 Ok 35%
4 Ok 35%
5 Ok 35%
switch>
```

7.3.2.4 System Status

The show system environment all command lists the temperature, cooling, fan, and power supply information that the individual show environment commands display, as described in Temperature Status, Fans, and Power.

Example

This command displays the temperature, cooling, fan, and power supply status:

```
switch> show system environment all
System temperature status is: Ok
                                                  Alert      Critical
Sensor  Description              Temperature      Threshold  Threshold
------- ------------------------ ---------------- ---------- ----------
1       Front-panel temp sensor   22.750C          65C        75C
2       Fan controller 1 sensor   24.000C          75C        85C
3       Fan controller 2 sensor   29.000C          75C        85C
4       Switch chip 1 sensor      41.000C          105C       115C
5       VRM 1 temp sensor         49.000C          105C       110C

System cooling status is: Ok
Ambient temperature: 22C
Airflow: port-side-intake
Fan Tray  Status           Speed
--------- ---------------- ------
1         Ok                 35%
2         Ok                 35%
3         Ok                 35%
4         Ok                 35%
5         Ok                 35%

Power                          Input    Output   Output
Supply  Model       Capacity   Current  Current  Power    Status
------- ----------- ---------  -------- -------- -------- -------
1       PWR-650AC   650W        0.44A   10.50A   124.0W   Ok
```

7.3.3 Locating Components on the Switch

When a component requires service, the switch administrator may use the locator-led command to assist a technician in finding the component. The command causes the status LED on the specified component to flash, and also displays a “service requested” message on the LCD panel of modular switches or lights the blue locator light on the front of fixed switches. Use the show locator-led command to display all locator LEDs currently enabled on the switch.

- This command enables the locator LED on fan tray ***3***:

```
switch# locator-led fantray 3
Enabling locator led for FanTray3 switch#
```

- This command displays all locator LEDs enabled on the switch:

```
switch# show locator-led
There are no locator LED enabled switch#
```

7.4 Environment Commands

Environment Control Configuration Commands

- environment fan-speed
- environment insufficient-fans action
- environment overheat action
- locator-led

Environment Display Commands

- show environment power
- show environment temperature
- show locator-led
- show system environment all
- show system environment cooling
- show system environment power budget

7.4.1 environment fan-speed

The **`environment fan-speed`** command determines the method of controlling the speed of the switch fans. The switch automatically controls the fan speed by default.

The switch normally controls the fan speed to maintain optimal operating temperatures. The fans can be configured to operate at a constant speed regardless of the switch temperature conditions.

The **`no environment fan-speed`** and **`default environment fan-speed`** commands restore the default action of automatic fan-speed control by removing the **environment fan-speed override** statement from ***running-config***.

Note: Overriding the system fan speed is unsupported and should only be done under the direction of an Arista Networks engineer. You can risk damaging hardware by setting the fan speed too low. Doing so without direction from Arista Networks can be grounds for voiding your warranty.

Command Mode

Global Configuration

Command Syntax

`environment fan-speed` **ACTION**

`no environment fan-speed`

`default environment fan-speed`

Parameters

ACTION Fan speed control method. Valid settings include:

- **auto** Fan speed is controlled by the switch.

  This option restores the default setting by removing the **`environment fan-speed override`** command from ***running-config***
- **override *percent*** Fan speed is set to the specified percentage of the maximum. Valid ***percent*** settings range from ***30*** to ***100***.

Examples

- This command overrides the automatic fan speed control and configures the fans to operate at 50% of maximum speed.

```
switch(config)# environment fan-speed override 50
==========================================================================
WARNING: Overriding the system fan speed is unsupported and should
only be done under the direction of an Arista Networks engineer.
You can risk damaging hardware by setting the fan speed too low
and doing so without direction from Arista Networks can be grounds
for voiding your warranty.
To set the fan speed back to automatic mode, use the
'environment fan-speed auto' command
==========================================================================
switch(config)#
```

- This command restores control of the fan speed to the switch.

```
switch(config)# environment fan-speed auto
switch(config)#
```

7.4.2 environment insufficient-fans action

The **`environment insufficient-fans`** command controls the switch response to the insufficient fan condition. By default, the switch initiates a shutdown procedure when it senses insufficient fans.

The switch operates normally when one fan is not operating. Non-functioning modules should not be removed from the switch unless they are immediately replaced; adequate switch cooling requires the installation of all components, including a non-functional fan.

Two non-operational fans trigger an ***insufficient fan shutdown*** condition. This condition normally initiates a power down procedure.

The **`no environment insufficient-fans`** and **`default environment insufficient-fans`** commands restore the default shutdown response to the insufficient-fans condition by removing the **`environment insufficient-fans action ignore`** statement from ***running-config***.

Note: Overriding the system shutdown behavior when the system has insufficient fans inserted is unsupported and should only be done under the direction of an Arista Networks engineer. You risk damaging hardware by not shutting down the system in this situation, and doing so without direction from Arista Networks can be grounds for voiding your warranty.

Command Mode

Global Configuration

Command Syntax

`environment insufficient-fans action` **REMEDY**

`no environment insufficient-fans action`

`default environment insufficient-fans action`

Parameters

REMEDY Configures action when switch senses an insufficient fan condition. Settings include:

- **ignore** Switch continues operating when insufficient fans are operating.
- **shutdown** Switch shuts power down when insufficient fans are operating.

The **shutdown** parameter restores default behavior by removing the **`environment insufficient-fans`** command from ***running-config***.

Examples

- This command configures the switch to continue operating after it senses insufficient fan condition.

```
switch(config)# environment insufficient-fans action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
has insufficient fans inserted is unsupported and should only be done
under the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment insufficient-fans action shutdown' command.
======================================================================
```

- This command configures the switch to shut down when it senses an insufficient fan condition.

```
switch(config)# environment insufficient-fans action shutdown
switch(config)#
```

7.4.3 environment overheat action

The **`environment overheat`** command controls the switch response to an overheat condition. By default, the switch shuts down when it senses an overheat condition.

Note: Overriding the system shutdown behavior when the system is overheating is unsupported and should only be done under the direction of an Arista Networks engineer. You risk damaging hardware by not shutting down the system in this situation, and doing so without direction from Arista Networks can be grounds for voiding your warranty.

Arista switches include internal temperature sensors. The number and location of the sensors vary with each switch model. Each sensor is assigned temperature thresholds that denote alert and critical conditions. Temperatures that exceed the threshold trigger the following:

- **Alert Threshold**: All fans run at maximum speed and a warning message is logged.
- **Critical Threshold**: The component is shut down immediately and its Status LED flashes orange.

In modular systems, cards are shut down when their temperatures exceed the critical threshold. The switch normally shuts down if the temperature remains above the critical threshold for three minutes.

The **`no environment overheat action`** and **`default environment overheat action`** commands restore the default shutdown response to the environment overheat condition by removing the **`environment overheat action ignore`** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`environment overheat action` **REMEDY**

`no environment overheat action`

default environment overheat action

Parameters

REMEDY Reaction to an overheat condition. Default value is **shutdown**.

- **shutdown** Switch shuts power down by an overheat condition.
- **ignore** Switch continues operating during an overheat condition.

Examples

- This command configures the switch to continue operating after it senses an overheat condition.

```
switch(config)# environment overheat action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
is overheating is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment overheat action shutdown' command.
======================================================================
switch(config)#
```

- This command configures the switch to shut down when it senses an overheat condition.

```
switch(config)# environment overheat action shutdown
switch(config)#
```

7.4.4 locator-led

When a component requires service, the **locator-led** command activates a locator to assist a technician in finding the component. The command causes the status LED on the specified component to flash, and also displays a “service requested” message on the LCD panel of modular switches or lights the blue locator light on the front of fixed switches. The available locators vary by platform; to see a list of the locator LEDs available on the switch, use the **locator-led ?** command. To disable the locator LED, use the **no locator-led** command.

Command Mode

Privileged EXEC

Command Syntax

locator-led {fantray ***tray_num*** | interface ***interface*** | module ***module_num*** | powersupply ***supply_num***}

no locator-led {fantray ***tray_num*** | interface ***interface*** | module ***module_num*** | powersupply ***supply_num***}

Parameters

- **fantray *tray_num*** Activates locator on specified fan tray.
- **interface *interface*** Activates locator on specified interface.
- **module *module_num*** Activates locator on specified module.
- **powersupply *supply_num*** Activates locator on specified power supply.

Examples:

- This command enables the locator LED on ***fantray 3***.

```
switch# locator-led fantray 3
Enabling locator led for FanTray3 switch#
```

- This command disables the locator LED on ***fantray 3***.

```
switch# no locator-led fantray 3
Disabling locator led for FanTray3 switch#
```

- This command displays the locator LEDs available on the switch.

```
switch# locator-led ?
  fantray      Fan tray LED
  interface    Interface LED
  module       Module LED
  powersupply  Power supply LED switch#
```

7.4.5 show environment power

The **`show environment power`** command displays the status of all power supplies in the switch.

Command Mode

EXEC

Command Syntax

`show environment power` [**INFO_LEVEL**]

Parameters

INFO_LEVEL Specifies level of detail that the command displays. Options include:

- ***no parameter*** Displays current and power levels for each supply.
- **detail** Also includes status codes that can report error conditions.

Example

This command displays the status of power supplies on the switch.

```
switch>show environment power
Power                          Input    Output   Output
Supply  Model      Capacity  Current  Current  Power    Status
------- ---------- --------- -------- -------- -------- -------
1       PWR-760AC  760W       0.81A    11.00A   132.8W   Ok
2       PWR-760AC  760W       0.00A    0.00A    0.0W AC  Loss switch>
```

7.4.6 show environment temperature

The **`show environment temperature`** command displays the operating temperature of all sensors on the switch.

Command Mode

EXEC

Command Syntax

`show environment temperature` [**MODULE_NAME**] [**INFO_LEVEL**]

Parameters

- **MODULE_NAME** Specifies modules for which data is displayed. This parameter is only available on modular switches. Options include:
  - ***no parameter*** All modules (identical to **all** option).
  - **fabric *fab_num*** Specified fabric module. Number range varies with switch model.
  - **linecard *line_num*** Line card module. Number range varies with switch model.
  - **supervisor *super_num*** Supervisor module. Number range varies with switch model.
  - ***mod_num*** Supervisor (1 to 2) or line card (3 to 18) module.
  - **all** All modules.
- **INFO_LEVEL** Specifies level of detail that the command displays. Options include:
  - ***no parameter*** Displays table that lists the temperature and thresholds of each sensor.
  - **detail** Displays data block for each sensor listing the current temperature and historic data.

Display Values

System temperature status is the first line that the command displays. Values report the following:

- ***Ok*** All sensors report temperatures below the alert threshold.
- ***Overheating*** At least one sensor reports a temperature above its alert threshold.
- ***Critical*** At least one sensor reports a temperature above its critical threshold.
- ***Unknown*** The switch is initializing.
- ***Sensor Failed*** At least one sensor is not functioning.

Examples

- This command displays a table that lists the temperature measured by each sensor.

```
switch> show environment temperature
System temperature status is: Ok
                                                 Alert      Critical
Sensor  Description              Temperature  Threshold  Threshold
------- ------------------------ ------------ ---------- ----------
1       Front-panel temp sensor   30.750C       65C        75C
2       Fan controller 1 sensor   32.000C       75C        85C
3       Fan controller 2 sensor   38.000C       75C        85C
4       Switch chip 1 sensor      50.000C       105C       115C
5       VRM 1 temp sensor         60.000C       105C       110C
 switch>
```

- This command lists the temperature detected by each sensor, and includes the number of previous alerts, the time of the last alert, and the time of the last temperature change.

```
switch> show environment temperature detail
TempSensor1 - Front-panel temp sensor
                      Current State   Count    Last Change
  Temperature         30.750C
  Max Temperature     35.000C                  4 days, 23:35:24 ago
  Alert                False           0        never

TempSensor2 - Fan controller 1 sensor
                      Current State    Count    Last Change
  Temperature          32.000C
  Max Temperature      36.000C                  4 days, 23:32:46
 ago
  Alert                 False           0        never

TempSensor3 - Fan controller 2 sensor
                       Current State    Count     Last Change
```

```
  Temperature             38.000C
  Max Temperature         41.000C                    4 days, 23:37:56
 ago
  Alert                   False             0        never

TempSensor4 - Switch chip 1 sensor
                         Current State    Count    Last Change
  Temperature             51.000C
  Max Temperature         53.000C                     4 days, 23:35:16
 ago
  Alert                   False               0       never

TempSensor5 - VRM 1 temp sensor
                          Current State     Count    Last Change
  Temperature              60.000C
  Max Temperature          62.000C                     4 days,
 22:54:51 ago
  Alert                    False              0        never switch>
```

7.4.7 show locator-led

The **`show locator-led`** command displays the status of locator LEDs enabled on the switch.

Command Mode

Privileged EXEC

Command Syntax

`show locator-led`

Example

This command displays all locator LEDs enabled on the switch.

```
switch# show locator-led
There are no locator LED enabled switch#
```

7.4.8 show system environment all

The **`show system environment all`** command displays temperature, cooling, and power supply status.

Command Mode

EXEC

Command Syntax

`show system environment all`

Example

This command displays the switch's temperature, cooling, and power supply status

```
switch> show system environment all
System temperature status is: Ok
                                              Alert      Critical
Sensor  Description              Temperature  Threshold  Threshold
------- ------------------------ ------------ ---------- ----------
1       Front-panel temp sensor   31.000C       65C        75C
```

```
2       Fan controller 1 sensor   32.000C       75C        85C
3       Fan controller 2 sensor   38.000C       75C        85C
4       Switch chip 1 sensor      50.000C       105C       115C
5       VRM 1 temp sensor         60.000C       105C       110C

System cooling status is: Ok
Ambient temperature: 31C
Airflow: port-side-intake
Fan Tray  Status           Speed
--------- ---------------- ------
1         Ok                 52%
2         Ok                 52%
3         Ok                 52%
4         Ok                 52%
5         Ok                 52%

Power                          Input    Output   Output
Supply  Model       Capacity   Current  Current  Power    Status
------- ----------- ---------  -------- -------- -------- -------
1       PWR-760AC    760W      0.81A    11.00A    132.6W  Ok
2       PWR-760AC    760W      0.00A     0.00A      0.0W  AC Loss switch>
```

7.4.9 show system environment cooling

The **`show system environment cooling`** command displays fan status, air flow direction, and ambient temperature on the switch.

Command Mode

EXEC

Command Syntax

`show system environment cooling` [**INFO_LEVEL**]

Parameters

- **INFO_LEVEL** Specifies level of detail that the command displays. Options include:
  - ***no parameter*** Displays the fan status, air flow direction, and ambient switch temperature.
  - **detail** Also displays actual and configured fan speed of each fan.

Status

- System Cooling Status
  - **`Ok`** No more than one fan has failed or is not inserted.
  - **`Insufficient fans`** More than one fan has failed or is not inserted. This status is also displayed if fans with different airflow directions are installed. The switch shuts down if the error is not resolved.
  - **`Ambient temperature`** Temperature of the surrounding area.
  - **`Airflow`** Indicates the direction of the installed fans:
    - **`port-side-intake`** All fans flow air from the front (port side) to the rear of the chassis.
    - **`port-side-exhaust`** All fans flow air from the rear to the front (port side) of the chassis.
    - **`incompatible fans`** Fans with different airflow directions are inserted.
    - **`Unknown`** The switch is initializing.
- **Fan Tray Status** table Displays the status and operating speed of each fan. Status values indicate the following conditions:
  - **`OK`** The fan is operating normally.
  - **`Failed`** The fan is not operating normally.
  - **`Unknown`** The system is initializing.
  - **`Not Inserted`** The system is unable to detect the specified fan.
  - **`Unsupported`** The system detects a fan that the current software version does not support.

Example

This command displays the fan status, air flow direction, and ambient switch temperature.

```
switch> show system environment cooling
System cooling status is: Ok
Ambient temperature: 30C
Airflow: port-side-intake
Fan Tray  Status           Speed
--------- ---------------- ------
1         Ok                 51%
2         Ok                 51%
3         Ok                 51%
4         Ok                 51%
5         Ok                 51%
switch>
```

7.4.10 show system environment power budget

The **show system environment power budget** command displays the configured power budget for PoE switches and a table to show the current consumption.

Command Mode

EXEC

Command Syntax

show system environment power budget

This command displays the configured power budget for a PoE switch and a table to show the current consumption

```
switch>show system environment power budget
Budget is 300.0W out of 1425.2W of total power
Active devices using up to 60.0W

Device                  Consumed
Name                    Power
----------------------- -----------------------
Ethernet37              30.0W
Ethernet38              30.0W
Total                   60.0W switch>
```

Chapter 8

Upgrades and Downgrades

This chapter describes the procedures for upgrading or downgrading the switch software.

This chapter contains these sections:

- Upgrade/Downgrade Overview
- Smart System Upgrade
- Standard Upgrades and Downgrades
- Upgrade/Downgrade Commands

8.1 Upgrade/Downgrade Overview

Upgrading or downgrading the Arista Extensible Operating System (EOS) is accomplished by replacing the EOS image and reloading the switch. Depending on the switch model and the software change being made, it may be possible to minimize (or virtually eliminate) downtime and packet loss during an upgrade. There are two upgrade methods for the EOS:

Smart System Upgrade: SSU significantly decreases downtime and packet loss during upgrades. SSU is available on selected platforms, and is ideal for leaf switches and other non-redundant deployments.

Standard Upgrades and Downgrades: In those cases where an accelerated upgrade is not needed or not an option (such as for software downgrades and on unsupported platforms), performing a standard upgrade or downgrade using the steps described here will minimize downtime and packet loss.



Note: To upgrade the software on switches participating in an MLAG, see Upgrading MLAG Peers.

8.2 Smart System Upgrade

Smart System Upgrade (SSU) significantly reduces reload time by streamlining and optimizing the reload procedure for upgrades, and by continuing to send LACP PDUs while the CPU is rebooting, keeping port channels operational during the reload. SSU leverages protocols capable of graceful restart to minimize traffic loss during upgrade.

Features capable of hitless restart under SSU include:

- QinQ
- 802.3ad Link Aggregation/LACP
- 802.3x flow control
- BGP (BGP graceful restart must be enabled: see Configuring BGP)
- MP-BGP (BGP graceful restart must be enabled: see Configuring BGP)
- 128-way Equal Cost Multipath Routing (ECMP)
- VRF
- route maps
- L2 MTU
- QoS



Note: SSU is not compatible with VRRP. If VRRP is configured on the switch, another upgrade method must be used.

8.2.1 Upgrading the EOS image with Smart System Upgrade

Using SSU to upgrade the active EOS image is a five-step process:

1. Prepare switch for upgrade (Prepare the Switch for SSU).
2. Transfer image file to the switch (Transfer the Image File for SSU). (Not required if desired file is on the switch).
3. Modify ***boot-config*** file to point to the desired image file (Modify boot-config).
4. Start the SSU process (Start the SSU Process).
5. Verify that the upgrade was successful (Verify Success of the Upgrade).

8.2.1.1 Prepare the Switch for SSU

To prepare the switch for SSU, take the following steps:

- Backing Up Critical Software
- Making Room on the Flash Drive
- Verifying Connectivity
- Verifying Configuration
- Configuring BGP



Note: configuring BGP graceful restart resets BGP sessions. If configuring BGP graceful restart as part of the SSU process, ensure that BGP sessions are stable and all BGP routing information has been learned and advertised before proceeding with SSU.

Backing Up Critical Software

Before upgrading the EOS image, ensure that copies of the currently running EOS version and the ***running-config*** file are available in case of corruption during the upgrade process. To copy the ***running-config*** file, use the **`copy running-config`** command. In this example, ***running-config*** is copied to a file in the flash drive on the switch.

```
switch# copy running-config flash:/cfg_06162014
Copy completed successfully.
switch#
```

Making Room on the Flash Drive

Determine the size of the new EOS image. Then verify that there is enough space available on the flash drive for two copies of this image, plus a recommended 240MB (if available) for diagnostic information in case of a fatal error. Use the **`dir`** command to check the “bytes free” figure.

```
switch# dir flash:
Directory of flash:/
-rwx   293168526      Nov 4    22:17   EOS4.11.0.swi
-rwx          36      Nov 8    10:24   boot-config
-rwx       37339      Jun 16   14:18   cfg_06162014
606638080 bytes total (602841088 bytes free)
```

Verifying Connectivity

Ensure that the switch has a management interface configured with an IP addresses and default gateway. See Assigning a Virtual IP Address to Access the Active Ethernet Management Port and Configuring a Default Route to the Gateway. Confirm that the switch can be reached through the network by using the command and pinging the default gateway.

```
switch# show interfaces status
Port    Name     Status     Vlan       Duplex   Speed      Type
```

```
Et3/1            notconnect   1         auto    auto     1000BASE-T

<-------OUTPUT OMITTED FROM EXAMPLE-------->
Ma1/1            connected    routed    unconf   unconf    Unknown switch#ping
 1.1.1.10
PING 172.22.26.1 (172.22.26.1) 72(100) bytes of data.
80 bytes from 1.1.1.10: icmp_seq=1 ttl=64 time=0.180 ms
80 bytes from 1.1.1.10: icmp_seq=2 ttl=64 time=0.076 ms
80 bytes from 1.1.1.10: icmp_seq=3 ttl=64 time=0.084 ms
80 bytes from 1.1.1.10: icmp_seq=4 ttl=64 time=0.073 ms
80 bytes from 1.1.1.10: icmp_seq=5 ttl=64 time=0.071 ms
```

Verifying Configuration

Verify that the switch configuration is valid for SSU by using the **show reload fast-boot** command. If parts of the configuration are blocking execution of SSU, an error message will be displayed explaining what they are. For SSU to proceed, the configuration conflicts must be corrected before issuing the **reload fast-boot** command.

```
switch# show reload fast-boot
switch#'reload fast-boot' cannot proceed due to the following:
  Spanning-tree portfast is not enabled for one or more ports
  Spanning-tree BPDU guard is not enabled for one or more ports switch#
```

Note: The **show reload hitless** and **reload hitless** commands can still be used, but their effect is identical to the commands shown above.

Configuring BGP

For hitless restart of BGP and MP-BGP, BGP graceful restart must first be enabled using the **graceful-restart** command. The default restart time value (***300*** seconds) is appropriate for mostconfigurations.

The BGP configuration mode in which the **graceful-restart** command is issued determines which BGP connections will restart gracefully.

Note: configuring BGP graceful restart resets BGP sessions. If configuring BGP graceful restart as part of the SSU process, ensure that BGP sessions are stable and all BGP routing information has been learned and advertised before proceeding with SSU.

- For all BGP connections, use the **graceful-restart** command in BGP configuration mode:

```
switch# config
switch(config)# router bgp 64496
switch(config-router-bgp)# graceful-restart
switch(config-router-bgp)#
```

- For all BGP connections in a specific VRF, use the **graceful-restart** command in BGP VRF configuration mode:

```
switch# config
switch(config)# router bgp 64496
switch(config-router-bgp)# vrf purple
switch(config-router-bgp-vrf-purple)# graceful-restart
switch(config-router-bgp-vrf-purple)# exit
switch(config-router-bgp)#
```

- For all BGP connections in a specific BGP address family, use the **graceful-restart**command in BGP address-family configuration mode:

```
switch# config
switch(config)# router bgp 64496
```

```
switch(config-router-bgp)# address-family ipv6
switch(config-router-bgp-af)# graceful-restart
switch(config-router-bgp-af)# exit
switch(config-router-bgp)#
```

8.2.1.2 Transfer the Image File for SSU

The target image must be copied to the file system on the switch, typically onto the flash drive. After verifying that there is space for two copies of the image plus an optional 240MB for diagnostic information, use the **copy** command to copy the image to the flash drive, then confirm that the new image file has been correctly transferred.

These command examples transfer an image file to the flash drive from various locations.

USB Memory

Command

copy usb1:/***sourcefile*** flash:/***destfile***

Example

```
sch# copy usb1:/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

FTP Server

Command

copy ftp:/***ftp-source***/***sourcefile*** flash:/***destfile***

Example

```
switch# copy ftp:/user:password@10.0.0.3/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

SCP

Command

copy scp://***scp-source***/***sourcefile*** flash:/***destfile***

Example

```
switch# copy scp://user@10.1.1.8/user/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

HTTP

Command

copy http://***http-source***/***sourcefile*** flash:/***destfile***

Example

```
switch# copy http://10.0.0.10/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

Once the file has been transferred, verify that it is present in the directory, then confirm the MD5 checksum using the **verify** command. The MD5 checksum is available from the EOS download page of the Arista website.

```
switch# dir flash:
Directory of flash:/
-rwx     293168526   Nov 4     22:17     EOS4.14.2.swi
```

```
-rwx            36   Nov 8     10:24     boot-config
-rwx         37339   Jun 16    14:18     cfg_06162014
-rwx     394559902   May 30    02:57     EOS4.13.1.swi

606638080 bytes total (208281186 bytes free)
switch# verify /md5 flash:EOS-4.14.4.swi
verify /md5 (flash:EOS-4.14.4.swi) =c277a965d0ed48534de6647b12a86991 switch#
```

8.2.1.3 Modify boot-config

After transferring and confirming the desired image file, use the **`boot system`** command to update the ***boot-config*** file to point to the new EOS image.

This command changes the ***boot-config*** file to point to the image file located in flash memory at ***EOS-4.14.4.swi***.

```
switch# configure terminal
switch(config)# boot system flash:/EOS-4.14.4.swi
```

Use the **`show boot-config`** command to verify that the ***boot-config*** file is correct:

```
switch(config)# show boot-config
Software image: flash:/EOS-4.14.4.swi
Console speed: (not set)
Aboot password (encrypted): $1$ap1QMbmz$DTqsFYeauuMSa7/Qxbi2l1
```

Save the configuration to the ***startup-config*** file with the **`write`** command.

```
switch# write
```

8.2.1.4 Start the SSU Process

After updating the ***boot-config*** file, verify that your configuration supports SSU (if you have not already done so) by using the **`show reload fast-boot`** command. If parts of the configuration are blocking execution of SSU, an error message will be displayed explaining what they are.

```
switch# show reload fast-boot
switch#'reload fast-boot' cannot proceed due to the following:
  Spanning-tree portfast is not enabled for one or more ports
  Spanning-tree BPDU guard is not enabled for one or more ports switch#
```

Then start the SSU process using the **`reload fast-boot`** command to reload the switch and activate the new image. The CLI will identify any changes that must be made to the configuration before starting SSU, prompt to save any modifications to the system configuration, and request confirmation before reloading.

```
switch# reload fast-boot
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Copy completed successfully.
Proceed with reload? [confirm]y
```



Note: The **`show reload hitless`** and **`reload hitless`** commands can also be used, but their effect is identical to the commands shown above.

8.2.1.5 Verify Success of the Upgrade

Before making any configuration changes to the switch after reload, verify that the SSU process is complete using the command **`show boot stages log`**. If the process is complete, the last message should be "Hitless boot stages complete."

```
switch# show boot stages log
Timestamp           Delta Begin Msg
2022-10-03 12:42:06 000.000000 Asu Hitless boot stages started
2022-10-03 12:42:06 000.001592 stage CriticalAgent started
2022-10-03 12:42:06 000.001834   event CriticalAgent:PhyEthtool completed

[ . . . ]

2022-10-03 12:43:02 056.316874 stage BootSanityCheck is complete
2022-10-03 12:43:02 056.317491 Asu Hitless boot stages complete switch#
```

Completion of the SSU process may also be verified by checking the syslog for the following message:

```
LAUNCHER-6-BOOT_STATUS: 'reload fast-boot' reconciliation complete
```

To verify whether the SSU upgrade was successful, use the **`show reload cause`** command. If a fatal error occurred during the upgrade process, the switch will have completely rebooted and the fatal error will be displayed along with the directory in which diagnostic information can be found. If the SSU upgrade has succeeded, it will read "Hitless reload requested by the user."

Fatal Error Display

```
switch# show reload cause
Reload Cause 1:
-------------------
Fatal error occurred during Asu Hitless boot. (stageMgr - LinkStatusUpdate
 timed out)

Reload Time:
------------
Reload occurred at Sun Oct 02 12:06:37 2022 PDT.

Recommended Action:
-------------------
The system rebooted due to a fatal error.
If the problem persists, contact your customer support representative.

Debugging Information:
------------------------------
/mnt/flash/persist/fatalError-2022-10-02_120637 switch#
```

Successful Upgrade Display

```
switch# show reload cause
Reload Cause 1:
-------------------
Hitless reload requested by the user.

Reload Time:
------------
Reload occurred at Mon Oct 03 13:29:31 2022 PDT.

Recommended Action:
-------------------
```

```
No action necessary.

Debugging Information:
------------------------------
None available.
switch#
```

The **`show version`** command confirms whether the correct image is loaded. The ***Software image version*** line displays the version of the active image file.

```
switch# show version
switch#show version
Arista DCS-7050QX-32-F
Hardware version: 02.00
Serial number: JPE14071098
System MAC address: 001c.7355.556f
Software image version: 4.14.5F-2353054.EOS4145F
Architecture: i386
Internal build version: 4.14.5F-2353054.EOS4145F
Internal build ID: e8748ea7-916d-4217-878f-4bfe2adc7122
Uptime: 4 minutes
Total memory: 3981328 kB
Free memory: 1342408 kB switch#
```



Note: If a fatal error occurs during the SSU process, the new EOS image will still be loaded and booted.

8.3 Standard Upgrades and Downgrades

Standard software upgrades and downgrades on Arista switches are accomplished by installing a different EOS image and reloading the switch. On switches with redundant supervisors, the EOS image must be installed on both supervisors. Using the procedure described below will minimize packet loss during a standard upgrade or downgrade.

These sections describe standard switch upgrade and downgrade procedures.:

- Upgrading or Downgrading the EOS on a Single-Supervisor Switch
- Upgrading or Downgrading the EOS on a Dual-Supervisor Switch

8.3.1 Upgrading or Downgrading the EOS on a Single-Supervisor Switch

Modifying the active EOS image is a five-step process:

1. Prepare switch for upgrade (Prepare the Switch for SSU).
2. Transfer image file to the switch (Transfer the Image File). (Not required if desired file is on the switch).
3. Modify ***boot-config*** file to point to the desired image file (Modify boot-config for Single-Supervisor switch).
4. Reload switch (Reload).
5. Verify that switch is running the new image (Verify the New Image for Single-Supervisor Switch).

8.3.1.1 Prepare the Switch for SSU

To prepare the switch for SSU, take the following steps:

- Backing Up Critical Software
- Making Room on the Flash Drive
- Verifying Connectivity
- Verifying Configuration
- Configuring BGP

Note: configuring BGP graceful restart resets BGP sessions. If configuring BGP graceful restart as part of the SSU process, ensure that BGP sessions are stable and all BGP routing information has been learned and advertised before proceeding with SSU.

Backing Up Critical Software

Before upgrading the EOS image, ensure that copies of the currently running EOS version and the ***running-config*** file are available in case of corruption during the upgrade process. To copy the ***running-config*** file, use the **`copy running-config`** command. In this example, ***running-config*** is copied to a file in the flash drive on the switch.

```
switch# copy running-config flash:/cfg_06162014
Copy completed successfully.
switch#
```

Making Room on the Flash Drive

Determine the size of the new EOS image. Then verify that there is enough space available on the flash drive for two copies of this image, plus a recommended 240MB (if available) for diagnostic information in case of a fatal error. Use the **`dir`** command to check the “bytes free” figure.

```
switch# dir flash:
Directory of flash:/
-rwx   293168526      Nov 4    22:17   EOS4.11.0.swi
-rwx          36      Nov 8    10:24   boot-config
-rwx       37339      Jun 16   14:18   cfg_06162014
606638080 bytes total (602841088 bytes free)
```

Verifying Connectivity

Ensure that the switch has a management interface configured with an IP addresses and default gateway. See Assigning a Virtual IP Address to Access the Active Ethernet Management Port and Configuring a Default Route to the Gateway. Confirm that the switch can be reached through the network by using the command and pinging the default gateway.

```
switch# show interfaces status
Port    Name     Status     Vlan       Duplex   Speed      Type
Et3/1            notconnect   1         auto    auto     1000BASE-T

<-------OUTPUT OMITTED FROM EXAMPLE-------->
Ma1/1            connected   routed     unconf   unconf    Unknown switch#ping
 1.1.1.10
PING 172.22.26.1 (172.22.26.1) 72(100) bytes of data.
80 bytes from 1.1.1.10: icmp_seq=1 ttl=64 time=0.180 ms
80 bytes from 1.1.1.10: icmp_seq=2 ttl=64 time=0.076 ms
80 bytes from 1.1.1.10: icmp_seq=3 ttl=64 time=0.084 ms
80 bytes from 1.1.1.10: icmp_seq=4 ttl=64 time=0.073 ms
80 bytes from 1.1.1.10: icmp_seq=5 ttl=64 time=0.071 ms
```

Verifying Configuration

Verify that the switch configuration is valid for SSU by using the **`show reload fast-boot`** command. If parts of the configuration are blocking execution of SSU, an error message will be displayed explaining what they are. For SSU to proceed, the configuration conflicts must be corrected before issuing the **`reload fast-boot`** command.

```
switch# show reload fast-boot
switch#'reload fast-boot' cannot proceed due to the following:
```

```
Spanning-tree portfast is not enabled for one or more ports
Spanning-tree BPDU guard is not enabled for one or more ports switch#
```

Note: The **show reload hitless** and **reload hitless** commands can still be used, but their effect is identical to the commands shown above.

Configuring BGP

For hitless restart of BGP and MP-BGP, BGP graceful restart must first be enabled using the **graceful-restart** command. The default restart time value (***300*** seconds) is appropriate for mostconfigurations.

The BGP configuration mode in which the **graceful-restart** command is issued determines which BGP connections will restart gracefully.

Note: configuring BGP graceful restart resets BGP sessions. If configuring BGP graceful restart as part of the SSU process, ensure that BGP sessions are stable and all BGP routing information has been learned and advertised before proceeding with SSU.

- For all BGP connections, use the **graceful-restart** command in BGP configuration mode:

```
switch# config
switch(config)# router bgp 64496
switch(config-router-bgp)# graceful-restart
switch(config-router-bgp)#
```

- For all BGP connections in a specific VRF, use the **graceful-restart** command in BGP VRF configuration mode:

```
switch# config
switch(config)# router bgp 64496
switch(config-router-bgp)# vrf purple
switch(config-router-bgp-vrf-purple)# graceful-restart
switch(config-router-bgp-vrf-purple)# exit
switch(config-router-bgp)#
```

- For all BGP connections in a specific BGP address family, use the **graceful-restart**command in BGP address-family configuration mode:

```
switch# config
switch(config)# router bgp 64496
switch(config-router-bgp)# address-family ipv6
switch(config-router-bgp-af)# graceful-restart
switch(config-router-bgp-af)# exit
switch(config-router-bgp)#
```

8.3.1.2 Transfer the Image File

The target image must be copied to the file system on the switch, typically onto the flash drive. After verifying that there is space for the image, use the CLI **copy** command to copy the image to the flash drive, then confirm that the new image file has been correctly transferred.

These command examples transfer an image file to the flash drive from various locations.

USB Memory

Command

copy usb1:/***sourcefile*** flash:/***destfile***

Example

```
sch# copy usb1:/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

FTP Server

Command

copy ftp:/***ftp-source***/***sourcefile*** flash:/***destfile***

Example

```
sch# copy ftp:/user:password@10.0.0.3/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

SCP

Command

copy scp://***scp-source***/***sourcefile*** flash:/***destfile***

Example

```
sch# copy scp://user@10.1.1.8/user/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

HTTP

Command

copy http://***http-source***/***sourcefile*** flash:/***destfile***

Example

```
sch# copy http://10.0.0.10/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

Once the file has been transferred, verify that it is present in the directory, then confirm the MD5 checksum using the **verify** command. The MD5 checksum is available from the EOS download page of the Arista website.

```
switch# dir flash:
Directory of flash:/
-rwx     293168526   Nov 4     22:17     EOS4.11.0.swi
-rwx            36   Nov 8     10:24     boot-config
-rwx         37339   Jun 16    14:18     cfg_06162014
-rwx     394559902   May 30    02:57     EOS-4.12.2.swi

606638080 bytes total (208281186 bytes free)
switch#53# verify /md5 flash:EOS-4.13.2.swi
verify /md5 (flash:EOS-4.13.2.swi) =c277a965d0ed48534de6647b12a86991
```

8.3.1.3 Modify boot-config for Single-Supervisor switch

After transferring and confirming the desired image file, use the **boot system** command to update the ***boot-config*** file to point to the new EOS image.

This command changes the ***boot-config*** file to point to the image file located in flash memory at ***EOS-4.12.2.swi***.

```
switch# configure terminal
switch(config)# boot system flash:/EOS-4.13.2.swi
```

Use the **`show boot-config`** command to verify that the ***boot-config*** file is correct:

```
switch(config)# show boot-config
Software image: flash:/EOS-4.13.2.swi
Console speed: (not set)
Aboot password (encrypted): $1$ap1QMbmz$DTqsFYeauuMSa7/Qxbi2l1
```

Save the configuration to the ***startup-config*** file with the **`write`** command.

```
switch# write
```

8.3.1.4 Reload

After updating the ***boot-config*** file, reset the switch to activate the new image. The reload command resets the switch, resulting in temporary downtime and packet loss on single supervisor switches.

When reloading from the console port, all rebooting messages are displayed on the terminal. From any port except the console, the CLI displays this text:

```
switch# reload
The system is going down for reboot NOW!
```



Note: The EOS boot process makes a copy of the .swi image file in the internal flash while booting, so sufficient space for ***two copies*** must be present when loading the new EOS image. If the switch is reloaded without adequate space on the flash drive, it will boot to the Aboot prompt from which you can delete files from /mnt/flash to free up additional space. Exiting Aboot will begin the boot process again.

8.3.1.5 Verify the New Image for Single-Supervisor Switch

After the switch finishes reloading, log into the switch and use the **`show version`** command to confirm the correct image is loaded. The **`Software image version`** line displays the version of the active image file.

```
switch# show version
Arista DCS-7150S-64-CL-F
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.13.2F
Architecture:           i386
Internal build version: 4.13.2F-1649184.4132F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

Uptime:                 14 hours and 48 minutes
Total memory:           4017088 kB
Free memory:            1569760 kB switch>
```

8.3.2 Upgrading or Downgrading the EOS on a Dual-Supervisor Switch

Modifying the active EOS image is a four-step process:

1. Prepare switch for upgrade (Prepare the Switch for Dual-Supervisor Switch).
2. Transfer image file to primary supervisor (Transfer the Image File to the Primary Supervisor). (Not required if desired file is on switch).
3. Use the install command to install the new EOS image and update ***boot-config*** (Install the New EOS Image).
4. Verify that the switch is running the new image (Verify the New Image).

Note: Due to a change in the supervisor heartbeat timeout, booting one supervisor with a post-SSO image (version ***4.10.0-SSO***, ***4.11.X*** and later) while the other supervisor is running a pre-SSO image will cause the supervisor running the pre-SSO image to reload. This causes a disruption as both supervisors are inactive for a short time. To minimize downtime, upgrade the images on both supervisors and reload the entire chassis using the install command.

8.3.2.1 Prepare the Switch for Dual-Supervisor Switch

To prepare the switch for an EOS upgrade, take the following steps:

- Back up essential files.
- Ensure that you are logged in to the primary supervisor.
- Ensure that the primary supervisor is reachable and that the management interfaces are configured.
- Ensure that there is enough room on both supervisors for the new image file.
- Ensure that any extensions running on the active supervisor are also available on the standby.

Before upgrading the EOS image, ensure that backup copies of the currently running EOS version and the ***running-config*** file are available in case of corruption during the upgrade process. To copy the ***running-config*** file, use the **`copy running-config`** command. In this example, ***running-config*** is copied to a file called ***backup2*** on the flash drive.

```
switch# copy running-config backup2
Copy completed successfully.
switch#
```

Ensure that you are logged in to the primary supervisor, not the standby. Use the show redundancy status command, and verify that ***my state*** reads **`ACTIVE`** and not **`STANDBY`**.

```
switch# show redundancy status
my status = Active
peer state = STANDBY HOT
Unit = Secondary
Unit ID = 1
Redundancy Protocol (Operational) = Stateful Switchover
Redundancy Protocol (Configured) = Stateful Switchover
Communications = Up
Ready for switchover
Last switchover time = 25 days, 19:51:34 ago
Last switchover reason = Other supervisor stopped sending heartbeats
```

Ensure that the switch has a management interface configured with an IP addresses and default gateway. Refer the sections, Assigning a Virtual IP Address to Access the Active Ethernet Management Port and Configuring a Default Route to the Gateway (see Assigning a Virtual IP Address to Access the Active Ethernet Management Port and Configuring a Default Route to the Gateway), and confirm that both management interfaces are in the up state and can ping the default gateway by using the**`show interfaces status`** command and **`ping`** command.

Note: If the management VRF interface is used, use the virtual management interface (management 0) instead of the IP address on the physical management interface.

```
switch# show interfaces status
Port    Name      Status      Vlan       Duplex   Speed      Type
Et3/1             notconnect   1          auto    auto     1000BASE-T

<-------OUTPUT OMITTED FROM EXAMPLE-------->
Ma1/1             connected   routed     unconf   unconf    Unknown switch#ping
 1.1.1.10
PING 172.22.26.1 (172.22.26.1) 72(100) bytes of data.
80 bytes from 1.1.1.10: icmp_seq=1 ttl=64 time=0.180 ms
```

```
80 bytes from 1.1.1.10: icmp_seq=2 ttl=64 time=0.076 ms
80 bytes from 1.1.1.10: icmp_seq=3 ttl=64 time=0.084 ms
80 bytes from 1.1.1.10: icmp_seq=4 ttl=64 time=0.073 ms
80 bytes from 1.1.1.10: icmp_seq=5 ttl=64 time=0.071 ms
```

Determine the size of the new EOS image. Then verify that there is enough space available on the flash drive for two copies of this image (use the **dir** command to check the bytes free figure).

```
switch# dir flash:
Directory of flash:/
-rwx   293168526      Nov 4    22:17   EOS4.11.0.swi
-rwx          36      Nov 8    10:24   boot-config
-rwx       37339      Jun 16   14:18   cfg_06162014
<-------OUTPUT OMITTED FROM EXAMPLE-------->
606638080 bytes total (602841088 bytes free)
Standby supervisor:
switch# dir supervisor-peer:mnt/flash/
Directory of flash:/
-rwx 293168526 Nov 4 22:17 EOS4.11.0.swi
-rwx 36 Nov 8 10:24 boot-config
-rwx 37339 Jun 16 14:18 cfg_06162014
<-------OUTPUT OMITTED FROM EXAMPLE-------->
606638080 bytes total (602841088 bytes free)
```

And, finally, ensure that any extensions running on the primary supervisor are also available on the secondary supervisor.

8.3.2.2 Transfer the Image File to the Primary Supervisor

Load the desired image to the file system on the primary supervisor, typically into the flash. Use the CLI **copy** command to load files to the flash on the primary supervisor, then confirm that the new image file has been correctly transferred.

These command examples transfer an image file to flash from various locations.

USB Memory

Command

copy usb1:/*sourcefile* flash:/*destfile*

Example

```
Sch#copy usb1:/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

FTP Server

Command

copy ftp:/*ftp-source*/*sourcefile* flash:/*destfile*

Example

```
sch# copy ftp:/user:password@10.0.0.3/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

SCP

Command

copy scp://***scp-source***/***sourcefile*** flash:/***destfile***

Example

```
sch# copy scp://user@10.1.1.8/user/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

HTTP

Command

copy http://***http-source***/***sourcefile*** flash:/***destfile***

Example

```
sch# copy http://10.0.0.10/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

Once the file has been transferred, verify that it is present in the directory, then confirm the MD5 checksum using the **`verify`** command. The MD5 checksum for each available image can be found on the EOS download page of the Arista website.

```
switch# dir flash:
Directory of flash:/
-rwx      293168526     Nov 4    22:17     EOS4.11.0.swi
-rwx             36     Nov 8    10:24     boot-config
-rwx          37339     Jun 16   14:18     cfg_06162014
-rwx      394559902     May 30   02:57     EOS-4.12.2.swi

<-------OUTPUT OMITTED FROM EXAMPLE-------->

606638080 bytes total (208281186 bytes free)
switch#53# verify /md5 flash:EOS-4.13.2.swi
verify /md5 (flash:EOS-4.13.2.swi) =c277a965d0ed48534de6647b12a86991
```

8.3.2.3 Install the New EOS Image

Once the EOS image has been copied to the flash drive of the primary supervisor, use the install command to update the ***boot-config***, copy the new image to the secondary supervisor and reload both supervisors. When upgrading to a new image, both supervisors will briefly be unavailable; using the **`install`** command minimizes packet loss during reload.

```
switch(config)# install source EOS-4.13.2.swi reload
Preparing new boot-config... done.
Copying new software image to standby supervisor... done.
Copying new boot-config to standby supervisor... done.
Committing changes on standby supervisor... done.
Reloading standby supervisor... done.
Committing changes on this supervisor... done.
Reloading this supervisor...
```

8.3.2.4 Verify the New Image

After the switch finishes reloading, log into the switch and use the **`show version`** command to confirm the correct image is loaded. The **`Software image version`** line displays the version of the active image file.

```
switch# show version
Arista DCS-7504
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.13.2F
```

```
Architecture:           i386
Internal build version: 4.13.2F-1649184.4132F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

Uptime:                 1 hour and 36 minutes
Total memory:           4017088 kB
Free memory:            1473280 kB switch#
```

8.4 Upgrade/Downgrade Commands

- install
- reload fast-boot
- reload hitless

8.4.1 install

The **`install`** command copies the specified EOS image onto the switch (if the source is external), configures the ***boot-config*** file to point to the specified EOS image, copies the image to the standby supervisor (on dual-supervisor switches), and optionally reloads the switch to run the new EOS.

Command Mode

Privileged EXEC

Command Syntax

`install source` ***source_path*** [destination ***destination_path***][now][reload]

Parameters

- ***source_path*** file path and name of EOS image. If no file path is specified, the switch will look for the image on the flash drive of the primary supervisor.
- **destination** ***destination_path*** destination file path and name of the EOS image. If no destination or name is specified, the EOS image will be stored on the flash drive with its original file name.
- **now** command is executed immediately without further prompts.
- **reload** supervisor is reloaded after the image and updated ***boot-config*** file are installed. On dual-supervisor switches, reloads both supervisors, after which control is returned to the primary supervisor.

Example

This command updates the ***boot-config*** file to point to the **`EOS.swi`** file on the primary supervisors flash drive, copies the image and ***boot-config*** file to the secondary supervisor, and reboots both.

```
switch(config)# install source EOS.swi reload
Preparing new boot-config... done.
Copying new software image to standby supervisor... done.
Copying new boot-config to standby supervisor... done.
Committing changes on standby supervisor... done.
Reloading standby supervisor... done.
Committing changes on this supervisor... done.
Reloading this supervisor...
```

8.4.2 reload fast-boot

Smart System Upgrade (SSU) allows critical switches to be upgraded with minimal downtime and packet loss by optimizing the reload procedure and leveraging protocols capable of graceful restart. The **`reload fast-`**

boot command starts the SSU process using the EOS image specified by the ***boot-config*** file (configured by the **boot system** command).

When the **reload fast-boot** command is entered, the switch sends a message prompting the user to save the configuration if it contains unsaved modifications, then asks the user to confirm the reload request.

Command Mode

Privileged EXEC

Command Syntax

reload fast-boot

Guidelines

- SSU is supported only for upgrades (not downgrades).
- SSU is not supported if the EOS upgrade requires an FPGA upgrade.
- Enough free space must be available on the flash drive to store two copies of the target EOS image. It is also recommended that an additional 240MB be available to store diagnostic information.

Examples

- This command starts the Smart Software Upgrade process.

```
switch# reload fast-boot
Proceed with reload? [confirm]
```

- If there are issues with the current switch configuration that prevents SSU from being performed, the switch lists the changes that must be made before SSU can begin.

```
switch# reload fast-boot
switch#'reload fast-boot' cannot proceed due to the following:
Spanning-tree portfast is not enabled for one or more ports
Spanning-tree BPDU guard is not enabled for one or more ports
 switch#
```

- When the **reload fast-boot** command is entered, the switch sends a message prompting the user to save the configuration if it contains unsaved modifications, then asks the user to confirm the reload request.

```
switch# reload fast-boot
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Copy completed successfully.
Proceed with reload? [confirm]y
```

8.4.3 reload hitless

The **reload hitless** command is a legacy command now identical to the **reload fast-boot** command. It starts the Smart System Upgrade (SSU) process using the EOS image specified by the ***boot-config*** file (configured by the boot system command).

Command Mode

Privileged EXEC

Command Syntax

reload hitless

Guidelines

- SSU is supported only for upgrades (not downgrades).
- SSU is not supported if the EOS upgrade requires an FPGA upgrade.
- Enough free space must be available on the flash drive to store two copies of the target EOS image. It is also recommended that an additional 240MB be available to store diagnostic information.

Examples

- This command starts the SSU process.

```
switch# reload hitless
Proceed with reload? [confirm]
```

- If there are issues with the current switch configuration that prevents SSU from being performed, the switch lists the changes that must be made before SSU can begin.

```
switch# reload hitless
switch#'reload hitless' cannot proceed due to the following:
Spanning-tree portfast is not enabled for one or more ports
Spanning-tree BPDU guard is not enabled for one or more ports
 switch#
```

- When the **reload hitless** command is entered, the switch sends a message prompting the user to save the configuration if it contains unsaved modifications, then asks the user to confirm the reload request.

```
switch# reload hitless
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Copy completed successfully.
Proceed with reload? [confirm]y
```

Chapter 9

Security

The Security chapter contains the following sections:

- User Security
- Control Plane Security
- Data Plane Security

9.1 User Security

This section covers the following:

- AAA Configuration

9.1.1 AAA Configuration

This section describes Authentication, Authorization, and Accounting (AAA), and contains these topics:

- Authentication, Authorization, and Accounting Overview
- Configuring the Security Services
- Server Groups
- Activating Security Services
- Role-Based Authorization
- TACACS+ Configuration Examples
- AAA Commands

9.1.1.1 Authentication, Authorization, and Accounting Overview

This section contains the following topics:

- Methods
- Configuration Statements
- Encryption

9.1.1.1.1 Methods

The switch controls access to EOS commands by authenticating user identity and verifying user authorization. Authentication, Authorization, and Accounting (AAA) activities are conducted through three data services - a local security database, TACACS+ servers, and RADIUS servers. Configuring the Security Services describes these services.

9.1.1.1.2 Configuration Statements

Enabling AAA on the switch requires two steps:

1. Configure security service parameters.

   The switch provides configuration commands for each security service:

   - A local file supports authentication through **username** and enable password commands.
   - TACACS+ servers provide security services through **`tacacs-server`** commands.

- RADIUS servers provide security services through **`radius-server`** commands.
- Configuring the Security Services describes security service configuration commands.

2. Activate AAA services.

   EOS provides **`aaa authorization`**, **`aaa authentication`**, and **`aaa accounting`** commands to select the primary and backup services. Activating Security Services provides information on implementing a security environment.

9.1.1.1.3 Encryption

The switch uses clear-text passwords and server access keys to authenticate users and communicate with security systems. To prevent accidental disclosure of passwords and keys, ***running-config*** stores their corresponding encrypted strings. The encryption method depends on the type of password or key.

Commands that configure passwords or keys can accept the clear-text password or an encrypted string that was generated by the specified encryption algorithm with the clear-text password as the seed.

9.1.1.2 Configuring the Security Services

The switch can access three security data services to authenticate users and authorize switch tasks: a local file, TACACS+ servers, and RADIUS Servers.

This section contains the following topics:

- Local Security File
- TACACS+
- RADIUS
- AAA with LDAP

9.1.1.2.1 Local Security File

The local file uses passwords to provide these authentication services:

- authenticate users as they log into the switch.
- control access to configuration commands.
- control access to the switch root login.

The local file contains username-password combinations to authenticate users. Passwords also authorize access to configuration commands and the switch root login.

9.1.1.2.1.1 Passwords

The switch recognizes passwords as clear text and encrypted strings.

- **Clear-text** passwords are the text that a user enters to access the CLI, configuration commands, or the switch root login.
- **Encrypted strings** are SHA-512-encrypted strings generated with the **clear text** as the seed. The local file stores passwords in this format to avoid unauthorized disclosure. When a user enters the clear-text password, the switch generates the corresponding secure hash and compares it to the stored version.



Note: The switch cannot recover the clear text from which an encrypted string is generated.

Valid passwords contain the characters A-Z, a-z, 0-9 and any of these punctuation characters:

! @ # $ % ^ & * ( ) - _ = + { } [ ] ; : < > , . ? / ~ \

9.1.1.2.1.2 Usernames

Usernames control access to the EOS and all switch commands. The switch is typically accessed through an SSH login, using a previously defined username-password combination. To create a new username or modify an existing username, use the username command.

Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

@ # $ % ^ & * - _ = + ; < > , . ~ |

The default username is **admin**, which is described in Admin Username.

Examples

- These equivalent commands create the username ***john*** and assign it the password ***x245***. The password is entered in clear text because the encrypt-type parameter is omitted or zero.

```
switch(config)# username john secret x245
switch(config)# username john secret 0 x245
```

- This command creates the username ***john*** and assigns it to the text password that corresponds to the encrypted string ***1sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1***. The string was generated by an MD5-encryption program using ***x245*** as the seed.

```
switch(config)# username john secret 5 $1$sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1
```

The username is authenticated by entering ***x245*** when the CLI prompts for a password.

- This command creates the username ***jane*** without securing it with a password. It also removes a password if the ***jane*** username exists.

```
switch(config)# username jane nopassword
```

- This command removes the username ***william*** from the local file.

```
switch(config)# no username william
```

9.1.1.2.1.3 Logins by Unprotected Usernames

The default switch configuration allows usernames that are not password-protected to log in only from the console. The aaa authentication policy local allow-nopassword-remote-login command configures the switch to allow unprotected usernames to log in from any port. To reverse this setting to the default state, use no form of aaa authentication policy local allow-nopassword-remote-login.

Note: Allowing remote access to accounts without passwords is a severe security risk. Arista Networks recommends assigning strong passwords to all usernames.

Examples

- This command configures the switch to allow unprotected usernames to log in from any port.

```
switch(config)# aaa authentication policy local allow-nopassword-remote-lo
gin
```

- This command configures the switch to allow unprotected usernames to log in only from the console port.

```
switch(config)# no aaa authentication policy local allow-nopassword-remote-
login
```

9.1.1.2.1.4 Enable Command Authorization

The **enable** command controls access to Privileged EXEC and all configuration command modes. The enable password authorizes users to execute the **enable** command. When the enable password is set, the CLI displays a password prompt when a user attempts to enter ***Privileged EXEC*** mode.

```
main-host> enable
Password:
main-host#
```

If an incorrect password is entered three times in a row, the CLI displays the EXEC mode prompt.

If no enable password is set, the CLI does not prompt for a password when a user attempts to enter ***Privileged EXEC*** mode.

To set the enable password, use the enable password command.

Examples

- These equivalent commands assign ***xyrt1*** as the enable password.

```
switch(config)# enable password xyrt1
switch(config)# enable password 0 xyrt1
```

- This command assigns the enable password to the clear text ***12345*** corresponding to the encrypted string ***$1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/***. The string was generated by an MD5-encryption program using ***12345*** as the seed.

```
switch(config)# enable password 5 $1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/
```

- This command deletes the enable password.

```
switch(config)# no enable password
```

9.1.1.2.1.5 Root Account Password

The root account accesses the root directory in the underlying Linux shell. When it is not password protected, you can log into the root account only through the console port. After you assign a password to the root account, you can log into it through any port.

To set the password for the root account, use the aaa root command.

Examples

- These equivalent commands assign ***f4980*** as the root account password.

```
switch(config)# aaa root secret f4980
switch(config)# aaa root secret 0 f4980
```

- This command assigns the text ***ab234*** that corresponds to the encrypted string ***1HW05LEY8$QEVw6JqjD9VqDfh.O8r.b***. as the root password.

```
switch(config)# aaa root secret 5 $1$HW05LEY8$QEVw6JqjD9VqDfh.O8r.b
```

- This command removes the password from the root account.

```
switch(config)# aaa root nopassword
```

- This command disables the root login.

```
switch(config)# no aaa root
```

9.1.1.2.2 TACACS+

Terminal Access Controller Access-Control System Plus (TACACS+), derived from the TACACS protocol defined in ***RFC 1492***, is a network protocol that provides centralized user validation services. TACACS+ information is maintained on a remote database. EOS support of TACACS+ services requires access to a TACACS+ server.

TACACS+ manages multiple network access points from a single server. The switch defines a TACACS+ server connection by its address and port, allowing the switch to conduct multiple data streams to a single server by addressing different ports on the server.

These sections describe steps that configure access to TACACS+ servers. Configuring TACACS+ access is most efficiently performed when TACACS+ is functioning prior to configuring switch parameters.

9.1.1.2.2.1 Configuring TACACS+ Parameters

TACACS+ parameters define settings for the switch to communicate with TACACS+ servers. A set of values can be configured for individual TACACS+ servers that the switch accesses. Global parameters define settings for communicating with servers for which parameters are not individually configured.

The switch supports the following TACACS+ parameters.

Encryption Key

The encryption key is code that the switch and the TACACS+ server share to facilitate communications.

- The tacacs-server host command defines the encryption key for a specified server.
- The tacacs-server key command defines the global encryption key.

Examples

- This command configures the switch to communicate with the TACACS+ server assigned the host name ***TAC_1*** using the encryption key ***rp31E2v***.

```
switch(config)# tacacs-server host TAC-1 key rp31E2v
```

- This command configures ***cv90jr1*** as the global encryption key.

```
switch(config)# tacacs-server key 0 cv90jr1
```

- This command assigns ***cv90jr1*** as the global key, using the corresponding encrypted string.

```
switch(config)# tacacs-server key 7 020512025B0C1D70
```

Session Multiplexing

The switch supports multiplexing sessions on a single TCP connection.

- The tacacs-server host command configures the multiplexing option for a specified server.
- There is no global multiplexing setting.

Example

This command configures the switch to communicate with the TACACS+ server at ***10.12.7.9*** and indicates the server supports session multiplexing on a TCP connection.

```
switch(config)# tacacs-server host 10.12.7.9 single-connection
```

Timeout

The timeout is the period the switch waits for a successful connection to, or response from, the TACACS+ server. The default is ***5*** seconds.

- The tacacs-server host command defines the timeout for a specified server.
- The tacacs-server timeout command defines the global timeout.

Examples

- This command configures the switch to communicate with the TACACS+ server assigned the host name ***TAC_1*** and configures the timeout period as ***20*** seconds.

  ```
  switch(config)# tacacs-server host TAC_1 timeout 20
  ```

- This command configures ***40*** seconds as the period that the server waits for a response from a TACACS+ server before issuing an error.

  ```
  switch(config)# tacacs-server timeout 40
  ```

Port

The port specifies the port number through which the switch and the servers send information. The TACACS+ default port is ***49***.

- The tacacs-server host command specifies the port number for an individual TACACS+ server.
- The global TACACS+ port number cannot be changed from the default value of ***49***.

Example

This command configures the switch to communicate with the TACACS+ server at ***10.12.7.9*** through port ***54***.

```
switch(config)# tacacs-server host 10.12.7.9 port 54
```

9.1.1.2.2.2 TACACS+ Status

To display the TACACS+ servers and their interactions with the switch, use the show tacacs command.

Example

This command lists the configured TACACS+ servers.

```
switch(config)# show tacacs

server1: 10.1.1.45
Connection opens: 15
Connection closes: 6
Connection disconnects: 6
Connection failures: 0
Connection timeouts: 2
Messages sent: 45
Messages received: 14
```

```
Receive errors: 2
Receive timeouts: 2
Send timeouts: 3

Last time counters were cleared: 0:07:02 ago
```

To reset the TACACS+ status counters, use the clear aaa counters tacacs+ command.

Example

This command clears all TACACS+ status counters.

```
switch(config)# clear aaa counters tacacs
```

9.1.1.2.3 RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a networking protocol that provides centralized AAA services for computers connecting to and using network resources. RADIUS is used to manage access to the Internet, internal networks, wireless networks, and integrated email services.

These sections describe steps that configure RADIUS server access. Configuring RADIUS parameters is most efficiently performed when RADIUS is functioning prior to configuring switch parameters.

9.1.1.2.3.1 RADIUS Vendor-Specific Attribute-Value Pairs

RADIUS servers and client companies extend basic RADIUS functionality through vendor-specific attributes. A dictionary file includes a list of RADIUS attribute-value pairs that Arista switches use to perform AAA operations through the RADIUS server.

Arista switches use the following attribute values:

- Arista Vendor number: ***30065***
- Attribute: Arista-AVPair 1 string

Acceptable string values for Arista-AVPair include:

- **shell:priv-lvl=<privilege level of a user, 0-15>**
- **shell:roles=<list of roles for a user>**

Example

This is a sample dictionary file that identifies Arista RADIUS vendor-specific attribute value pairs.

```
#
# dictionary.arista
#
VENDOR          Arista    30065
#   Standard Attribute
BEGIN-VENDOR    Arista
ATTRIBUTE       Arista-AVPair    1    string
END-VENDOR      Arista
```

9.1.1.2.3.2 Configuring RADIUS Defaults

RADIUS policies specify settings for the switch to communicate with RADIUS servers. A set of values can be configured for individual RADIUS servers that the switch accesses. Global parameters define settings for communicating with servers for which parameters are not individually configured.

The switch defines the following RADIUS parameters.

Encryption Key

The encryption key is the key shared by the switch and RADIUS servers to facilitate communications.

- The radius-server host command defines the encryption key for a specified server.
- The radius-server key command specifies the global encryption key.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name ***RAD-1*** using the encryption key ***rp31E2v***.

```
switch(config)# radius-server host RAD-1 key rp31E2v
```

- This command configures ***cv90jr1*** as the global encryption key.

```
switch(config)# radius-server key 0 cv90jr1
```

- This command assigns ***cv90jr1*** as the key by specifying the corresponding encrypted string.

```
switch(config)# radius-server key 7 020512025B0C1D70
```

Timeout

The timeout is the period that the switch waits for a successful connection to, or response from, a RADIUS server. The default period is ***5*** seconds.

- The radius-server host command defines the timeout for a specified server.
- The radius-server key command defines the global timeout.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name ***RAD-1*** and configures the timeout period as ***20*** seconds.

```
switch(config)# radius-server host RAD-1 timeout 20
```

- This command configures ***50*** seconds as the period that the server waits for a response from a RADIUS server before issuing an error.

```
switch(config)# radius-server timeout 50
```

Retransmit

Retransmit is the number of times the switch attempts to access the RADIUS server after the first server timeout expiry. The default value is ***3*** times.

- The radius-server host command defines the retransmit for a specified server.
- The radius-server retransmit command defines the global retransmit value.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name ***RAD-1*** and configures the retransmit value as ***2***.

```
switch(config)# radius-server host RAD-1 retransmit 2
```

- This command configures the switch to attempt five RADIUS server contacts after the initial timeout. If the timeout parameter is set to ***50*** seconds, then the total period that the switch waits for a response is ((5+1)*50) = ***300*** seconds.

```
switch(config)# radius-server retransmit 5
```

Deadtime

Deadtime is the period when the switch ignores a non-responsive RADIUS server or a server that does not answer retransmit attempts after timeout expiry. Deadtime is disabled if a value is not specified.

- The radius-server host command defines the deadtime for a specified server.
- The radius-server deadtime command defines the global deadtime setting.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name ***RAD-1*** and configures the deadtime period as ***90*** minutes.

```
switch(config)# radius-server host RAD-1 deadtime 90
```

- This command programs the switch to ignore a server for two hours if the server does not respond to a request during the timeout-retransmit period.

```
switch(config)# radius-server deadtime 120
```

Port

The port specifies the port number through which the switch and servers send information.

- The radius-server host command specifies the port numbers for an individual RADIUS server.
- The global RADIUS port numbers cannot be changed from the default values of ***1812*** for an authorization port and ***1813*** for an accounting port.

Example

These commands configure the switch to communicate with the RADIUS server named ***RAD-1*** through port number ***1850*** for authorization and port number ***1851*** for accounting.

```
switch(config)# radius-server host RAD-1 auth-port 1850
switch(config)# radius-server host RAD-1 acct-port 1851
```

To remove the configuration for this server, use **`no radius-server host`** command and specify the hostname or IP address with both the authorization and accounting port numbers.

9.1.1.2.3.3 DSCP Support for CPU-generated Traffic

The Differentiated Services Code Point (DSCP) is a 6 bit field in the IP header, which marks traffic for providing Quality of Service (QoS). All protocol-specific traffic from the switch is marked with the configured DSCP value set individually for the following network management protocols:

- RADIUS
- TACACS
- SNMP
- SSH
- sFlow

Configuring DSCP Value

The following commands are applicable to all platforms for configuring DSCP value.

Example

This command configures the DSCP value of ***62*** for RADIUS-server.

```
switch(config)# radius-server qos dscp 62
```

This command configures the DSCP value of ***36*** for TACACS-server.

Example

```
switch(config)# tacacs-server qos dscp 36
```

This command configures the DSCP value of ***36*** for snmp-server.

```
switch(config)# snmp-server qos dscp 36
```

Example

This command configures the DSCP value of ***36*** for sFlow.

```
switch(config)# sFlow qos dscp 36
```

This command configures the DSCP value of ***36*** for snmp-server.

```
switch(config)# snmp-server qos dscp 36
```

9.1.1.2.3.4 RADIUS Status

The show radius command displays configured RADIUS servers and their interactions with the switch.

Examples

- This command lists the configured RADIUS servers.

```
switch(config)# show radius

server1: 10.1.1.45
Messages sent: 24
Messages received: 20
Requests accepted: 14
Requests rejected: 8
Requests timeout: 2
Requests retransmitted: 1
Bad responses: 1
Last time counters were cleared: 0:07:02 ago
```

  To reset the RADIUS status counters, use the clear aaa counters radius command.
- This command clears all RADIUS status counters.

```
switch(config)# clear aaa counters radius
```

9.1.1.2.4 AAA with LDAP

The switches support AAA with LDAP protocol for authentication and authorization using TLS communication with a remote LDAP server, and interoperates with Microsoft's ActiveDirectory when configured with LDAP plugins. LDAP authentication configuration is required for LDAP to work. AAA requests to servers are made in the order of their configuration. Once a server is marked as unreachable, it is tried only after all other servers are also found unreachable.

Configuring LDAP Authentication

For all platforms, the **`ldap`** command is configured from the management **ldap** mode and requires configuration files to provide remote authentication.

Active Directory Server with LDAP Plug-in Configured

The file extract below configures the authentication for ***rdn attribute user*** and ***search filter***.

```
aaa authentication login default group ldap local
aaa authorization exec default group ldap local
!
management ldap
   server host ldap-server.samplecompany.com
   !
   server defaults
      base-dn dc=samplecompany,dc=com
      rdn attribute user cn
      ssl-profile testProfile
      authorization group policy basic-role-example
      search username cn=ldap-admin-acct,OU=ServiceAccounts,OU=Sample,dc=s
amplecompany,dc=com password 0 secretString
   !
   group policy basic-role-example
      search filter objectclass group attribute member
      group "Network Admin" role network-admin
      group "Network Newbie" role network-operator
!
management security
   ssl profile testProfile
      fips restrictions
      trust certificate caCert
!
```

The file extract below configures the management **ldap** mode.

```
management ldap
   server host ldap-server.samplecompany.com
      ssl-profile testProfile2
      authorization group policy company1
   !
   Server host ldap-server.company2.com
   !
   server defaults
      base-dn dc=samplecompany,dc=com
      rdn attribute user cn
      ssl-profile testProfile1
      authorization group policy basic-role-example
      search username cn=ldap-admin-acct,OU=ServiceAccounts,OU=Sample,dc=s
amplecompany,dc=com password 0 secretString
   !
   group policy basic-role-example
      search filter objectclass group attribute member
      group "Network Admin" role network-admin
      group "Network Newbie" role network-operator
   !
   group policy company1
      search filter objectclass group attribute member
      group "Network Admin2" role network-admin
      group "Network Newbie2" role network-operator
   !
```

Use LDAP with the following configuration as a minimum.

```
aaa authentication login default group ldap
!
management ldap
   server host <ldap server hostname/ip>
   !
   server defaults
      base-dn <base distinguished name>
      rdn attribute user <relative distinguished attribute name>
      search username <full distinguished name> password <password>
```

The configuration sets up aaa authentication with LDAP. The LDAP server supports IPv4, IPv6, hostnames, and VRFs for specifying the address. The **RDN** , relative distiguished name, is typically an attribute/value pair to specify a user. When a user attempts to connect to the switch, the admin username searches recursively for the **RDN**s which match the passed-in username from the base-dn folder to generate a shortened list of potential **DN**s, which are then searched for a match with the provided password.

Configuring LDAP Authorization

Active Directory Server with LDAP Plug-in Configured

The file extract below configures the authorization for a user.

```
aaa authorization exec default group ldap
!
management ldap
   server defaults
      authorization group policy basic-role-example
   !
   group policy basic-role-example
      search filter objectclass group attribute member
      group "Network Admin" role network-admin
      group "Network Newbie" role network-operator
```

The **group** / **role** maps an LDAP group to an EOS role for RBAC. The matching is done so that the first group that is matched against results in the role being mapped to the user. **`before`** and **`after`** commands are used to insert rules in the appropriate priority.

The LDAP admin account uses the **`search filter`** command to search for LDAP groups which contain the user, where **objectclass** defines the object which contains the LDAP group and **attribute** is the entry attribute name which contains the DN of the group member.

TLS Communication

LDAP supports TLS communication using SSL profiles. A trust certificate, or multiple intermediate certificates, is required to verify the root of trust of the LDAP server. The server will not be used for authentication if **ssl profiles** are configured and the server does not support TLS or fails x509 verification. Other **ssl profiles** supported commands are:

- fips restrictions
- crl
- tls version
- cipher-list

Active Directory Server with LDAP Plug-in Configured

The file extract below configures TLS communication.

```
management ldap
   !
```

```
   server defaults
      ssl-profile testProfile
management security
   ssl profile testProfile
      trust certificate <root of trust>
```

9.1.1.3 Server Groups

A server group is a collection of servers that are associated with a single group name. Subsequent authorization and authentication commands can access all servers in a group by invoking the group name. The switch supports ***TACACS+*** and ***RADIUS*** server groups.

The **`aaa group server`** commands create server groups and place the switch in a ***server-group*** configuration mode to assign servers to the group. Commands that reference an existing group place the switch in a ***server-group*** configuration mode to modify the group.

These commands create named server groups and enter the appropriate command mode for the specified group:

- aaa group server radius
- aaa group server tacacs+

The server (server-group-RADIUS configuration mode) commands add servers to the configuration mode server group. Servers must be previously configured with a radius-server host or tacacs-server host command before they are added to a group.

Examples

- This command creates the TACACS+ server group named ***TAC-GR*** and enters ***server-group*** configuration mode for the new group.

  ```
  switch(config)# aaa group server tacacs+ TAC-GR
  switch(config-sg-tacacs+-TAC-GR)#
  ```

- These commands add two servers to the ***TAC-GR*** server group. To add servers to this group, the switch must be in ***sg-tacacs+-TAC-GR*** configuration mode.

  The CLI remains in ***server-group*** configuration mode after adding the ***TAC-1*** server (port ***49***) and the server located at ***10.1.4.14*** (port ***151***) to the group.

  ```
  switch(config-sg-tacacs+-TAC-GR)# server TAC-1
  switch(config-sg-tacacs+-TAC-GR)# server 10.1.4.14 port 151
  switch(config-sg-tacacs+-TAC-GR)#
  ```

- This command exits ***server-group*** configuration mode.

  ```
  switch(config-sg-tacacs+-TAC-GR)# exit
  switch(config)#
  ```

- This command creates the RADIUS server group named ***RAD-SV1*** and enters s***erver-group*** configuration mode for the new group.

  ```
  switch(config)# aaa group server radius RAD-SV1
  switch(config-sg-radius-RAD-SV1)#
  ```

- These commands add two servers to the ***RAD-SV1*** server group. To add servers to this group, the switch must be in s***g-radius-RAD-SV1*** configuration mode.

The CLI remains in server-group configuration mode after adding the RAC-1 server (authorization port ***1812***, accounting port ***1813***) and the server located at ***10.1.5.14*** (authorization port ***1812***, accounting port ***1850***) to the group.

```
switch(config-sg-radius-RAD-SV1)# server RAC-1
switch(config-sg-radius-RAD-SV1)# server 10.1.5.14 acct-port 1850
switch(config-sg-radius-RAD-SV1)#
```

9.1.1.4 Role-Based Authorization

Role-based authorization is a method of restricting access to CLI commands through the assignment of profiles, called ***roles***, to user accounts. Each role consists of rules that permit or deny access to a set of commands within specified command modes.

All roles are accessible to the local security file through a username parameter and to remote users through ***RADIUS*** or ***TACACS+*** servers. Each role can be applied to multiple user accounts. Only one role may be applied to a user.

9.1.1.4.1 Role Types

The switch defines two types of roles: user-defined and built-in.

- User-defined roles are created and edited through CLI commands.
- Built-in roles are supplied with the switch and are not user-editable.

Built-in roles supplied by the switch are ***network-operator*** and ***network-admin***.

9.1.1.4.2 Role Structure

A role is an ordered list of rules that restricts access to specified commands from users on whom it is applied. Roles consist of deny and permit rules. Each rule references a set of command modes and contains a regular expression that specifies one or more CLI commands. Commands are compared sequentially to the rules within a role until a rule's regular expression matches the command.

- Commands that match a regular expression in a permit rule are executed.
- Commands that match a regular expression in a deny rule are disregarded.
- Commands that do not match a regular expression are evaluated against the next rule in the role.

Upon its entry in the CLI, a command is compared to the first rule of the role. Commands that match the rule are ***executed*** (permit rule) or ***disregarded*** (deny rule). Commands that do not match the rule are compared to the next rule. This process continues until the command either matches a rule or the rule list is exhausted. The switch disregards commands not matching any rule.

9.1.1.4.3 Role Rules

Role rules consist of four components: sequence number, filter type, mode expression, and command expression.

Sequence Number

The sequence number designates a rule’s placement in the role. Sequence numbers range in value from 1 to 256. Rule commands that do not include a sequence number append the rule at the end of the list, deriving its sequence number by adding 10 to the sequence number of the last rule in the list.

Example

These rules have sequence numbers ***10*** and ***20***.

```
10 deny mode exec command reload
```

```
20 deny mode config command (no |default )?router
```

Filter Type

The filter type specifies the disposition of matching commands. Filter types are permit and deny. Commands matching permit rules are executed. Commands matching deny rules are disregarded.

Example

These rules are deny and permit rules, respectively.

```
10 deny mode exec command reload
20 permit mode config command interface
```

Mode Expression

The mode expression specifies the command mode under which the command expression is effective. The mode expression may be a regular expression or a designated keyword. Rules support the following mode expressions:

- **exec** EXEC and Privileged EXEC modes
- **config** Global Configuration Mode
- **config-all** All configuration modes, including Global Configuration Mode
- ***short_name***
- short key name of a command mode (exact match)
- ***long_name*** long key name of a command mode (regular expression match of one or more modes)
- ***no parameter*** all command modes

The prompt command configures the CLI to display a configuration mode's key name:

- %P long key name
- %p short key name

Examples

- These commands use the prompt command to display short key name (**if**) and long key name (**if-Et1**) for ***interface ethernet 1***.

  ```
  switch(config)# prompt switch%p
  switch(config)# interface ethernet 1
  switch(config-if)# exit
  switch(config)# prompt switch%P
  switch(config)# interface ethernet 1
  switch(config-if-Et1)#
  ```

  The command supports the use of regular expressions to reference multiple command modes.
- These regular expressions correspond to the listed command modes:
  - **if-Vlan(1|2)** matches ***interface-VLAN 1*** or ***interface-VLAN 2***.
  - **if** matches all interface modes.
  - **acl-text1** matches ACL configuration mode for ***text1*** ACL.

Command Expression

The command expression is a regular expression that corresponds to one or more CLI commands.

Examples

These regular expressions correspond to the specified commands:

- **`reload`** reload command

- **`(no |default )? router`** commands that enter routing protocol configuration modes
- **`(no |default )?(ip|mac) access-list`** commands that enter ACL configuration modes
- **`(no |default )?(ip|mac) access-group`** commands that bind ACLs to interfaces
- **lacp | spanning-tree** LACP and STP commands
- **.*** all commands

9.1.1.4.4 Creating and Modifying Roles

This section contains the following topics:

- Built-in Role
- Managing Roles
- Modifying Roles

9.1.1.4.4.1 Built-in Role

The switch provides the following two built-in roles:

- **network-operator** Allows all commands in EXEC (Privileged) modes. Commands in all other modes are denied.
- **network-admin** Allows all CLI commands in all modes.

The **network-admin** role is typically assigned to the **admin** user to allow it to run any command.

Built-in roles are not editable.

Example

These **`show users roles`** commands display the contents of the built-in roles.

```
switch(config)# show users roles network-operator
The default role is network-operator
role: network-operator
        10 deny mode exec command bash|\|
        20 permit mode exec command .*
switch(config)# show users roles network-admin
The default role is network-operator
role: network-admin
        10 permit command .*
switch(config)#
```

9.1.1.4.4.2 Managing Roles

Creating and Opening a Role

Roles are created and modified in ***Role*** configuration mode. To create a role, enter the role command with the role's name. The switch enters ***Role*** configuration mode. If the command is followed by the name of an existing role, subsequent commands edit that role.

Example

This command places the switch in ***Role*** configuration mode to create a role named ***sysuser***.

```
switch(config)# role sysuser
switch(config-role-sysuser)#
```

Saving Role Changes

Role configuration mode is a group-change mode; changes are saved by exiting the mode.

Examples

- These commands create a role, then add a deny rule to the role. Because the changes are not yet saved, the role remains empty, as shown by show users roles.

```
switch(config)# role sysuser
switch(config-role-sysuser)# deny mode exec command reload
switch(config-role-sysuser)# show users roles sysuser
The default role is network-operator

switch(config-role-sysuser)#
```

- To save all current changes to the role and exit role configuration mode, type **exit**.

```
switch(config-role-sysuser)# exit
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
switch(config)#
```



Note: After exiting role mode, ***running-config*** must be saved to ***startup-config*** to preserve role changes past system restarts.

Discarding Role Changes

The **abort** command exits ***Role*** configuration mode without saving pending changes.

Example

These commands enter ***Role*** configuration mode to add deny rules, but discard the changes before saving them to the role.

```
switch(config)# role sysuser
switch(config-role-sysuser)# deny mode exec command reload
switch(config-role-sysuser)# abort
switch(config)# show users roles sysuser
The default role is network-operator

switch(config)#
```

9.1.1.4.4.3 Modifying Roles

Adding Rules to a Role

The deny (Role) command adds a deny rule to the configuration mode role. The permit (Role) command adds a permit rule to the configuration mode role.

To append a rule to the end of a role, enter the rule without a sequence number while in Role Configuration Mode. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Example

These commands enter the first three rules into a new role.

```
switch(config)# role sysuser
switch(config-role-sysuser)# deny mode exec command reload
switch(config-role-sysuser)# deny mode config command (no |default )?router
switch(config-role-sysuser)# permit command .*
switch(config-role-sysuser)# exit
```

```
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config command (no |default )?router
        30 permit command .*
switch(config)#
```

Inserting a Rule

To insert a rule into a role, enter the rule with a sequence number between the existing rules numbers.

Example

This command inserts a rule between the first two rules by assigning it the sequence number 15.

```
switch(config)# role sysuser
switch(config-role-sysuser)# 15 deny mode config-all command lacp
switch(config-role-sysuser)# exit
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        15 deny mode config-all command lacp
        20 deny mode config command (no |default )router
        30 permit command .*
switch(config)#
```

Deleting a Rule

To remove a rule from the current role, perform one of these commands:

- Enter **no**, followed by the sequence number of the rule to be deleted.
- Enter **no**, followed by the rule be deleted.
- Enter **default**, followed by the sequence number of the rule to be deleted.
- Enter **default**, followed by the rule to be deleted.

Example

- These equivalent commands remove rule ***30*** from the list.

```
switch(config-role-sysuser)# no 30
switch(config-role-sysuser)# default 30
switch(config-role-sysuser)# no permit command .*

switch(config-role-sysuser)# default permit command .*
```

- This role results from entering one of the preceding commands.

```
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        15 deny mode config-all command lacp|spanning-tree
        20 deny mode config command (no |default )router
switch(config)#
```

Redistributing Sequence Numbers

Sequence numbers determine the order of the rules in a role. After a list editing session where existing rules are deleted and new rules are inserted between existing rules, the sequence number distribution may not be uniform. Redistributing rule numbers changes adjusts the sequence number of rules to provide a constant difference between adjacent rules. The resequence (Role) command adjusts the sequence numbers of role rules.

Example

The **resequence** command renumbers rules in the sysuser role. The sequence number of the first rule is ***100***; subsequent rules numbers are incremented by ***20***.

```
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config-all command lacp|spanning-tree
        25 deny mode config command (no |default )?router
        30 permit command .*
switch(config)# role sysuser
switch(config-role-sysuser)# resequence 100 20
switch(config-role-sysuser)# exit
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        100 deny mode exec command reload
        120 deny mode config-all command lacp|spanning-tree
        140 deny mode config command (no |default )?router
        160 permit command .*
switch(config)#
```

9.1.1.4.5 Assigning a Role to a Username

Roles are assigned to local users through the username command and to remote users through RADIUS servers or TACACS+ servers. Each user is assigned one role. Each role can be assigned to multiple local and remote users.

9.1.1.4.5.1 Default Roles

Users that are not explicitly assigned a role are assigned the default role. The aaa authorization policy local default-role command designates the default role. The ***network-operator*** built-in role is the default role when the default role is not configured.

Examples

- These commands assign ***sysuser*** as the default role, then display the name of the default role.

```
switch(config)# aaa authorization policy local default-role sysuser
switch(config)# show users roles
The default role is sysuser

switch(config)#
```

- These commands restore ***network-operator*** as the default role by deleting the **aaa authorization policy local default-role** statement from ***running-config***, then display the default role name.

```
switch(config)# no aaa authorization policy local default-role
switch(config)# show users roles
```

```
The default role is network-operator

switch(config)#
```

9.1.1.4.5.2 Local Security File (Username Command)

Roles are assigned to users with the username command's **role** parameter. A username whose ***running-config*** **`username`** statement does not include a **role** parameter is assigned the default role.

The **role** parameter function in a command creating a username is different from its function in a command editing an existing name.

Assigning a Role to a New Username

A **`username`** command creating a username explicitly assigns a role to the username by including the **role** parameter; commands without a **role** parameter assigns the default role to the username.

ExampleThese commands create two usernames. The first user is assigned a role; the second user assumes the default role.

```
switch(config)# username FRED secret 0 axced role sysuser1
switch(config)# username JANE nopassword
switch(config)# show running-config
<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
username FRED role sysuser1 secret 5 $1$dhJ6vrPV$PFOvJCX/vcqyIHV.vd.l20
username JANE nopassword
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Editing the Role of an Existing Username

The role of a previously configured username may be edited by a **`username`** command without altering its password. The role assignment of a username is not changed by **`username`** commands that do not include a **role** parameter.

Examples

- These commands assign a role to a previously configured username.

```
switch(config)# username JANE role sysuser2
switch(config)# show running-config

<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
username FRED role sysuser1 secret 5 $1$dhJ6vrPV$PFOvJCX/vcqyIHV.vd.l20
username JANE role sysuser2 nopassword
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

- These commands reverts a username to the default role by removing its role assignment.

```
switch(config)# no username FRED role
switch(config)# show running-config
<-------OUTPUT OMITTED FROM EXAMPLE-------->
!

username FRED secret 5 $1$dhJ6vrPV$PFOvJCX/vcqyIHV.vd.l20
username JANE role sysuser2 nopassword
```

```
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Displaying the Role Assignments

The show users accounts command displays role assignment of the configured users. The show users detail command displays roles of users that are currently logged into the switch.

Examples

- This command displays the configured users and their role assignments.

```
switch(config)# show users accounts
user: FRED
       role: <unknown>
       privilege level: 1
user: JANE
       role: sysuser2
       privilege level: 1
user: admin
       role: network-admin
       privilege level: 1
switch(config)#
```

- This command displays information about the active AAA login sessions.

```
switch(config)# show aaa session
Session  Username  Roles             TTY    State Duration  Auth          Remote Host
-------  --------- ------------      ------ ----- --------  ------------- ------------
2         admin     network-operator ttyS0   E     0:01:21  local
4         Fred      sysadmin         telnet  E     0:02:01  local         sf.example.com
6         Jane      sysuser2         ssh     E     0:00:52  group radius  ny.example.com
9         admin     network-admin    ssh     E     0:00:07  local         bj.example.com
10        max       network-admin    telnet  E     0:00:07  local         sf.example.com
```

9.1.1.4.5.3 Radius Servers

A role can be assigned to a remote user authenticated through a RADIUS server. Roles are assigned through the vendor-specific Attribute-Value (AV) pair named “Arista-AVPair.” The switch extracts the remote user’s role upon a successful authentication when RADIUS authentication is enabled.

Example

This file extract is sample FreeRadius server code that includes the AV pair that assigns roles to three remote users.

```
# Sample RADIUS server users file
"Jane"          Cleartext-Password := "Abc1235"
                Arista-AVPair = "shell:roles=sysuser2",
                Service-Type = NAS-Prompt-User
"Mary"          Cleartext-Password := "xYz$2469"
                Arista-AVPair = "shell:roles=sysadmin",
                Service-Type = NAS-Prompt-User
"Fred"          Cleartext-Password := "rjx4#222"
                Arista-AVPair = "shell:roles=network-operator",
                Service-Type = NAS-Prompt-User
```

The aaa authentication login command selects the user authentication service (see Configuring Service Lists).

Example

This command configures the switch to authenticate users through all RADIUS servers.

```
switch(config)# aaa authentication login default group radius
```

9.1.1.4.5.4 Enable Role-Based Access Control

To enable Role-Based Access Control on the switch, apply the following configuration:

```
switch(config)# aaa authorization commands all default local
```

9.1.1.5 Activating Security Services

After configuring the access databases, **`aaa authentication`**, **`aaa authorization`**, and **`aaa accounting`** commands designate active and backup services for handling access requests.

These sections describe the methods of selecting the database that the switch uses to authenticate users and authorize access to network resources.

9.1.1.5.1 Authenticating Usernames and the Enable Password

Service lists specify the services the switch uses to authenticates usernames and the enable password.

9.1.1.5.1.1 Service List Description

Service list elements are service options, ordered by their priority.

Note: When the local file is one of the service list elements, any attempts to locally authenticate a username that is not included in the local file will result in the switch continuing to the next service list element.

Example

This is an example service list for username authentication:

1. ***Location_1*** server group - specifies a server group (see Server Groups).
2. ***Location_2*** server group - specifies a server group.
3. TACACS+ servers - specifies all hosts for which a **`tacacs-server host`** command exists.
4. Local file - specifies the local file.
5. None - specifies that no authentication is required - all access attempts succeed.

To authenticate a username, the switch checks ***Location_1*** server group. If a server in the group is available, the switch authenticates the username through that group. Otherwise, it continues through the list until it finds an available service or utilizes option 5, which allows the access attempt to succeed without authentication.

9.1.1.5.1.2 Configuring Service Lists

Service lists are incorporated into these **`aaa authentication`** commands to specify services the switch uses to authenticate usernames and the enable password.

- aaa authentication login specifies services the switch uses to authenticate usernames.
- aaa authentication enable specifies services the switch uses to authenticate the enable password.

Examples

- This command configures the switch to authenticate usernames through the ***TAC-1*** server group. The local database is the backup method if ***TAC-1*** servers are unavailable.

  ```
  switch(config)# aaa authentication login default group TAC-1 local
  ```

- This command configures the switch to authenticate usernames through all ***TACACS+*** servers, then all RADIUS servers if the ***TACACS+*** servers are not available. If the RADIUS servers are unavailable, the switch does not authenticate any login attempts.

```
switch(config)# aaa authentication login default group tacacs+ group radius
 none
```

- This command configures the switch to authenticate the enable password through all ***TACACS+*** servers, then through the local database if the ***TACACS+*** servers are unavailable.

```
switch(config)# aaa authentication enable default group TACACS+ local
```

9.1.1.5.2 AAA Time-based Lockout

AAA time-based lockout enables managing remote user unsuccessful login attempts for a configurable time duration.

- aaa authentication policy lockout failure command locks the remote user from getting access for a specific duration of time after specific consecutive unsuccessful login attemps within a lockout period. In the following example, a user is allowed 4 attempts to log in within a duration of 1 day (the default window). If the user has 4 unsuccessful consecutive logins, the person will be locked out of the account for 360 seconds.

```
switch(config)# aaa authentication policy lockout failure 4
duration 360
```

- show aaa authentication lockout command displays the status of locked-out users.

```
switch# show aaa authentication lockout
```

- clear aaa authentication lockout command clears the locked status of a user so as to allow access within a lockout period.

```
switch# clear aaa authentication lockout
```

9.1.1.5.3 Authorization

Authorization commands control EOS shell access, CLI command access, and configuration access through the console port. The switch also supports role-based authorization, which allows access to specified CLI commands by assigning command profiles (or roles) to usernames. See Role-Based Authorization for details.

During the exec authorization process, TACACS+ server responses may include attribute-value (AV) pairs. The switch recognizes the mandatory AV pair named ***priv-lvl=x*** (where **x** is between ***0*** and ***15***).

By default, a TACACS+ server that sends any other mandatory AV pair is denied access to the switch. The receipt of optional AV pairs by the switch has no affect on decisions to permit or deny access to the TACACS + server. The tacacs-server policy command programs the switch to allow access to TACACS+ servers that send unrecognized mandatory AV pairs.

Authorization to switch services is configured by the following **`aaa authorization`** commands.

- To specify the method of authorizing the opening of an EOS shell, enter aaa authorization exec.
- To specify the method of authorizing CLI commands, enter aaa authorization commands.

Examples

- This command specifies that TACACS+ servers authorize users attempting to open a CLI shell.

```
switch(config)# aaa authorization exec default group tacacs+
```

- This command programs the switch to authorize configuration commands (privilege level ***15***) through the local file and to deny command access to users not listed in the local file.

```
switch(config)# aaa authorization commands all default local
```

- This command programs the switch to permit all commands entered on the CLI.

```
switch(config)# aaa authorization commands all default none
```

- This command configures the switch to permit access to TACACS+ servers that send unrecognized mandatory AV pairs.

```
switch(config)# tacacs-server policy unknown-mandatory-attribute ignore
```

All commands are typically authorized through aaa authorization commands. However, the **no aaa authorization config-commands** command disables the authorization of configuration commands. In this state, authorization to execute configuration commands can be managed by controlling access to global configuration commands. The default setting authorizes configuration commands through the policy specified for all other commands.

- To enable the authorization of configuration commands with the policy specified for all other commands, enter aaa authorization config-commands.
- To require authorization of commands entered on the console, enter aaa authorization serial-console.

  By default, EOS does not verify authorization of commands entered on the console port.

Examples

- This command disables the authorization of configuration commands.

```
switch(config)# no aaa authorization config-commands
```

- This command enables the authorization of configuration commands.

```
switch(config)# aaa authorization config-commands
```

- This command configures the switch to authorize commands entered on the console, using the method specified through a previously executed **aaa authorization** command.

```
switch(config)# aaa authorization serial-console
```

9.1.1.5.4 Accounting

The accounting service collects information for billing, auditing, and reporting. The switch supports TACACS + and RADIUS accounting by reporting user activity to either the TACACS+ server or RADIUS server in the form of accounting records.

The switch supports two types of accounting:

- **EXEC**: Provides information about user CLI sessions.
- **Commands**: Command authorization for all commands, including configuration commands that are associated with a privilege level.

The accounting mode determines when accounting notices are sent. Mode options include:

- **start-stop**: a **start** notice is sent when a process begins; a **stop** notice is sent when it ends.
- **stop-only**: a **stop** accounting record is generated after a process successfully completes.

Accounting is enabled by the aaa accounting command.

Examples

- This command configures the switch to maintain start-stop accounting records for all commands executed by switch users and submits them to all TACACS+ hosts.

```
switch(config)# aaa accounting commands all default start-stop group tacacs+
```

- This command configures the switch to maintain stop accounting records for all user EXEC sessions performed through the console and submits them to all TACACS+ hosts.

```
switch(config)# aaa accounting exec console stop group tacacs+
```

9.1.1.6 TACACS+ Configuration Examples

These sections describe two sample TACACS+ host configurations.

9.1.1.6.1 Single Host Configuration

The example single host configuration consists of a TACACS+ server with these attributes:

- IP address: ***10.1.1.10***.
- encryption key: ***example_1***.
- port number: ***49*** (global default).
- timeout: ***5*** seconds (global default).

The switch authenticates the username and enable command against all TACACS+ servers which, in this case, is one host. If the TACACS+ server is unavailable, the switch authenticates with the local file.

1. This step configures TACACS+ server settings port number and timeout are global defaults.

```
switch(config)# tacacs-server host 10.1.1.10 key example_1
```

2. This step configures the login authentication service.

```
switch(config)# aaa authentication login default group tacacs+ local
```

3. This step configures the **enable** command password authentication service.

```
switch(config)# aaa authentication enable default group tacacs+ local
```

9.1.1.6.2 Multiple Host Configuration

The example multiple host configuration consists of three TACACS+ servers at these locations:

- IP address ***10.1.1.2*** - port ***49***.
- IP address ***172.16.4.12*** - port ***4900***.
- IP address ***192.168.2.10*** - port ***49***.

The configuration combines the servers into these server groups:

- ***Bldg_1*** group consists of the servers at ***10.1.1.2*** and ***172.16.4.12***.
- ***Bldg_2*** group consists of the servers at ***192.168.2.10***.

All servers use these global TACACS+ defaults:

- encryption key - ***example_2***.
- timeout - ***10 seconds***.

The switch authenticates these access methods:

- username access against ***Bldg_1*** group then, if they are not available, against the local file.
- enable command against ***Bldg_2*** group, then ***Bldg_1*** group, then against the local file.

1. TACACS+ Host commands:

These commands configure the IP address and ports for the three TACACS+ servers. The port for the first and third server is default ***49***.

```
switch(config)# tacacs-server host 10.1.1.12
switch(config)# tacacs-server host 172.16.4.12 port 4900
switch(config)# tacacs-server host 192.168.2.10
```

2. Global Configuration Commands:

   These commands configure the global encryption key and timeout values.

```
switch(config)# tacacs-server key example_2
switch(config)# tacacs-server timeout 10
```

3. Group Server Commands:

   The **`aaa group server`** commands create the server groups and place the CLI in server group configuration mode, during which the servers are placed in the group. The port number must be included if it is not the default port, as in the line that adds ***192.168.1.1***.

```
switch(config)# aaa group server tacacs+ Bldg_1
switch(config-sg-tacacs+-Bldg_1)# server 10.1.1.2
switch(config-sg-tacacs+-Bldg_1)# server 192.168.1.1 port 4900
switch(config-sg-tacacs+-Bldg_1)# exit
switch(config)# aaa group server tacacs+ Bldg_2
switch(config-sg-tacacs+-Bldg_2)# server 192.168.2.2
switch(config-sg-tacacs+-Bldg_2)# exit
switch(config)#
```

4. Login and enable configuration authentication responsibility commands:

   These commands configure the username and enable command password authentication services.

```
switch(config)# aaa authentication login default group Bldg_1 local
switch(config)# aaa authentication enable default group Bldg_1 group Bldg_2
 local
```

9.1.1.7 AAA Commands

Local Security File Commands

- aaa root
- enable password
- show privilege
- show users
- show users accounts
- username
- username ssh-key

Accounting, Authentication, and Authorization Commands

- aaa accounting
- aaa accounting dot1x
- aaa accounting system
- aaa authentication dot1x
- aaa authentication enable
- aaa authentication login
- aaa authentication policy local allow-nopassword-remote-login

- aaa authentication policy lockout failure
- aaa authentication policy log
- aaa authorization commands
- aaa authorization config-commands
- aaa authorization exec
- aaa authorization policy local default-role
- aaa authorization serial-console
- clear aaa authentication lockout
- clear aaa counters
- clear aaa counters radius
- clear aaa counters tacacs+
- show aaa
- show aaa authentication lockout
- show aaa counters
- show aaa methods
- show management ldap
- show users detail

Server (RADIUS and TACACS+) Configuration Commands

- ip radius source-interface
- ip tacacs source-interface
- radius-server deadtime
- radius-server host
- radius-server key
- radius-server retransmit
- radius-server timeout
- show radius
- show tacacs
- tacacs-server host
- tacacs-server key
- tacacs-server policy
- tacacs-server timeout

Server Group Configuration Commands

- aaa group server radius
- aaa group server tacacs+
- server (server-group-RADIUS configuration mode)
- server (server-group-TACACS+ configuration mode)

Role-Based Authorization Configuration Commands

- deny (Role)
- no <sequence number> (Role)
- permit (Role)
- resequence (Role)
- role
- show users roles

9.1.1.7.1 aaa accounting dot1x

The **aaa accounting dot1x** command enables the accounting of requested 802.1X services for network access.

The **no aaa accounting dot1x** and **default aaa accounting dot1x** commands disable the specified method list by removing the corresponding **aaa accounting dot1x** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

aaa accounting dot1x default [**METHOD_1**][**METHOD_2**][**METHOD_N**]

no aaa accounting dot1x default

default aaa accounting dot1x default

Parameters

- **MODE** accounting mode that defines when accounting notices are sent. Options include:
  - **start-stop** a *start* notice is sent when a process begins; a *stop* notice is sent when it ends.
- **METHOD_X** server groups (methods) to which the switch can send accounting records. The switch sends the method list to the first listed group that is available.
- Parameter value is not specified if **MODE** is set to **none**. If **MODE** is not set to **none**, the command must provide at least one method. Each method is composed of one of the following:
- ***group name*** the server group identified by *name*.
  - ***group radius*** server group that includes all defined RADIUS hosts.
  - ***logging*** server group that includes all defined TACACS+ hosts.

Examples

- This example configures ***IEEE 802.1X*** accounting on the switch.

```
switch(config)# aaa accounting dot1x default start-stop group radius
switch(config)#
```

- This example disables ***IEEE 802.1X*** accounting on the switch.

```
switch(config)# no aaa accounting dot1x default
switch(config)#
```

9.1.1.7.2 aaa accounting system

The **aaa accounting system** command performs accounting for all system-level events.

The **no aaa accounting system** and **default aaa accounting system** commands clear the specified method list by removing the corresponding **aaa accounting system** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

aaa accounting system default [**METHOD_1**][**METHOD_2**] **...** [**METHOD_N**]

no aaa accounting system default

default aaa accounting system default

Parameters

- **MODE** accounting mode that defines when accounting notices are sent. Options include:
  - **none** no notices are sent.
  - **start-stop** a *start* notice is sent when a process begins; a *stop* notice is sent when it ends.
  - **stop-only** a *stop* accounting record is generated after a process successfully completes.
- **METHOD_X** server groups (methods) to which the switch can send accounting records. The switch sends the method list to the first listed group that is available.
- Parameter value is not specified if **MODE** is set to **none**. If **MODE** is not set to **none**, the command must provide at least one method. Each method is composed of one of the following:
  - **group *name*** the server group identified by *name.*
    - **group radius** server group that includes all defined RADIUS hosts.
    - **group tacacs+** server group that includes all defined TACACS+ hosts.
    - **logging** server group that includes all defined TACACS+ hosts.

Examples

- This command configures AAA accounting to not use any accounting methods for system events.

```
switch(config)# aaa accounting system default none
switch(config)#
```

- This command configures the switch to maintain stop accounting records for system events to all defined RADIUS hosts.

```
switch(config)# aaa accounting system default stop-only group radius
switch(config)#
```

9.1.1.7.3 aaa accounting

The **aaa accounting** command configures accounting method lists for a specified authorization type. Each list consists of a prioritized list of methods. The accounting module uses the first available listed method for the authorization type.

The **no aaa accounting** and **default aaa accounting** commands clear the specified method list by removing the corresponding **aaa accounting** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

aaa accounting **TYPE CONNECTION MODE** [**METHOD_1**][**METHOD_2**] ... [**METHOD_N**]

no aaa accounting **TYPE CONNECTION**

default aaa accounting **TYPE CONNECTION**

Parameters

- **TYPE** authorization type for which the command specifies a method list. Options include:
  - **EXEC** records user authentication events.

    - **COMMANDS ALL** records all entered commands.
    - **COMMANDS** *level* records entered commands of the specified *level* (ranges from ***0*** to ***15***).
- **CONNECTION** connection type of sessions for which method lists are reported. Options include:
    - **console** console connection.
    - **default** all connections not covered by other command options.
- **MODE** accounting mode that defines when accounting notices are sent. Options include:
    - **none** no notices are sent.
    - **start-stop** a *start* notice is sent when a process begins; a *stop* notice is sent when it ends.
    - **stop-only** a *stop* accounting record is generated after a process successfully completes.
- **METHOD_X** server groups (methods) to which the switch can send accounting records. The switch sends the method list to the first listed group that is available.
- Parameter value is not specified if **MODE** is set to **none**. If **MODE** is not set to **none**, the command must provide at least one method. Each method is composed of one of the following:
    - ***group name*** the server group identified by *name*.
    - ***group radius*** server group that includes all defined RADIUS hosts.
    - ***group tacacs+*** server group that includes all defined TACACS+ hosts.
    - ***logging*** log all accounting messages to Syslog.

Examples

- This command configures the switch to maintain start-stop accounting records for all commands executed by switch users and submits them to all TACACS+ hosts.

```
switch(config)# aaa accounting commands all default start-stop group
 tacacs+
switch(config)#
```

- This command configures the switch to maintain stop accounting records for all user EXEC sessions performed through the console and submits them to all TACACS+ hosts.

```
switch(config)# aaa accounting exec console stop group tacacs+
switch(config)#
```

- This command configures the switch to maintain start-stop accounting records for all commands executed by switch users and submits them to all TACACS+ hosts.

```
switch(config)# aaa accounting commands all default start-stop group
 tacacs+
switch(config)#
```

- This command configures the switch to maintain stop accounting records for all user EXEC sessions performed through the console and submits them to all TACACS+ hosts.

```
switch(config)# aaa accounting exec console stop group tacacs+
switch(config)#
```

9.1.1.7.4 aaa authentication dot1x

The **aaa authentication dot1x** command configures the default authentication list of requested 802.1X services for network access.

The **no aaa authentication dot1x** and **default aaa authentication dot1x** commands remove the default authentication list for IEEE 802.1X.

Command Mode

Global Configuration

Command Syntax

`aaa authentication dot1x default` group {***group_name*** | ***radius***}

`no aaa authentication dot1x default`

`default aaa authentication dot1x`

Parameters

- **default** configures the default authentication list of requested 802.1X services for network access.
- **group** configures server group.
- ***group_name*** server group name; multiple group names can be entered in a single command.
- **radius** list of all defined RADIUS hosts.

Example

This command configures the switch in the **`auth1`** group for IEEE 802.1X authentication.

```
switch(config)# aaa authentication dot1x default group auth1
switch(config)#
```

9.1.1.7.5 aaa authentication enable

The **`aaa authentication enable`** command configures the service list that the switch references to authorize access to Privileged EXEC command mode.

The list consists of a prioritized list of service options. Available service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authentication
- no authentication

The switch authorizes access by using the first listed service option that is available. When the local file is a service list element, attempts to locally authenticate a username that is not in the local file result in the switch continuing to the next service list element.

When the list is not configured, it is set to **local**.

The **`no aaa authentication enable`** and **`default aaa authentication enable`** commands revert the list configuration as **local** by removing the **`aaa authentication enable`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`aaa authentication enable` ***default*** **METHOD_1** [**METHOD_2**] ... [**METHOD_N**]

`no aaa authentication enable` ***default***

`default aaa authentication enable` ***default***

Parameters

METHOD_X authentication service method list. The command must provide at least one method. Each method is composed of one of the following:

- **group *name*** the server group identified by *name*.
- **group radius** a server group that consists of all defined RADIUS hosts.
- **group tacacs+** a server group that consists of all defined TACACS+ hosts.
- **local** local authentication.
- **none** users are not authenticated; all access attempts succeed.

Example

This command configures the switch to authenticate the enable password through all configured TACACS+ servers. Local authentication is the backup if TACACS+ servers are unavailable.

```
switch(config)# aaa authentication default enable group TACACS+ local
switch(config)#
```

9.1.1.7.6 aaa authentication login

The **`aaa authentication login`** command configures service lists the switch references to authenticate usernames. Service lists consist of service options ordered by usage priority. The switch authenticates usernames through the first available service option. Supported service options include:

- a named server group.
- all defined TACACS+ hosts.
- all defined RADIUS hosts.
- local authentication.
- no authentication.

When the local file is a service list element, attempts to locally authenticate a username that is not in the local file result in the switch continuing to the next service list element.

The switch supports a **console** list for authenticating usernames through the console and a default list for authenticating usernames through all other connections.

- When the **console** list is not configured, the console connection uses the **default** list.
- When the **default** list is not configured, it is set to *local*.

The **`no aaa authentication login`** and **`default aaa authentication login`** commands revert the specified list configuration to its default by removing the corresponding **`aaa authentication`** login command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`aaa authentication login` **CONNECTION SERVICE_1** [**SERVICE_2**] **...** [**SERVICE_N**]

`no aaa authentication login` **CONNECTION**

`default aaa authentication login` **CONNECTION**

Parameters

- **CONNECTION** connection type of sessions for which authentication list is used.
  - **default** the default authentication list.
  - **console** the authentication list for console logins.
- **SERVICE_X** an authentication service. Settings include:

- ***group name*** identifies a previously defined server group.
- ***group radius*** a server group that consists of all defined RADIUS hosts.
- ***group tacacs+*** a server group that consists of all defined TACACS+ hosts.
- ***local*** local authentication.
- ***none*** The switch does not perform authentication. All access attempts succeed.

Examples

- This command configures the switch to authenticate usernames through the ***TAC-1*** server group. The local database is the backup method if ***TAC-1*** servers are unavailable.

```
switch(config)# aaa authentication login default group TAC-1 local
switch(config)#
```

- This command configures the switch to authenticate usernames through all TACACS+ servers, then all RADIUS servers if the TACACS+ servers are not available. If the RADIUS servers are also unavailable, the switch allows access to all login attempts without authentication.

```
switch(config)# aaa authentication login default group tacacs+ group
 radius none
switch(config)#
```

9.1.1.7.7 aaa authentication policy local allow-nopassword-remote-login

The **aaa authentication policy local allow-nopassword-remote-login** command permits usernames without passwords to log in from any port. The default switch setting only allows unprotected usernames to log in from the console.

The **no aaa authentication policy local allow-nopassword-remote-login** and **default aaa authentication policy local allow-nopassword-remote-login** commands return the switch to the default setting of allowing unprotected usernames to log in only from the console.

Command Mode

Global Configuration

Command Syntax

aaa authentication policy local allow-nopassword-remote-login

no aaa authentication policy local allow-nopassword-remote-login

default aaa authentication policy local allow-nopassword-remote-login

Examples

- This command configures the switch to allow unprotected usernames to log in from any port.

```
switch(config)# aaa authentication policy local allow-nopassword-
remote-login
switch(config)#
```

- This command configures the switch to allow unprotected usernames to log in only from the console port.

```
switch(config)# no aaa authentication policy local allow-nopassword-
remote-login
```

```
switch(config)#
```

9.1.1.7.8 aaa authentication policy lockout failure

The **aaa authentication policy lockout failure** command configures the switch to lock the remote user from getting access after specific unsuccessful login attempts within a lockout period.

The **no aaa authentication policy lockout failure** and the **default aaa authentication policy lockout failure** commands disable the lockout period configuration.

Command Mode

Global Configuration

Command Syntax

aaa authentication policy lockout failure ***failure_count*** duration ***duration_time*** {window ***window_time***}

no aaa authentication policy lockout failure

default aaa authentication policy lockout failure

Parameters

- ***failure_count*** the number of failed logins allowed during access. The valid number is between 1 and 255.
- duration ***duration_time*** the time in seconds to block a user account from login. The value is between ***1*** and ***4294967295*** seconds.
- window ***window_time*** the time in seconds to track failed logins within this duration. The value is between ***1*** and ***4294967295*** seconds while the default is 1 day.

Examples

- This command configures the system to allow four attempts to log in within a duration of 1 day (the default window). If the user has 4 unsuccessful consecutive logins, the person will be locked out of the account for 360 seconds.

```
switch(config)# aaa authentication policy lockout failure 4
duration 360
```

- This command configures the system to allow five attempts to log in within a duration of 1 day (the default window). If the user has 5 unsuccessful consecutive logins, the person will be locked out of the account for 60 seconds.

```
switch(config)# aaa authentication policy lockout failure 5
window 10 duration 60
```

9.1.1.7.9 aaa authentication policy log

The **aaa authentication policy log** command configures the switch to generate syslog messages for login authentication success or failure events.

The **no aaa authentication policy log** and the **default aaa authentication policy log** commands restore the default behavior of not generating syslog messages for these events.

Command Mode

Global Configuration

Command Syntax

aaa authentication policy {on-failure | on-success} **log**

no aaa authentication policy {on-failure | on-success} **log**

default aaa authentication policy {on-failure | on-success} **log**

Parameters

- **on-failure** generates syslog messages for failed login events.
- **on-success** generates syslog messages for successful login events.

Example

This command configures the switch to log successful and failed login attempts.

```
switch(config)# aaa authentication policy on-success log
switch(config)# aaa authentication policy on-failure log
```

9.1.1.7.10 aaa authorization commands

The **aaa authorization commands** command configures the service list that authorizes CLI command access. All switch commands are assigned a privilege level that corresponds to the lowest level command mode from which it can be executed:

- **Level 1**: Commands accessible from ***EXEC*** mode.
- **Level 15**: Commands accessible from any mode except ***EXEC***.

Command usage is authorized for each privilege level specified in the command.

The list consists of a prioritized list of service options. The switch authorizes access by using the first listed service option that is available. The available service options include:

- a named server group.
- all defined TACACS+ hosts.
- all defined RADIUS hosts.
- local authorization.
- no authorization.

The list is set to **none** for all unconfigured privilege levels, allowing all CLI access attempts to succeed.

The **no aaa authorization commands** and **default aaa authorization commands** commands revert the list contents to **none** for the specified privilege levels.

Command Mode

Global Configuration

Command Syntax

aaa authorization commands PRIV default SERVICE_1[**SERVICE_2**] **...** [**SERVICE_N**]

no aaa authorization commands PRIV default

default aaa authorization commands PRIV default

Parameters

- **PRIV** Privilege levels of the commands. Options include:
  - ***level*** numbers from ***0*** and ***15***. Number, range, and comma-delimited list of numbers and ranges.
  - **all** commands of all levels.
- **SERVICE_X** Authorization service. Command must list at least one service. Options include:
  - **group *name*** the server group identified by name.
  - **group tacacs+** a server group that consists of all defined TACACS+ hosts.

- **local** local authorization.
- **none** the switch does not perform authorization. All access attempts succeed.

Examples

- This command authorizes configuration commands (privilege level ***15***) through the local file. The switch denies command access to users not listed in the local file.

```
switch(config)# aaa authorization commands all default local
switch(config)#
```

- This command authorizes all commands entered on the CLI.

```
switch(config)# aaa authorization commands all default none
switch(config)#
```

9.1.1.7.11 aaa authorization config-commands

The **`aaa authorization config-commands`** command enables authorization of commands in any configuration mode, such as Global Configuration and all interface configuration modes. Commands are authorized through the policy specified by the aaa authorization commands setting. Authorization is enabled by default, so issuing this command has no effect unless ***running-config*** contains the **`no aaa authorization config-commands`** command.

The **`no aaa authorization config-commands`** command disables configuration command authorization. When configuration command authorization is disabled, ***running-config*** contains the **`no aaa authorization config-commands`** command. The **`default aaa authorization config-commands`** command restores the default setting by removing the **`no aaa authorization config-commands`** from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`aaa authorization config-commands`

`no aaa authorization config-commands`

`default aaa authorization config-commands`

Examples

- This command enables the authorization of configuration commands.

```
switch(config)# aaa authorization config-commands
switch(config)#
```

- This command disables the authorization of configuration commands.

```
switch(config)# no aaa authorization config-commands
switch(config)#
```

9.1.1.7.12 aaa authorization exec

The **aaa authorization exec** command configures the service list that the switch references to authorize access to open an EOS CLI shell.

The list consists of a prioritized list of service options. The switch authorizes access by using the first listed service option to which the switch can connect. When the switch cannot communicate with an entity that provides a specified service option, it attempts to use the next option in the list.

The available service options include:

- a named server group.
- all defined TACACS+ hosts.
- all defined RADIUS hosts.
- local authentication.
- no authentication.

When the list is not configured, it is set to **none**, allowing all CLI access attempts to succeed.

The **no aaa authorization exec** and **default aaa authorization exec** commands set the list contents to **none**.

Command Mode

Global Configuration

Command Syntax

aaa authorization exec default **METHOD_1** [**METHOD_2**] **...** [**METHOD_N**]

no aaa authorization exec default

default aaa authorization exec default

Parameters

- **METHOD_X** authorization service (method). The switch uses the first listed available method.

  The command must provide at least one method. Each method is composed of one of the following:
- **group *name*** the server group identified by name.
  - **group radius** a server group that consists of all defined RADIUS hosts.
    - **group tacacs+** a server group that consists of all defined TACACS+ hosts.
    - **local** local authentication.
    - **none** the switch does not perform authorization. All access attempts succeed.

Guidelines

During the EXEC authorization process, the TACACS+ server response may include attribute-value (AV) pairs. The switch recognizes **priv-lvl=x** (where **x** is an integer between ***0*** and ***15***), which is a mandatory AV pair. A TACACS+ server that sends any other mandatory AV pair is denied access to the switch. The receipt of optional AV pairs by the switch has no affect on decisions to permit or deny access to the TACACS + server.

Example

This command specifies that the TACACS+ servers authorize users that attempt to open an EOS CLI shell.

```
switch(config)# aaa authorization exec default group tacacs+
switch(config)#
```

9.1.1.7.13 aaa authorization policy local default-role

The **aaa authorization policy local** command specifies the name of the default role. A role is a data structure that supports local command authorization through its assignment to user accounts. Roles consist of permit and deny rules that define authorization levels for specified commands. Applying a role to a username authorizes the user to execute commands specified by the role.

The default role is assigned to the following users:

- local or remote users assigned to a role that is not configured.
- local users to whom a role is not assigned.

When the default-role is not specified, **network-operator** is assigned to qualified users as the default role. The network-operator role authorizes assigned users access to all CLI commands in EXEC and Privileged EXEC modes.

The **no aaa authentication policy local default-role** and **default aaa authentication policy local default-role** commands remove the **authentication policy local default-role** statement from ***running-config***. Removing this statement restores **network-operator** as the default role.

Command Mode

Global Configuration

Command Syntax

aaa authorization policy local default-role ***role_name***

no aaa authorization policy local default-role

default aaa authorization policy local default-role

Parameters

role_name Name of the default role.

Related Command

The role command places the switch in ***role*** configuration mode for creating and editing roles.

Examples

- This command configures the sysuser as the default role.

```
switch(config)# aaa authorization policy local default-role sysuser
switch(config)#
```

- This command restores **network-operator** as the default role.

```
switch(config)# no aaa authorization policy local default-role
switch(config)#
```

- This command displays the contents of the **network-operator** role.

```
switch# show users roles network-operator
The default role is network-operator
role: network-operator
        10 deny mode exec command bash|\|
        20 permit mode exec command .*
switch#
```

9.1.1.7.14 aaa authorization serial-console

The **aaa authorization serial-console** command configures the switch to authorize commands entered through the console. By default, commands entered through the console do not require authorization.

The **no aaa authorization serial-console** and **default aaa authorization serial-console** commands restore the default setting.

Command Mode

Global Configuration

Command Syntax

aaa authorization serial-console

no aaa authorization serial-console

default aaa authorization serial-console

Example

This command configures the switch to authorize commands entered on the console, using the method specified through a previously executed aaa authorization commands command.

```
switch(config)# aaa authorization serial-console
switch(config)#
```

9.1.1.7.15 aaa group server radius

The **aaa group server radius** command enters the Server-group-RADIUS Configuration Mode for the specified group name. The command creates the specified group if it was not previously created. Commands are available to add servers to the group.

A server group is a collection of servers that are associated with a single label. Subsequent authorization and authentication commands access all servers in a group by invoking the group name. Server group members must be previously configured with a radius-server host command.

The **no aaa group server radius** and **default aaa group server radius** commands delete the specified server group from ***running-config***.

Command Mode

Global Configuration

Command Syntax

aaa group server radius ***group_name***

no aaa group server radius ***group_name***

default aaa group server radius ***group_name***

Parameters

group_name name (text string) assigned to the group. Cannot be identical to a name already assigned to a TACACS+ server group.

Commands Available in Server-group-RADIUS Configuration Mode

server (server-group-RADIUS configuration mode).

Related Command

aaa group server tacacs+.

Example

This command creates the RADIUS server group named ***RAD-SV1*** and enters ***Server-group-RADIUS*** configuration mode for the new group.

```
switch(config)# aaa group server radius RAD-SV1
switch(config-sg-radius-RAD-SV1)#
```

9.1.1.7.16 aaa group server tacacs+

The **`aaa group server tacacs+`** command enters Server-group-TACACS+ Configuration Mode for the specified group name. The command creates the specified group if it was not previously created. Commands are available to add servers to the group.

A server group is a collection of servers that are associated with a single label. Subsequent authorization and authentication commands access all servers in a group by invoking the group name. Server group members must be previously configured with a tacacs-server host command.

The **`no aaa group server tacacs+`** and **`default aaa group server tacacs+`** commands delete the specified server group from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`aaa group server tacacs+` ***group_name***

`no aaa group server tacacs+` ***group_name***

`default aaa group server tacacs+` ***group_name***

Parameters

group_name name (text string) assigned to the group. Cannot be identical to a name already assigned to a RADIUS server group.

Commands Available in Server-group-TACACS+ Configuration Mode

server (server-group-TACACS+ configuration mode)

Related Command

aaa group server radius

Example

This command creates the TACACS+ server group named TAC-GR and enters the Server-group-TACAS+ Configuration Mode for the new group.

```
switch(config)# aaa group server tacacs+ TAC-GR
switch(config-sg-tacacs+-TAC-GR)#
```

9.1.1.7.17 aaa root

The **`aaa root`** command specifies the password security level for the root account and can assign a password to the account.

The **no aaa root** and **default aaa root** commands disable the root account by removing the **aaa root** command from ***running-config***. The root account is disabled by default.

Command Mode

Global Configuration

Command Syntax

aaa root **SECURITY_LEVEL** [**ENCRYPT_TYPE**] [***password***]

no aaa root

default aaa root

Parameters

- **SECURITY_LEVEL** password assignment level. Settings include:
  - **secret** the root account is assigned to the password.
  - **nopassword** the root account is not password protected.
- **ENCRYPT_TYPE** encryption level of the ***password*** parameter. This parameter is present only when **SECURITY_LEVEL** is **secret**. Settings include:
  - **no parameter** the password is entered as clear text.
  - **0** the password is entered as clear text. Equivalent to **no parameter**.
  - **5** the password is entered as an MD5-encrypted string.
  - **sha512** the password is entered as an SHA-512-encrypted string.
- ***password*** text that authenticates the username. The command includes this parameter only if **SECURITY_LEVEL** is **secret**.
  - ***password*** must be in clear text if **ENCRYPT_TYPE** specifies clear text.
  - ***password*** must be an appropriately encrypted string if **ENCRYPT_TYPE** specifies encryption.

Encrypted strings entered through this parameter are generated elsewhere.

Examples

- These equivalent commands assign ***f4980*** as the root account password.

```
switch(config)# aaa root secret f4980
switch(config)# aaa root secret 0 f4980
```

- This command assigns the text (***ab234***) that corresponds to the encrypted string of ***1HW05LEY8$QEVw6JqjD9VqDfh.O8r.b.*** as the root password.

```
switch(config)# aaa root secret 5 $1$HW05LEY8$QEVw6JqjD9VqDfh.O8r.b
switch(config)#
```

- This command removes the password from the root account.

```
switch(config)# aaa root nopassword
switch(config)#
```

- This command disables the root login.

```
switch(config)# no aaa root
switch(config)#
```

9.1.1.7.18 clear aaa authentication lockout

The **clear aaa authentication lockout** command clears the locked status of a user so as to allow access within a lockout period. If no user is specified, the command clears the locked status of all users.

Command Mode

Privileged EXEC

Command Syntax

clear aaa authentication lockout [user *user_name*]

Parameter

- user ***user_name*** the specific name of the user.

Example

- This command clears the locked status of the user ***Alice***.

```
switch# clear aaa authentication lockout user Alice
```

9.1.1.7.19 clear aaa counters radius

The **clear aaa counters radius** command resets the counters that track the statistics for the RADIUS servers that the switch accesses. The show radius command displays the counters reset by the **clear aaa counters radius** command.

Command Mode

Privileged EXEC

Command Syntax

clear aaa counters radius

Example

These commands display the effect of the **clear aaa counters radius** command on the RADIUS counters.

```
switch# show radius
RADIUS server            : radius/10
          Connection opens:        204
         Connection closes:          0
    Connection disconnects:        199
       Connection failures:         10
       Connection timeouts:          2
             Messages sent:       1490
         Messages received:       1490
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: never
switch# clear aaa counters radius
switch# show radius
RADIUS server            : radius/10
          Connection opens:          0
         Connection closes:          0
    Connection disconnects:          0
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:          0
         Messages received:          0
```

```
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: 0:00:03 ago switch#
```

9.1.1.7.20 clear aaa counters tacacs+

The **clear aaa counters tacacs+** command resets the counters that track the statistics for the TACACS+ servers that the switch accesses. The show tacacs command displays the counters reset by the **clear aaa counters tacacs+** command.

Command Mode

Privileged EXEC

Command Syntax

clear aaa counters tacacs+

Example

These commands display the effect of the **clear aaa counters tacacs+** command on the tacacs+ counters.

```
switch# show tacacs
TACACS+ server            : tacacs/49
          Connection opens:      15942
         Connection closes:          7
    Connection disconnects:       1362
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:      34395
         Messages received:      34392
            Receive errors:          0
           Receive timeouts:          2
             Send timeouts:          0

Last time counters were cleared: never

TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address
 associated with the
Loopback0 interface
switch# clear aaa counters tacacs+
switch# show tacacs
TACACS+ server            : tacacs/49
          Connection opens:          0
         Connection closes:          0
    Connection disconnects:          0
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:          0
         Messages received:          0
            Receive errors:          0
           Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: 0:00:03 ago
switch#
```

```
TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address
 associated with the
Loopback0 interface switch#
```

9.1.1.7.21 clear aaa counters

The **clear aaa counters** command resets the counters that track the number of service transactions performed by the switch since the last time the counters were reset. The show aaa counters command displays the counters reset by the **`clear aaa counters`** command.

Command Mode

Privileged EXEC

Command Syntax

`clear aaa counters` [**SERVICE_TYPE**]

Example

These commands display the effect of the **`clear aaa counters`** command on the AAA counters.

```
switch# clear aaa counters
switch# show aaa counters
Authentication
          Successful:          0
              Failed:          0
 Service unavailable:          0

Authorization
             Allowed:          1
              Denied:          0
 Service unavailable:          0

Accounting
          Successful:          0
               Error:          0
             Pending:          0

Last time counters were cleared: 0:00:44 ago
```

9.1.1.7.22 deny (Role)

The **`deny`** command adds a deny rule to the configuration mode role. Deny rules prohibit access of specified commands from usernames to which the role is applied. Sequence numbers determine rule placement in the role. Commands are compared sequentially to rules within a role until it matches a rule. A commands authorization is determined by the first rule it matches. Sequence numbers for commands without numbers are derived by adding 10 to the number of the roles last rule.

Deny rules use regular expressions to denote commands. A **mode** parameter specifies command modes from which commands are restricted. Modes are denoted either by predefined keywords, a command modes short key, or a regular expression that specifies the long key of one or more command modes.

The **`no deny`** and **`default deny`** commands remove the specified rule from the configuration mode role. The no <sequence number> (Role) command also removes the specified rule from the role.

Command Mode

Role Configuration

Command Syntax

[**SEQ_NUM**] `deny` [**MODE_NAME**] command ***command_name***

`no deny` [**MODE_NAME**] command ***command_name***

`default deny` [**MODE_NAME**] command ***command_name***

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding 10 to the number of the roles last rule.
  - ***1 - 256*** Number assigned to entry.
- **MODE_NAME** Command mode from which command access is prohibited. Values include:
  - ***no parameter*** All command modes.
  - **mode *short_name*** Exact match of a modes short key name.
  - **mode *long_name*** Regular expression matching long key name of one or more modes.
  - **mode config *Global*** configuration mode.
  - **mode config-all** All configuration modes, including ***global*** configuration mode.
  - **mode exec *EXEC*** and ***Privileged EXEC*** modes.
- ***command_name*** Regular expression that denotes the name of one or more commands.

Guidelines

These CLI prompt format commands program the prompt to display the following mode keys:

- **%p** Short mode key.
- **%P** Long mode key.

`Deny` statements are saved to ***running-config*** only upon exiting **Role** configuration mode.

Related Command

The role command places the switch in **Role** configuration mode.

Example

These commands append a **`deny`** rule at the end of the ***sysuser*** role that restricts access to the **`reload`** command from ***EXEC*** and ***Privileged EXEC*** mode.

```
switch(config)# role sysuser
switch(config-mode-sysuser)# deny mode exec command reload
switch(config-mode-sysuser)#
```

9.1.1.7.23 enable password

The **`enable password`** command creates a new enable password or changes an existing password.

The **`no enable password`** and **`default enable password`** commands delete the **`enable password`** by removing the **`enable password`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`enable password` [**ENCRYPT_TYPE**] ***password***

no enable password

default enable password

Parameters

- **ENCRYPT_TYPE** encryption level of the ***password*** parameter. Settings include:
  - ***no parameter*** the password is entered as clear text.
  - ***0*** the password is entered as clear text. Equivalent to <no parameter>.
  - ***5*** the password is entered as an MD5 encrypted string.
  - ***sha512*** the password is entered as an SHA-512-encrypted string.
- ***password*** text that authenticates the username.
  - ***password*** must be in clear text if **ENCRYPT_TYPE** specifies clear text.
  - ***password*** must be an appropriately encrypted string if **ENCRYPT_TYPE** specifies encryption.

Encrypted strings entered through this parameter are generated elsewhere.

Examples

- These equivalent commands assign ***xyrt1*** as the enable password.

```
switch(config)#enable password xyrt1
switch(config)#enable password 0 xyrt1
```

- This command assigns the enable password to the clear text (***12345***) that corresponds to the encrypted string ***$1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/***. The string was generated by an MD5-encryption program using ***12345*** as the seed.

```
switch(config)# enable password 5 $1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/
switch(config)#
```

- This command deletes the enable password.

```
switch(config)# no enable password
switch(config)#
```

9.1.1.7.24 ip radius source-interface

The **ip radius source-interface** command specifies the interface from which the IPv4 address is derived for use as the source for outbound RADIUS packets. When a source interface is not specified, the switch selects an interface.

The **no ip radius source-interface** and **default ip radius source-interface** commands remove the **ip radius source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip radius [**VRF_INST**] **source-interface** **INT_NAME**

no ip radius [**VRF_INST**] **source-interface**

default ip radius [**VRF_INST**] **source-interface**

Parameters

- **VRF_INST** specifies the VRF instance used to communicate with the specified server.

  - ***no parameter*** switch communicates with the server using the default VRF.
  - **vrf *vrf_name*** switch communicates with the server using the specified user-defined VRF.
- **INT_NAME** Interface type and number. Options include:
  - **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **interface loopback *l_num*** Loopback interface specified by ***l_num***.
  - **interface management *m_num*** Management interface specified by ***m_num***.
  - **interface port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **interface vlan *v_num*** VLAN interface specified by ***v_num***.

Example

This command configures the source address for outbound RADIUS packets as the IPv4 address assigned to the loopback interface.

```
switch(config)# ip radius source-interface loopback 0
switch(config)#
```

9.1.1.7.25 ip tacacs source-interface

The **ip tacacs source-interface** command specifies the interface from which the IPv4 address is derived for use as the source for outbound TACACS+ packets. When a source interface is not specified, the switch selects an interface.

The **no ip tacacs source-interface** and **default ip tacacs source-interface** commands remove the **ip tacacs source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip tacacs [**VRF_INST**] **source-interface** **INT_NAME**

no ip tacacs [**VRF_INST**] **source-interface**

default ip tacacs [**VRF_INST**] **source-interface**

Parameters

- **VRF_INST** specifies the VRF instance used to communicate with the specified server.
  - ***no parameter*** switch communicates with the server using the default VRF.
  - **vrf *vrf_name*** switch communicates with the server using the specified user-defined VRF.
- **INT_NAME** Interface type and number. Options include:
  - **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **interface loopback *l_num*** Loopback interface specified by ***l_num***.
  - **interface management *m_num*** Management interface specified by ***m_num***.
  - **interface port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **interface vlan *v_num*** VLAN interface specified by ***v_num***.

Example

This command configures the source address for outbound TACACS+ packets as the IPv4 address assigned to the loopback interface.

```
switch(config)# ip tacacs source-interface loopback 0
switch(config)#
```

9.1.1.7.26 no <sequence number> (Role)

The **no <sequence number>** command removes the rule with the specified sequence number from the configuration-mode role. The **default <sequence number>** command also removes the specified rule.

Command Mode

Role Configuration

Command Syntax

no ***sequence_num***

default ***sequence_num***

Parameters

sequence_num sequence number of rule to be deleted. Values range from ***1*** to ***256***.

Guidelines

Role statement changes are saved to ***running-config*** only upon exiting ***Role*** configuration mode.

Related Command

The role command places the switch in ***Role*** configuration mode.

Example

These commands display the rules in the ***sysuser*** role, remove rule ***30*** from the role, then display the edited role.

```
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config command (no |default )?router
        30 deny mode config command (no |default )?(ip|mac) access-
list
        40 deny mode if command (no |default )?(ip|mac) access-group
        50 deny mode config-all command lacp|spanning-tree
        60 permit command .*
switch(config)# role sysuser
switch(config-role-sysuser)# no 30
switch(config-role-sysuser)# exit
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config command (no |default )?router
        40 deny mode if command (no |default )?(ip|mac) access-group
        50 deny mode config-all command lacp|spanning-tree
        60 permit command .*
```

```
switch(config)#
```

9.1.1.7.27 radius-server deadtime

The **radius-server deadtime** command defines global deadtime period, when the switch ignores a non-responsive RADIUS server. A non-responsive server is one that fails to answer any attempt to retransmit after a timeout expiry. Deadtime is disabled if a value is not configured.

The **no radius-server deadtime** and **default radius-server deadtime** commands restore the default global deadtime period of three minutes by removing the **radius-server deadtime** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

radius-server deadtime ***dead_interval***

no radius-server deadtime

default radius-server deadtime

Parameters

dead_interval period that the switch ignores non-responsive servers (minutes). Values range from ***1*** to ***1000***. Default is ***3***.

Example

This command programs the switch to ignore a server for two hours if it fails to respond to a request during the period defined by timeout and retransmit parameters.

```
switch(config)# radius-server deadtime 120
switch(config)#
```

9.1.1.7.28 radius-server host

The **radius-server host** command sets parameters for communicating with a specific RADIUS server. These values override global settings when the switch communicates with the specified server.

A RADIUS server is defined by its server address, authorization port, and accounting port. Servers with different address-authorization port-accounting port combinations have separate configurations.

The **no radius-server host** and **default radius-server** commands remove settings for the RADIUS server configuration at the specified address-authorization port-accounting port location by deleting the corresponding **radius-server host** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

radius-server host **ADDR** [**VRF_INST**][**AUTH**][**ACCT**][**TIMEOUT**][**DEAD**][**RETRAN**][**ENCRYPT**]

no radius-server host [**ADDR**][**VRF_INST**][**AUTH**][**ACCT**]

default radius-server host [**ADDR**][**VRF_INST**][**AUTH**][**ACCT**]

Parameters

- **ADDR** RADIUS server location. Options include:
  - ***ipv4_addr*** server's IPv4 address.
  - ***host_name*** server's DNS host name (FQDN).
- **VRF_INST** specifies the VRF instance used to communicate with the specified server.
  - ***no parameter*** switch communicates with the server using the default VRF.
  - **vrf *vrf_name*** switch communicates with the server using the specified user-defined VRF.
- **AUTH** Authorization port number.
  - ***no parameter*** default port of ***1812***.
  - **auth-port *number*** number ranges from ***1*** to ***65535***.
- **ACCT** Accounting port number.
  - ***no parameter*** default port of ***1813***.
  - **acct-port *number*** numbers range from ***1*** to ***65535***.
- **TIMEOUT** timeout period (seconds). Ranges from ***1*** to ***1000***.
  - ***no parameter*** assigns global timeout value (see radius-server timeout).
  - **timeout *number*** assigns number as the timeout period. Ranges from ***1*** to ***1000***.
- **DEAD** period (minutes) when the switch ignores a non-responsive RADIUS server.
  - ***no parameter*** assigns global deadtime value (see radius-server deadtime).
  - **deadtime *number*** specifies deadtime, where number ranges from ***1*** to ***1000***.
- **RETRAN** attempts to access RADIUS server after the first timeout expiry.
  - ***no parameter*** assigns global retransmit value (see radius-server retransmit).
  - **retransmit *number*** specifies number of attempts, where number ranges from ***1*** to ***100***.
- **ENCRYPT** encryption key that switch and server use to communicate.
  - ***no parameter*** assigns global encryption key (see radius-server key).
  - **key *key_text*** where ***key_text*** is in clear text.
  - **key 5 *key_text*** where ***key_text*** is in clear text.
  - **key 7 *key_text*** where ***key_text*** is provide in an encrypted string.

Examples

- This command configures the switch to communicate with the RADIUS server located at ***10.1.1.5***. The switch uses the global timeout, deadtime, retransmit, and key settings to communicate with this server, and communicates through port ***1812*** for authorization and ***1813*** for accounting.

```
switch(config)# radius-server host 10.1.1.5
switch(config)#
```

- This command configures the switch to communicate with the RADIUS server assigned the host name ***RAD-1***. Communication for authorization is through port ***1850***; communication for accounting is through port ***1813*** (the default).

```
switch(config)# radius-server host RAD-1 auth-port 1850
switch(config)#
```

9.1.1.7.29 radius-server key

The **`radius-server key`** command defines the global encryption key the switch uses when communicating with any RADIUS server for which a key is not defined.

The **no radius-server key** and **default radius-server key** commands remove the global key from ***running-config***.

Command Mode

Global Configuration

Command Syntax

radius-server key [**ENCRYPT_TYPE**] ***encrypt_key***

no radius-server key

default radius-server key

Parameters

- **ENCRYPT_TYPE** encryption level of *encrypt_key*.
  - ***no parameter*** encryption key is entered as clear text.
  - ***0*** encryption key is entered as clear text. Equivalent to ***no parameter***.
  - ***7 encrypt_key*** is an encrypted string.
- ***encrypt_key*** shared key that authenticates the username.
  - ***encrypt_key*** must be in clear text if **ENCRYPT_TYPE** specifies clear text.
  - ***encrypt_key*** must be an encrypted string if **ENCRYPT_TYPE** specifies an encrypted string.

Encrypted strings entered through this parameter are generated elsewhere.

Related Command

radius-server host

Examples

- This command configures ***cv90jr1*** as the global encryption key.

```
switch(config)# radius-server key 0 cv90jr1
switch(config)#
```

- This command assigns ***cv90jr1*** as the key by specifying the corresponding encrypted string.

```
switch(config)# radius-server key 7 020512025B0C1D70
switch(config)#
```

9.1.1.7.30 radius-server retransmit

The **radius-server retransmit** command defines the global retransmit count, which specifies the number of times the switch attempts to access the RADIUS server after the first timeout expiry.

The **no radius-server retransmit** and **default radius-server retransmit** commands restore the global retransmit count to its default value of three by deleting the **radius-server retransmit** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

radius-server retransmit ***count***

no radius-server retransmit

default radius-server retransmit

Parameters

count retransmit attempts after first timeout expiry. Values range from ***1*** to ***100***. Default is ***3***.

Related Command

radius-server host

Example

This command configures the switch to attempt five RADIUS server contacts after the initial timeout. If the timeout parameter is set to ***50*** seconds, then the total period that the switch waits for a response is ((5+1)*50) = ***300*** seconds.

```
switch(config)# radius-server retransmit 5
switch(config)#
```

9.1.1.7.31 radius-server timeout

The **`radius-server timeout`** command defines the global timeout the switch uses when communicating with any RADIUS server for which a timeout is not defined.

The **`no radius-server timeout`** and **`default radius-server timeout`** commands restore the global timeout default period of five seconds by removing the **`radius-server timeout`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`radius-server timeout` ***time_period***

`no radius-server timeout`

`default radius-server timeout`

Parameters

time_period timeout period (seconds). Values range from ***1*** to ***1000***. Default is ***5***.

Related Commands

- radius-server host
- radius-server key
- radius-server deadtime
- radius-server retransmit

Example

This command configures the switch to wait ***50*** seconds for a RADIUS server response before issuing an error.

```
switch(config)# radius-server timeout 50
switch(config)#
```

9.1.1.7.32 resequence (Role)

The **resequence** command assigns sequence numbers to rules in the configuration mode role. Command parameters specify the number of the first rule and the numeric interval between consecutive rules.

The maximum sequence number is 256.

Command Mode

Role Configuration

Command Syntax

resequence ***start_num inc_num***

Parameters

- ***start_num*** sequence number assigned to the first rule. Value ranges from ***1*** to ***256***. Default is ***10***.
- ***inc_num*** numeric interval between consecutive rules. Value ranges from ***1*** to ***256***. Default is ***10***.

Guidelines

Role statement changes are saved to ***running-config*** only upon exiting ***Role*** configuration mode.

Related Command

The role command places the switch in ***Role*** configuration mode.

Example

The **resequence** command renumbers the rules in the ***sysuser*** role, starting the first rule at ***15*** and incrementing subsequent lines by ***5***.

```
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config command (no |default )?router
        40 deny mode if command (no |default )?(ip|mac) access-group
        50 deny mode config-all command lacp|spanning-tree
        60 permit command .*
switch(config)# role sysuser
switch(config-role-sysuser)# resequence 15 5
switch(config-role-sysuser)# exit
switch(config)# show users roles sysuser
The default role is network-operator

role: sysuser
        15 deny mode exec command reload
        20 deny mode config command (no |default )?router
        25 deny mode if command (no |default )?(ip|mac) access-group
        30 deny mode config-all command lacp|spanning-tree
        35 permit command .*
switch(config)#
```

9.1.1.7.33 permit (Role)

The **permit** command adds a permit rule to the configuration mode role. Permit rules authorize access to specified commands for usernames to which the role is applied. Sequence numbers determine rule placement in the role. Commands are compared sequentially to rules within a role until it matches a rule. A

command's authorization is determined by the first rule it matches. Sequence numbers for commands without numbers are derived by adding 10 to the number of the role's last rule.

Permit rules use regular expression to denote commands. A **mode** parameter specifies command modes in which commands are authorized. Modes are denoted either by predefined keywords, a command modes short key, or a regular expression that specifies the long key of one or more command modes.

The **`no deny`** and **`default deny`** commands remove the specified rule from the configuration mode role. The no <sequence number> (Role) command also removes the specified rule from the role.

Command Mode

Role Configuration

Command Syntax

[**SEQ_NUM**] **`permit`** [**MODE_NAME**] command ***command_name***

`no permit` [**MODE_NAME**] command] ***`command_name`***

`default permit` [**MODE_NAME**] command ***command_name***

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***<no parameter>*** Number is derived by adding 10 to the number of the roles last rule.
  - ***<1 - 256>*** Number assigned to entry.
- **MODE_NAME** Command mode in which command access is authorized. Values include:
  - ***no parameter*** All command modes.
  - **mode *short_name*** Exact match of a modes short-key name.
  - **mode *long_name*** Regular expression matching long-key name of one or more modes.
  - **mode config *Global*** configuration mode.
  - **mode config-all** All configuration modes, including ***global*** configuration mode.
  - **mode exec** EXEC and Privileged EXEC modes.
- ***command_name*** Regular expression that denotes the name of one or more commands.

Guidelines

These CLI prompt format commands program the prompt to display the following mode keys:

- **%p** Short-mode key.
- **%P** Long-mode key.

`Permit` statements are saved to ***running-config*** only upon exiting **Role** configuration mode.

Related Commands

The role command places the switch in **Role** Cconfiguration mode.

Example

These commands append a **`permit`** rule at the end of the sysuser role that authorizes all commands from ***VLAN 1*** or ***VLAN 2*** interface configuration modes.

```
switch(config)# role sysuser
switch(config-mode-sysuser)# permit mode if-Vl(1|2) command .*
switch(config-mode-sysuser)#
```

9.1.1.7.34 role

The **role** command places the switch in Role Configuration Mode, which is a group-change mode that modifies a role. A role is a data structure that supports local command authorization through its assignment to user accounts. Roles consist of permit and deny rules that define authorization levels for specified commands. Applying a role to a username authorizes the user to execute commands specified by the role.

The **role** command specifies the name of the role that subsequent commands modify and creates a role if it references a nonexistent role. All changes in a group change mode edit session are pending until the session ends:

- The **exit** command saves pending changes to ***running-config*** and returns the switch to Global Configuration Mode. Changes are also saved by entering a different configuration mode.
- The **abort** command discards pending changes, returning the switch to Global Configuration Mode.

The **no role** and **default role** commands delete the specified role by removing the role and its statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

role ***role_name***

no role ***role_name***

default role ***role_name***

Parameters

role_name Name of role.

Commands Available in Role Configuration Mode:

- deny (Role)
- permit (Role)
- no <sequence number> (Role)
- resequence (Role)

Related Commands

show users roles

Examples

- This command places the switch in ***Role*** configuration mode to modify the speaker role.

```
switch(config)# role speaker
switch(config-role-speaker)#
```

- This command saves changes to **speaker** role, then returns the switch to ***Global*** configuration mode.

```
switch(config-role-speaker)# exit
switch(config)#
```

- This command discards changes to **speaker**, then returns the switch to ***Global*** configuration mode.

```
switch(config-role-speaker)# abort
```

```
switch(config)#
```

9.1.1.7.35 server (server-group-RADIUS configuration mode)

The **`server (server-group-RADIUS configuration mode)`** command adds the specified RADIUS server to the configuration-mode group. Servers must be configured with the radius-server host command before adding them to the server group.

A RADIUS server is defined by its server address, authorization port, and accounting port. A group can contain multiple servers with the same IP address that have different authorization or accounting ports.

The **`no server`** and **`default server`** commands remove the specified server from the group.

Command Mode

Server-Group-RADIUS Configuration

Command Syntax

`server` **LOCATION** [**VRF_INST**][**AUTH**][**ACCT**]

`no server` **LOCATION** [**VRF_INST**][**AUTH**][**ACCT**]

`default server` **LOCATION** [**VRF_INST**][**AUTH**][**ACCT**]

Parameters

- **LOCATION** RADIUS server location. Options include:
  - ***ipv4_addr*** server's IPv4 address.
  - **host_name** server's DNS host name (FQDN).
- **VRF_INST** specifies the VRF instance used to communicate with the specified server.
  - ***no parameter*** switch communicates with the server using the default VRF.
  - **vrf *vrf_name*** switch communicates with the server using the specified user-defined VRF.
- **AUTH** Authorization port number.
  - ***no parameter*** default port of ***1812***.
  - **auth-port *number*** *number* ranges from ***1*** to ***65535***.
- **ACCT** Accounting port number.
  - ***no parameter*** default port of ***1813***.
  - **acct-port *number*** *number* ranges from ***1*** to ***65535***.

Related Commands

The aaa group server radius command places the switch in ***Server-group-RADIUS*** cConfiguration mode.

Example

These commands add two servers to the ***RAD-SV1*** server group.

```
switch(config)# aaa group server radius RAD-SV1
switch(config-sg-radius-RAD-SV1)# server RAC-1
switch(config-sg-radius-RAD-SV1)# server 10.1.5.14 acct-port 1851
switch(config-sg-radius-RAD-SV1)#
```

9.1.1.7.36 server (server-group-TACACS+ configuration mode)

The **`server (server-group-TACACS+ configuration mode)`** command adds the specified TACACS + server to the configuration-mode group. Servers must be configured with the tacacs-server host command before adding them to the server group.

A TACACS+ server is defined by its server address and port number. Servers with different address-port combinations have separate statements in ***running-config***.

The **`no server`** and **`default server`** commands remove the specified server from the group.

Command Mode

Server-group-TACACS+ Configuration

Command Syntax

`server` **LOCATION** [**VRF_INST**][**PORT**]

`no server` **LOCATION** [**VRF_INST**][**PORT**]

`default server` **LOCATION** [**VRF_INST**][**PORT**]

Parameters

- **LOCATION** TACACS+ server location. Options include:
  - ***ipv4_addr*** server's IPv4 address.
  - ***ipv6_addr*** server's IPv6 address.
  - **host_name** server's DNS host name (FQDN).
- **VRF_INST** specifies the VRF instance used to communicate with the specified server.
  - ***no parameter*** switch communicates with the server using the default VRF.
  - **vrf *vrf_name*** switch communicates with the server using the specified user-defined VRF.
- **PORT** TCP connection port number.
  - ***no parameter*** default port of ***49***.
  - **port** *number* ***number*** ranges from ***1*** to ***65535***.

Related Command

The aaa group server tacacs+ command places the switch in ***Server-group-TACACS+*** configuration mode.

Example

These commands add two servers to the TAC-GR server group with default port number ***49***.

```
switch(config)# aaa group server tacacs+ TAC-GR
switch(config-sg-tacacs+-TAC-GR)# server TAC-1
switch(config-sg-tacacs+-TAC-GR)# server 10.1.4.14
switch(config-sg-tacacs+-TAC-GR)#
```

9.1.1.7.37 show aaa

The **`show aaa`** command displays the user database. The command displays the encrypted enable password first, followed by a table of usernames and their corresponding encrypted password.

The command does not display unencrypted passwords.

Command Mode

Privileged EXEC

Command Syntax

show aaa

Example

This command displays the local user database.

```
switch# show aaa
Enable password (encrypted): $1$UL4gDWy6$3KqCPYPGRvxDxUq3qA/Hs/
Username  Encrypted passwd
--------  ----------------------------------
admin
janis     $1$VVnDH/Ea$iwsfnrGNO8nbDsf0tazp9/
thomas    $1$/MmXTUil$.fJxLfcumzppNSEDVDWq9.
switch#
```

9.1.1.7.38 show aaa authentication lockout

The **show aaa authentication lockout** command displays the status of locked-out users who could not log within the specified time and number of login attempts.

Command Mode

Privileged EXEC

Command Syntax

show aaa authentication lockout

Example

- This command displays the status of Alice, who is the locked out user. Alice's last failed login was at 17:50:06, and her lockout will be cleared at 17:51:06, in 58 seconds. When the duration of 58 seconds elapses, Alice’s name will no longer be displayed.

```
switch# show aaa authentication lockout
 User     Start Time                End Time                  Expires In
--------- ------------------------ ------------------------- ----------
 alice    Fri Jul 12 17:50:06 2020  Fri Jul 12 17:51:06 2020  0:00:58
```

9.1.1.7.39 show aaa counters

The **show aaa counters** command displays the number of service transactions performed by the switch since the last time the counters were reset.

Command Mode

Privileged EXEC

Command Syntax

show aaa counters

Example

This command displays the number of AAA transactions.

```
switch# show aaa counters
Authentication
           Successful:          30
               Failed:           0
 Service unavailable:            0
```

```
Authorization
             Allowed:        188
              Denied:          0
 Service unavailable:          0

Accounting
          Successful:          0
               Error:          0
             Pending:          0

Last time counters were cleared: never switch#
```

9.1.1.7.40 show aaa methods

The **show aaa methods** command displays all the named method lists defined in the specified Authentication, Authorization, and Accounting (AAA) service.

Command Mode

Privileged EXEC

Command Syntax

show aaa methods SERVICE_TYPE

Parameters

SERVICE_TYPE the service type of the method lists that the command displays.

- **accounting** accounting services.
- **authentication** authentication services.
- **authorization** authorization services.
- **all** accounting, authentication, and authorization services.

Example

This command configures the named method lists for all AAA services.

```
switch# show aaa methods all
Authentication method lists for LOGIN:
  name=default methods=group tacacs+, local
Authentication method list for ENABLE:
  name=default methods=local
Authorization method lists for COMMANDS:
  name=privilege0-15 methods=group tacacs+, local
Authentication method list for EXEC:
  name=exec methods=group tacacs+, local
Accounting method lists for COMMANDS:
  name=privilege0-15 default-action=none
Accounting method list for EXEC:
  name=exec default-action=none switch#
```

9.1.1.7.41 show management ldap

The **show management ldap** command displays information about the LDAP configuration.

Command Mode

EXEC

Command Syntax

`show management ldap`

Parameter

- ***no parameter*** state of the system.

The following command shows general information for LDAP.

```
switch# show management ldap
LDAP server: prod-dc-hq1.aristanetworks.com/389
   Binds requested: 6
   Binds successful: 6
   Binds failed: 0
   Binds timed out: 0
   FIPS is ON

Last time counters were cleared: 1:16:41 ago
```

The authentication action in LDAP is the ***bind***, which is equivalent to attempting a log-in. There will be two binds per login attempt, one for the admin account and one for the user account.

The FIPS mode is controlled by the SSL profile in AAA. To validate an SSL profile use the following:

```
switch# show management security ssl profile
   Profile           State
----------------- -----------
   testProfile       valid

To verify a user accounts authorization being performed by ldap, use "show users detail":
switch# show users detail
Session  Username  Roles         TTY  State Duration  Auth       Remote Host
-------- --------- ------------- ---- ----- --------- ---------- ----------------------------------------
1006     erahn     network-admin vty3 E     0:00:05   group ldap fd7a:629f:52a4:dc25:b08d:feff:feed:2ce7
```

To validate the role for a current session the **vty** information in the TTY column must be matched against the Line column in the following command. The row with a "*****" character at the start is the current session where the command was run:

```
switch#show users
   Line      User             Host(s)  Idle       Location
   1 con 0   admin            idle     01:19:00   -
   2 vty 10  srv-sw-ldaptest  idle     01:19:00   172.16.124.151
*  3 vty 3   erahn            idle     00:00:04   fd7a:629f:52a4:dc25:b08d:feff:feed:2ce7
```

9.1.1.7.42 show privilege

The **`show privilege`** command displays the current privilege level for the CLI session.

Command Mode

EXEC

Command Syntax

`show privilege`

Example

This command displays the current privilege level.

```
switch> show privilege
Current privilege level is 15 switch>
```

9.1.1.7.43 show radius

The **show radius** command displays statistics for the RADIUS servers that the switch accesses.

Command Mode

EXEC

Command Syntax

```
show radius
```

Example

This command displays statistics for connected RADIUS servers.

```
switch#show radius
RADIUS server            : radius/10
          Connection opens:        204
         Connection closes:          0
    Connection disconnects:        199
       Connection failures:         10
       Connection timeouts:          2
             Messages sent:       1490
         Messages received:       1490
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: never switch#
```

9.1.1.7.44 show tacacs

The **show tacacs** command displays statistics for the TACACS+ servers that the switch accesses.

Command Mode

EXEC

Command Syntax

```
show tacacs
```

Example

This command displays statistics for connected TACACS+ servers.

```
switch# show tacacs
TACACS+ server            : tacacs/49
          Connection opens:      15942
         Connection closes:          7
    Connection disconnects:       1362
       Connection failures:          0
       Connection timeouts:          0
```

```
             Messages sent:      34395
          Messages received:     34392
             Receive errors:         0
           Receive timeouts:         2
              Send timeouts:         0

Last time counters were cleared: never

TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address
 associated with the
Loopback0 interface switch#
```

9.1.1.7.45 show users accounts

The **show users accounts** command displays the names, roles, and privilege levels of users that are listed in ***running-config***. The SSH public key is also listed for names for which an SSH key is configured.

Command Mode

Privileged EXEC

Command Syntax

show users accounts

Example

This command displays the usernames that are configured on the switch.

```
switch# show users accounts
user: FRED
       role: <unknown>
       privilege level: 1
       ssh public key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDjUg2VDiB
X7In0q
HtN5PyHOWtYvIoeZsxF5YmesQ/rh++mbpT504dL7So+Bpr9T/0qIj+zilat8fX/J
lO42+3pjfkHY/+l
sT2EPNjGTK7uJv1wSGmhc3+90dNmJtr5YVlJFjjQ5m+5Pa+PGe3z4JIV1lY2Nh
LrV2fXtbciLdjnj6F
AlhXjiLt51DJhG13uUxGBJe0+NlGvpEsTJVJvMdJuS6weMi+xSXc9yQimVD2
weJBHsYFnghST2j0pAy
F2S7/EOU13pY42RztDSs42nMNNrutPT0q5Z17aAKvhpd0dDlc+qIwrCrXb
eIChHem7+0N8/zA3alBK4
eKSFSZBd3Pb admin@switch
switch#
user: JANE
       role: sysuser2
       privilege level: 1
user: admin
       role: network-admin
       privilege level: 1
```

9.1.1.7.46 show users detail

The **show users detail** command displays information about active AAA login sessions. Information includes username, roles, TTY, state of the session (pending or established), duration, authentication method, and if available, remote host and remote username.

Command Mode

Privileged EXEC

Command Syntax

show users detail

Example

This command displays information about the active AAA login sessions.

```
switch# show users detail
Session  Username Roles           TTY    State Duration  Auth          Remote Host
-------   ---------- ------------ ------ ----- --------  ------------- ------------
2      admin     network-admin    ttyS0  E     0:01:21  local
4      joe       sysadmin         telnet E     0:02:01  local        sf.example.com
6      alice     sysadmin         ssh    E     0:00:52  group radius ny.example.com
7      bob       sysadmin         ssh    E     0:00:48  group radius la.example.com
8      kim       network-admin1   ssh    E     0:00:55  group radius de.example.com
9      admin     network-admin    ssh    E     0:00:07  local        bj.example.com
10     max       network-admin    telnet E     0:00:07  local        sf.example.com
```

9.1.1.7.47 show users roles

The **show users roles** command displays the name of the default role and the contents of the specified roles. Commands that do not specify a role display the rules in all built-in and configured roles.

Command Mode

Privileged EXEC

Command Syntax

show users roles [**ROLE_LIST**]

Parameters

ROLE_LIST Roles that the command displays. Options include:

- ***no parameter*** Command displays all roles.
- ***role_name*** Name of role displayed by command.

Related Command

The role command places the switch in ***Role*** configuration mode, which is used to create new roles or modify existing roles.

Example

This command displays the contents of all user-defined and built-in roles.

```
switch# show users roles
The default role is network-operator

role: network-admin
        10 permit command .*
role: network-operator
        10 deny mode exec command bash|\|
        20 permit mode exec command .*
role: sysuser
        15 deny mode exec command reload
        20 deny mode config command (no |default )?router
        25 deny mode if command (no |default )?(ip|mac) access-group
        30 deny mode config-all command lacp|spanning-tree
        35 permit command .*
```

```
        40 deny mode exec command .*
        50 permit mode exec command show|clear (counters|platform)|
configure
```

9.1.1.7.48 show users

The **show users** command displays the usernames that are currently logged into the switch.

Command Mode

Privileged EXEC

Command Syntax

show users

Example

This command displays the users that are logged into the switch.

```
switch# show users
    Line       User       Host(s)              Idle       Location
   1 vty 2     john       idle                      1d  10.22.6.113
   2 vty 4     jane       idle                21:33:00  10.22.26.26
*  3 vty 6     ted        idle                00:00:01  10.17.18.71 switch#
```

9.1.1.7.49 tacacs-server host

The **tacacs-server host** command sets communication parameters for communicating with a specific TACACS+ server. These values override global settings when the switch communicates with the specified server.

A TACACS+ server is defined by its server address and port number. Servers with different combinations of address-port-VRF-multiplex settings have separate statements in ***running-config***.

The **no tacacs-server host** and **default tacacs-server host** commands remove settings for the TACACS+ server configuration at the specified address-port-VRF combination by deleting the corresponding **tacacs-server host** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

tacacs-server host **SERVER_ADDR** [**MULTIPLEX**][**VRF_INST**][**PORT**][**TIMEOUT**][**ENCRYPT**]

no tacacs-server host [**SERVER_ADDR**][**MULTIPLEX**][**VRF_INST**][**PORT**]

default tacacs-server host [**SERVER_ADDR**][**MULTIPLEX**][**VRF_INST**][**PORT**]

Parameters

- **SERVER_ADDR** TACACS+ server location. Options include:
  - ***ipv4_addr*** server's IPv4 address.
  - ***ipv6_addr*** server's IPv6 address.
  - ***host_name*** server's DNS host name (FQDN).
- **MULTIPLEX** TACACS+ server support of multiplex sessions on a TCP connection.
  - ***no parameter*** server does not support multiplexing.
  - **single-connection** server supports session multiplexing.

- **VRF_INST** specifies the VRF instance used to communicate with the specified server.
  - ***<no parameter>*** switch communicates with the server using the default VRF.
  - **vrf** *vrf_name* switch communicates with the server using the specified user-defined VRF.
- **PORT** port number of the TCP connection.
  - ***no parameter*** default port of ***49***.
  - **port *number*** port number ranges from ***1*** to ***65535***.
- **TIMEOUT** timeout period (seconds).
  - ***no parameter*** assigns the globally configured timeout value (see tacacs-server timeout).
  - **timeout *number*** timeout period (seconds). Number ranges from ***1*** to ***1000***.
- **ENCRYPT** encryption key the switch and server use to communicate. Settings include:
  - ***no parameter*** assigns the globally configured encryption key (see tacacs-server key).
  - **key *key_text*** where ***key_text*** is in clear text.
  - **key 5 *key_text*** where ***key_text*** is in clear text.
  - **key 7 *key_text*** where ***key_text*** is an encrypted string.

Examples

- This command configures the switch to communicate with the TACACS+ server located at ***10.1.1.5***. The switch uses the global timeout, encryption key, and port settings.

```
switch(config)# tacacs-server host 10.1.1.5
switch(config)#
```

- This command configures the switch to communicate with the TACACS+ server assigned the host name ***TAC_1***. The switch defines the timeout period as ***20*** seconds and the encryption key as ***rp31E2v***.

```
switch(config)# tacacs-server host TAC_1 timeout 20 key rp31E2v
switch(config)#
```

- This command configures the switch to communicate with the TACACS+ server located at ***10.12.7.9***, indicates that the server supports multiplexing sessions on the same TCP connection, and that access is through port ***54***.

```
switch(config)# tacacs-server host 10.12.7.9 single-connection port
 54
switch(config)#
```

9.1.1.7.50 tacacs-server key

The **`tacacs-server key`** command defines the global encryption key the switch uses when communicating with any TACACS+ server for which a key is not defined.

The **`no tacacs-server key`** and **`default tacacs-server key`** commands remove the global key from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`tacacs-server key` [**ENCRYPT_TYPE**] ***encrypt_key***

`no tacacs-server key`

default tacacs-server key

Parameters

- **ENCRYPT_TYPE**encryption level of *encrypt_key*.
  - ***no parameter*** encryption key is entered as clear text.
  - ***0*** encryption key is entered as clear text. Equivalent to ***no parameter***.
  - ***7 encrypt_key*** is an encrypted string.
- ***encrypt_key*** shared key that authenticates the username.
  - ***encrypt_key*** must be in clear text if **ENCRYPT_TYPE** specifies clear text.
  - ***encrypt_key*** must be an encrypted string if **ENCRYPT_TYPE** specifies an encrypted string.

  Encrypted strings entered through this parameter are generated elsewhere.

Related Command

tacacs-server host

Examples

- This command configures ***cv90jr1*** as the encryption key.

```
switch(config)# tacacs-server key 0 cv90jr1
switch(config)#
```

- This command assigns ***cv90jr1*** as the key by specifying the corresponding encrypted string.

```
switch(config)# tacacs-server key 7 020512025B0C1D70
switch(config)#
```

9.1.1.7.51 tacacs-server policy

The **tacacs-server policy** command programs the switch to permit access to TACACS+ servers that send mandatory attribute-value (AV) pairs that the switch does not recognize. By default, the switch denies access to TACACS+ servers when it receives unrecognized AV pairs from the server.

The switch recognizes the following mandatory AV pairs:

priv-lvl=x where x is an integer between ***0*** and ***15***.

The **no tacacs-server policy** and **default tacacs-server policy** commands restore the switch default of denying access to servers from which it receives unrecognized mandatory AV pair by deleting the **tacacs-server policy** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

tacacs-server policy unknown-mandatory-attribute ignore

no tacacs-server policy unknown-mandatory-attribute ignore

default tacacs-server policy unknown-mandatory-attribute ignore

Example

This command configures the switch to permit access to TACACS+ servers that send unrecognized mandatory AV pairs.

```
switch(config)# tacacs-server policy unknown-mandatory-attribute
 ignore
switch(config)#
```

9.1.1.7.52 tacacs-server timeout

The **tacacs-server timeout** command defines the global timeout the switch uses when communicating with any TACACS+ server for which a timeout is not defined.

The **no tacacs-server timeout** and **default tacacs-server timeout** commands restore the global timeout default period of five seconds by removing the **tacacs-server timeout** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

tacacs-server timeout ***time_period***

no tacacs-server timeout

default tacacs-server timeout

Parameters

time_period timeout period (seconds). Values range from ***1*** to ***1000***. Default is ***5***.

Related Command

tacacs-server host

Example

This command configures the switch to wait ***20*** seconds for a TACACS+ server response before issuing an error.

```
switch(config)# tacacs-server timeout 20
switch(config)#
```

9.1.1.7.53 username ssh-key

The **username ssh-key** command configures an SSH key for the specified username. Command options allow the key to be entered directly into the CLI or referenced from a file.

The specified username must be previously configured through a username command.

The **no username ssh-key** and **default username ssh-key** commands delete the SSH key for the specified username by removing the corresponding **username ssh-key** command from ***running-config***.

The **no username ssh-key role** and **default username ssh-key role** commands perform the following:

- delete the SSH key for the specified username by removing the corresponding **username ssh-key** command from ***running-config***.

- delete the role assignment from the specified username by editing the corresponding username statement in ***running-config***.

Command Mode

Global Configuration

Command Syntax

`username` ***name*** **`sshkey`** **KEY**

`no username` ***name*** **`sshkey`** [role]

`default username` ***name*** **`sshkey`** [role]

Parameters

- ***name*** username text that the user enters at the login prompt to access the CLI.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  @ # $ % ^ & * - _ = + ; < > , . ~ |
- **KEY** SSH key. Options include:
  - ***key_text*** username is associated with ssh key specified by ***key_text*** string.
  - **file** ***key_file*** username is associated with SSH key in the specified file.

Example

These commands create the username ***john***, assign it the password ***x245***, then associate it to the SSH key listed in the file named ***john-ssh***.

```
switch(config)# username john secret x245
switch(config)# username john sshkey file john-ssh
switch(config)#
```

9.1.1.7.54 username

The **`username`** command adds a username to the local file and optionally assigns a password to the username. If the command specifies an existing username, the command replaces the password in the local file. The command can also define a username without a password or remove the password from a username.

The **`no username`** command deletes the specified username by removing the corresponding **`username`** statement from ***running-config***. The **`default username`** command removes user-specified usernames, but restores the **admin** username to its default parameters.

The **`no username role`** command assigns the default role assignment to the specified **`username`** statement by editing the corresponding **username** statement in ***running-config***. The **`default username role`** command reverts the specified username to its default role by editing the corresponding **`username`** statement in ***running-config***. For the **admin** username, this restores **network-admin** as its role, even if the **admin** username has been deleted and then created again.

Command Mode

Global Configuration

Command Syntax

`username` ***name*** [**PRIVILEGE_LEVEL**] **SECURITY** [**ROLE_USER**]

`no username` ***name*** [**`role`**]

`default username` ***name*** [**`role`**]

All parameters except ***name*** can be placed in any order.

Parameters

- ***name*** username text that the user enters at the login prompt to access the CLI.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  @ # $ % ^ & * - _ = + ; < > , . ~ |
- **PRIVILEGE_LEVEL** user's initial session privilege level. This parameter is used when an authorization command includes the local option.
  - ***no parameter*** the privilege level is set to ***1***.
  - **privilege *rank*** where rank is an integer between ***0*** and ***15***.
- **SECURITY** password assignment option.
  - **nopassword** *name* is not password protected.
  - **secret *password name*** is protected by specified password (clear-text string).
  - **secret 0 *password name*** is protected by specified password (clear-text string).
  - **secret 5 *password name*** is protected by specified password. (MD5-encrypted string).
  - **secret sha5 *password name*** is protected by specified password (SHA-512-encrypted string).
- **ROLE_USER** specifies the role for performing command authorization. Options include:
  - ***no parameter*** user is assigned default role aaa authorization policy local default-role.
  - **role *role_name*** specifies role assigned to the user.

Guidelines

Encrypted strings entered through this parameter are generated elsewhere. The secret 5 option (**SECURITY**) is typically used to enter a list of username-passwords from a script.

The **SECURITY** parameter is mandatory for unconfigured usernames. For previously configured users, the command can specify a **PRIVILEGE_LEVEL** or **ROLE** without a **SECURITY** setting.

The **admin** username is provided by the initial configuration, but it can be deleted, and its parameters are editable. The initial **admin** configuration is:

```
username admin privilege 1 role network-admin nopassword
```

Note: when deleting the **admin** username, it is advisable to create at least one other username on the switch before saving the configuration.

Examples

- These equivalent commands create the username ***john*** and assign it the password ***x245***. The password is entered in clear text because the **ENCRYPTION** parameter is either omitted or zero.

  ```
  switch(config)# username john secret x245
  switch(config)# username john secret 0 x245
  ```
- This command creates the username ***john*** and assigns it to the text password that corresponds to the encrypted string ***1sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1***. The string was generated by an MD5-encryption program using ***x245*** as the seed.

  ```
  switch(config)# username john secret 5 $1$sU.7hptc$TsJ1qslCL7ZYVb
  yXNG1wg1
  switch(config)#
  ```

  A user authenticates the username ***john*** by entering ***x245*** when the CLI prompts for a password.

- This command creates the username ***jane*** without securing it with a password or removes a password if the ***jane*** username exists.

```
switch(config)# username jane nopassword
switch(config)#
```

- This command removes the username ***william*** from the local file.

```
switch(config)# no username william
switch(config)#
```

9.2 Control Plane Security

This section contains the following topics:

- Transport Layer Security
- 802.1X Port Security

9.2.1 Transport Layer Security

Transport Layer Security (TLS), the successor to Secure Sockets Layer (SSL), is a security protocol used to communicate between client and server. It establishes an encrypted communication channel to secure data.

By default, EOS uses a self signed certificate for client and server connections. However, some browsers or TLS libraries may refuse connections to the default self-signed certificates on EOS and in such cases it is recommended to install the TLS server certificates that meet the following criteria:

- RSA key sizes must be greater than or equal to ***2048*** bits.
- There must be less than ***825*** days to expiry.
- Certificate must use ***SHA-2*** family of Hashing function.

Note: Although Arista switches use TLS, the terms TLS and SSL are used interchangeably in this document.

Following are the two main components used by TLS for authentication of identity before any communication starts.

- Certificate
- Key

An SSL certificate is required to establish a secure connection between the client and server. The certificate includes all of the details which are necessary for authentication. Cryptographic keys are used to provide a secure channel of communication. TLS uses two cryptographic keys: a private keyknown only to the server and a public key embedded in the certificate. The keys are used to validate the certificate.

This chapter contains the following sections.

- Overview
- Configuration
- Rotating Certificate and Key Pair
- Resetting Diffie-Hellman Parameters
- Configuring the TLS Handshake Settings
- Syslog with TLS Support
- Displaying Certificate and Key Information
- TLS Commands

9.2.1.1 Overview

With the SSL certificate, key, and profile management framework we can manage and configure SSL certificates, keys and profiles. SSL is an application-layer protocol which transfers the data securely between the client and server using a combination of authentication, encryption, and data integrity. SSL uses certificates and private-public key pairs to provide this security. An user can configure an SSL profile which includes certificate, key and trusted CA certificates used in SSL communication. A user can manage certificates, keys, and also multiple SSL profiles. A SSL profile can be configured and attached to any other EOS configuration which supports SSL communication. The individual EOS configuration using this framework includes details of using the SSL profile in their configuration.

The only private keys supported are those using the RSA algorithm. Both the certificate and keys must be encoded in the Privacy Enhanced Mail (PEM) format.

Example

This is a code sample of a PEM encoded certificate.

```
$cat server.crt

-----BEGIN CERTIFICATE-----
MIIC3zCCAkgCAQkwDQYJKoZIhvcNAQEEBQAwcTELMAkGA1UEBhMCVVMxCzAJBgNV
BAgMAkNBMQswCQYDVQQHDAJTQzEPMA0GA1UECgwGQXJpc3RhMQwwCgYDVQQLDANT
RE4xCzAJBgNVBAMMAmNhMRwwGgYJKoZIhvcNAQkBFg1jYUBhcmlzdGEuY29tMCAX
DTE0MDgxMTIxNDQxN1oYDzIwNjkwNTE0MjE0NDE3WjB5MQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCQ0ExCzAJBgNVBAcMAlNDMQ8wDQYDVQQKDAZBcmlzdGExDDAKBgNV
BAsMA1NETjEPMA0GA1UEAwwGc2VydmVyMSAwHgYJKoZIhvcNAQkBFhFzZXJ2ZXJA
YXJpc3RhLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOBOP/jh
xk28sUH+lhM/mY6QoyLGcbnygwe/hzIjn2mASnf7uPFGhB62JTt7tQv2xmu/MJfs
aVsNeYXP3ZOcmRO0uk9suGVbII7QJUomnsq1dJh59UyMfws6V6ergmhwEZCDIirV
7nbUDz+uSdNutQL4w/VB+juuWXQ8ztbmygT2ymySaHRK3XnDrAiva0UUVbSmEHH0
wLPsNVNYUxJ4PpOB9luw4upe6ACF9SFtMDz3BDcrL6Gq5idWw3YkQfzBwEl+5hkF
hu0owON29I5T8FpAx+Hzpl48YWW65d/4F40S3XRN312xALM8RrQOU/Chx9Sfg0iJ
dsXWNagx1eyW2EECAwEAATANBgkqhkiG9w0BAQQFAAOBgQBedfuKHvNDpEkdO2AE
Kihs/YeRGgp+5g7hXU0U2TMAMS545ZQ99pFbnScmIC0m68aw1VXILuj+vlkxAM27
oc8iB+gG7oaFtJpWTvmIHqzeHWb0zrwjPhtXTafWEoam8sJZt38Pc4UVb7lQCd6v
ZCLZZJmC2IL0SG7bLN7yaALCSQ==
-----END CERTIFICATE-----
```

Example

This is a code sample of a PEM encoded RSA key.

```
$cat server.key
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA4E4/+OHGTbyxQf6WEz+ZjpCjIsZxufKDB7+HMiOfaYBKd/u4
8UaEHrYlO3u1C/bGa78wl+xpWw15hc/dk5yZE7S6T2y4ZVsgjtAlSiaeyrV0mHn1
TIx/CzpXp6uCaHARkIMiKtXudtQPP65J0261AvjD9UH6O65ZdDzO1ubKBPbKbJJo
dErdecOsCK9rRRRVtKYQcfTAs+w1U1hTEng+k4H2W7Di6l7oAIX1IW0wPPcENysv
oarmJ1bDdiRB/MHASX7mGQWG7SjA43b0jlPwWkDH4fOmXjxhZbrl3/gXjRLddE3f
XbEAszxGtA5T8KHH1J+DSIl2xdY1qDHV7JbYQQIDAQABAoIBACv/TU8NQi+HXqGa
RWe7Juyu9EDi+fXGWutPJz6vfBpenrzQNGOnOE0p3z2+szGIkz0ZQHfcWIISr46O
ymCk6+XQombn5XeEG2vH6jiUQLt0Qk2SRopgWJ8kL4NlAexoZxmYj0AlvGO0jtUn
47VEVt8hWpal/WZteYByWQQQOvoj7EhlANIKUGUG9yOcBcEApdHsgOcXewrbilZQ
d4kbpegxQhjKU9jLUXRsIPV2YFrDcT83/94PFTRk/vFBOnWS/Ygt5vRedcoF2HcR
TJMyE+JwnwGJzTPKwbPJgLnjrEGVrqEerCuTU9v0PpFxu0AOEeV1kPyNbRZzzC+p
al70ZI0CgYEA/Ge1Hj5OBjS2DLUYG0zhmc4xoiIqcgPJCn6QQmfqa6DbscQzXsnp
GqNlwvB76L0mwNOSABX53YaXhFCrYXMRsXWl95hK7NYCGnD6f/fYcr3u+NdBm5+N
```

```
qEiqcq779arAXqGaN/TGKsm507ngyLBrMbUlaPQEXfaBjrQL3f0k7qMCgYEA44AW
1ptUWy0Wx5oa1wdYVgGw+Wbxb8JY88pj29JhE3Qwhy9FBq7x0wZOoQDSkyvbu9aZ
bEiUtctXDDNE/Fy7XgEBMiqrn4R383pbZ5LnmsFKZEcADU1Pe4HjDgSq1rjNSJSA
xEz8xRQvIvy4kpp/tpKsN/Xg2ZBnQmqgj6LWv8sCgYBKeSsWnlmVOS5R94kCXR/f
qtg4N46Aj59dClT0Uwb29MJ95B8oI7k00+ttpllZJZ5unL5iahmMhG7maor2uOYK
j2UF9hh9YvPB633uDin+SQ5eu9yu11gLxE0Og5TyOoyCH3qKch2aeGTtFNY/QNaQ
FxvPqNg1BUva2EL8H/oqswKBgQDjbW1nZSjTbSPUrq4eQG2CrXYqHUtHmlYqgS2K
16nMNN8+hXbP05xUhX2dXqEkFzg3c7U0lupzQq/mtmpEjr+Qnhh/+kBP27G+aZdu
12FJR+oCjSf0JFFM+u/tV6UhuuUdpbeEhiI7Mo5cv6AUjvcVoVMhLmB1nvJbZxTU
AsoEOQKBgHHVuuq4kWDfORCYwTvZPX0byQu++QhlOFfClYBIVuCP4/cU0o3Kw1Z6
tf+zP2rBMOl5eD7IBHjAtOlNjXcFgiyymNxJDKJSmTpw1VePncmuT5UMq8rbljSW
Yg0QGwyvB1Cat2mpgqVWS7YT0nmTooWmQdO3Qp48f+bVvmO/dJXS
-----END RSA PRIVATE KEY-----
```

9.2.1.2 Configuration

- Configuring Certificates
- Configuring Keys
- Configuring a certificate with a RSA key in SSL Profile

9.2.1.2.1 Configuring Certificates

Copying a Certificate to the Switch

The **`copy file: certificate:`** command copies the certificate to the *certificate:* file system from any supported source URLs of the copy command. The source file may contain multiple PEM encoded certificates, but must not contain other entities such as keys.

Example

This command copies a server.crt certificate to the *certificate:* file system.

```
switch(config)#copy file:/tmp/ssl/server.crt certificate:
Copy completed successfully.
switch(config)#
```

Errors while Copying the Certificates

Examples

- The PEM encoded entities in the source file must all be certificates. If the source file contains different types of entities (e.g. a certificate and a key), the copy fails and an error message is displayed as shown.

```
switch(config)#copy file:tmp/ssl/mixed.crt certificate:
% Error copying file:tmp/ssl/mixed.crt to certificate: (Multiple
 types of entities in
certificate file not supported)
switch(config)#
```

- The source file must contain valid PEM encoded certificates. If the file contains invalid certificates, the copy fails and an error message is displayed as shown.

```
switch(config)#copy file:tmp/ssl/bad.crt certificate:
```

```
% Error copying file:tmp/ssl/bad.crt to certificate: (Invalid
 certificate)
switch(config)#
```

- Only certificates with RSA public keys are supported. If the certificate does not have an RSA public key, the copy fails and an error message is displayed as shown.

```
switch(config)#copy file:tmp/ssl/dsa.crt certificate:
% Error copying file:tmp/ssl/dsa.crt to certificate: (Certificate
 does not have
RSA key)
switch(config)#
```

Deleting a Certificate

The **delete certificate** command deletes a certificate configuration from the *certificate:* file system on the switch.

Example

This command deletes the *server.crt* certificate from the switch.

```
switch(config)#delete certificate:server.crt
switch(config)#
```

Generating Certificates

The following commands help the user to generate a self-signed certificate or Certificate Signing Request (CSR).

Examples

- This command generates a self-signed certificate or Certificate Signing Request (CSR). In the example below, the existing private key *test.key* is used to generate the certificates. The user will be prompted for the common name, two-letter country code, etc. A common name is required. The generated Certificate Signing Request (CSR) can be viewed on the CLI, whereas a self-signed certificate will be saved to the certificate: file system.

```
switch#security pki certificate generate self-signed test.crt key
 test.key
Common Name for use in subject: test
[...]
certificate:test.crt generated switch#
```

- This command specifies the digest and the validity (in days) of the certificate. The validity is applicable only for self-signed certificates.

```
switch#security pki certificate generate signing-request key
 test.key digest sha256 validity 365
Common Name for use in subject: test
[...]
certificate:test.crt generated switch#
```

- This command adds the certificate parameters such as common-name, country, email, and others.

```
switch#security pki certificate generate signing-request key
 test.key parameters common-name Test [country US ...]
certificate:test.crt generated switch#
```

9.2.1.2.2 Configuring Keys

Copying a Key to the Switch

The **copy** command copies an RSA key to the sslkey: file system. The key can be copied from any supported source URLs of the copy command. The source file must contain only one key. Password protected keys are not supported.

Example

This command copies a server.key RSA key to the sslkey: file system.

```
switch#copy file:/tmp/ssl/server.key sslkey:
Copy completed successfully.
switch#
```

Errors While Copying the Keys

Examples

- Only one PEM encoded key per file is supported. If the source file contains multiple PEM encoded keys, the copy fails and an error message is displayed as shown.

```
switch#copy file:tmp/ssl/multi.key sslkey:
% Error copying file:tmp/ssl/multi.key to sslkey: (Multiple PEM
 entities in
single file not supported)
```

- The source file must contain a valid PEM encoded RSA key. If the file contains an invalid RSA key, the copy fails and an error message is displayed as shown.

```
switch# copy file:tmp/ssl/bad.key sslkey:
% Error copying file:tmp/ssl/bad.key to sslkey: (Invalid RSA key)
```

- Password protected keys are not supported. If the source file contains a password protected key, the copy fails and an error message is displayed as shown.

```
switch#copy file:/tmp/ssl/pass.key sslkey:
% Error copying file:tmp/ssl/pass.key to sslkey: (Password protected
 keys are not supported)
```

Deleting a Key

The **delete** command deletes the key configuration from the switch.

Example

This command deletes the server.key key from the switch.

```
switch# delete sslkey:server.key
```

Generating Keys

The following commands help the user to generate RSA keys.

Examples

- This command generates a 2048-bit long RSA private key and saves it to sslkey:test.key.

```
switch# security pki key generate rsa 2048 test.key
```

- This command generates a 4096-bit long self-signed certificate RSA key and 2048-bit long certificate signing request RSA key.

```
switch# security pki certificate generate self-signed test.crt key
 test.key
generate rsa 4096
switch#security pki certificate generate signing-request key
 test.key
generate rsa 2048
```

9.2.1.2.3 Configuring a certificate with a RSA key in SSL Profile

A SSL profile is configured with a certificate and its corresponding RSA key. The public key information in the certificate must match the RSA key. This certificate and RSA key pair are used to authenticate to the peer during SSL negotiation. The individual EOS features that use SSL profile configuration will decide whether the certificate and key configuration is optional or mandatory.

Examples

-
```
switch# config
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# certificate server.crt
 key server.key
```

- In this case, if the RSA key configured in SSL profile does not match with the configured certificate, the SSL profile state becomes invalid, and an error message is displayed.

```
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# certificate server.crt key client.key
switch(config-mgmt-sec-ssl-profile-server)# show management security ssl profile
   Profile       State      Error
------------- ------------- ----------------------------------------
   server        invalid    Certificate 'server.crt' does not match
                            with key
```

9.2.1.2.3.1 Configuring SSL Profile with a Certificate Authority (CA)

During SSL negotiation with mutual authentication, the peer (or client) certificate is verified by checking if it is signed by one of these trusted certificates. For peer certificates that do not have a chain to a trusted certificate, the full bundle of certificates leading to the trusted certificates must be included. The individual EOS features that use SSL profile configuration will decide whether the trusted certificate configuration is optional or mandatory.

Example

```
switch# config
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# trust certificate ca1.crt
switch(config-mgmt-sec-ssl-profile-server)# trust certificate ca2.crt
```

9.2.1.2.3.2 Configuring Certificate Chains

Certificate chains are used to provide a chain of trust for the SSL Profile server certificate to a remote party. Several chain certificate commands can be issued to build a certificate chain with many intermediate CAs, regardless of the order. Use the **chain certificate** command to configure the certificate chain for a SSL profile. The **no** form of the command deletes the certificate configuration.

Examples

Assume that *server.crt* is issued by an intermediate CA *intermediate.crt* and *intermediate.crt* itself is issued by the root CA *ca.crt*, as shown in the following figure.

Figure 9-1: Certificate Chain Example



- These commands configure the certificate chain shown schematically in figure Configuring Certificate Chains above. *server.crt* is issued by an intermediate CA *intermediate.crt* and *intermediate.crt* is itself issued by the root CA *ca.crt*.

```
switch#(config)# management security
switch#(config-mgmt-security)# ssl profile server
switch#(config-mgmt-sec-ssl-profile-server)# certificate server.crt
 key server.key
switch#(config-mgmt-sec-ssl-profile-server)# chain certificate
 intermediate.crt
switch#(config-mgmt-sec-ssl-profile-server)# exit
switch(config)#
```

- The other peer can be configured to trust *ca.crt* in order to verify the certificate chain during the TLS handshake as shown below.

```
switch# config
switch#(config)# management security
switch(config-mgmt-security)# ssl profile client
switch(config-mgmt-sec-ssl-profile-client)# certificate client.crt
 key client.key
```

```
switch(config-mgmt-sec-ssl-profile-client)# trust certificate ca.crt
```

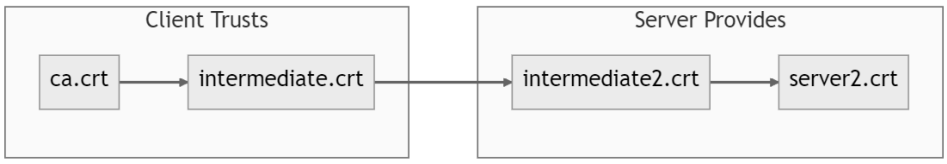
- To check the revocation status of the server certificate chain, the client can add the Certificate Revocation List (CRLs) to its SSL profile configuration. One CRL needs to be specified for every CA in the chain, even if its not revoking any certificate.

```
switch# config
switch#(config)# management security
switch(config-mgmt-security)# ssl profile client
switch(config-mgmt-sec-ssl-profile-client)# crl intermediate.crl
switch(config-mgmt-sec-ssl-profile-client)# crl ca.crl
```

Note: Both the chain certificate and **crl** commands look into the *certificate:* file system to find the right PEM file.

Several **chain certificate** commands can be used to build a certificate chain with multiple intermediate CAs, regardless of the order. The following diagram shows an example certificate chain.

Figure 9-2: Certificate Chain Example 2



This SSL profile can be configured in the following way. Note that the order of intermediate CAs does not matter.

```
switch(config)#management security
switch(config-mgmt-security)#ssl profile server2
switch(config-mgmt-sec-ssl-profile-server2)#certificate server2.crt
 key server2.key
switch(config-mgmt-sec-ssl-profile-server2)#chain certificate
 intermediate2.crt
switch(config-mgmt-sec-ssl-profile-server2)#chain certificate
 intermediate.crt
switch(config-mgmt-sec-ssl-profile-server2)#exit
switch(config-mgmt-security)#exit
switch(config)#
```

A certificate chain can be split into two parts, each part configured on a different peer. The location of the split can be anywhere, as long as between the client and the server, a complete certificate chain can be constructed. The following example shows a server and client SSL profile configuration with a split certificate chain.

Server side:

```
switch(config)#management security
switch(config-mgmt-security)#ssl profile server2
switch(config-mgmt-sec-ssl-profile-server2)#certificate server2.crt
 key server2.key
```

```
switch(config-mgmt-sec-ssl-profile-server2)#chain certificate
 intermediate2.crt
switch(config-mgmt-sec-ssl-profile-server2)#exit
switch(config-mgmt-security)#exit
switch(config)#
```

Client side:

```
switch(config)#management security
switch(config-mgmt-security)#ssl profile client
switch(config-mgmt-sec-ssl-profile-client)#certificate client.crt key
 client.key
switch(config-mgmt-sec-ssl-profile-client)#trust certificate ca.crt
switch(config-mgmt-sec-ssl-profile-client)#trust certificate
 intermediate.crt
switch(config-mgmt-sec-ssl-profile-client)#exit
switch(config-mgmt-security)#exit
switch(config)#
```

The following configuration will not work, as it results in invalid SSL profiles.

Server:

```
switch(config)#management security
switch(config-mgmt-security)#ssl profile server2
switch(config-mgmt-sec-ssl-profile-server2)#certificate server2.crt
 key server2.key
switch(config-mgmt-sec-ssl-profile-server2)#chain certificate
 intermediate.crt
switch(config-mgmt-sec-ssl-profile-server2)#show management security
 ssl profile
   Profile                      State      Additional Info

---------------------------- ------------ --------------------------
--------------
   server3                      invalid    Profile has invalid
 certificate chain
switch(config-mgmt-sec-ssl-profile-server3)#exit
switch(config-mgmt-security)#exit
switch(config)#
```

Client:

```
switch(config)#management security
switch(config-mgmt-security)#ssl profile client3
switch(config-mgmt-sec-ssl-profile-client3)#certificate client3.crt
 key client3.key
switch(config-mgmt-sec-ssl-profile-client3)#trust certificate
 intermediate.crt
switch(config-mgmt-sec-ssl-profile-client3)#show management security
 ssl profile
   Profile                      State      Additional Info

---------------------------- ------------ --------------------------
--------------
   client3                      invalid    Profile has invalid trusted
 certificate
                                           chain
switch(config-mgmt-sec-ssl-profile-client3)#exit
switch(config-mgmt-security)#exit
```

```
switch(config)#
```

9.2.1.2.3.3 Local Certificate Checks

EOS performs various checks on the certificates in an SSL profile before allowing the use of the profile. The way these checks is performed can be modified, added to or relaxed locally. The following are some of the checks that can be performed before any communication with the peer.

Examples

- Check whether the certificate has an extended key usage attribute:

```
switch(config-mgmt-sec-ssl-profile-client)# certificate requirement
 extended-key-usage
```

- Check whether all the trusted certificates or certificates in the chain have CA basic constraints set to true.

```
switch(config-mgmt-sec-ssl-profile-client)# trust certificate
 requirement
basic-constraints ca true
switch(config-mgmt-sec-ssl-profile-client)# chain certificate
 requirement
basic-constraints ca true
```

- Do not mark an expired certificate as invalid.

```
switch(config-mgmt-sec-ssl-profile-client)# certificate policy
 expiry-date ignore
```

9.2.1.2.3.4 Displaying SSL profile status and SSL profile errors

The **show management security ssl profile** command displays the SSL profile status information. To view a specific SSL profile status, use the name of the SSL profile. Otherwise, all SSL profile statuses are displayed.

Example

This command displays the status of the SSL profile server.

```
switch# show management security ssl profile server
 Profile      State
------------- -----------
 server       valid
```

If there are any errors in the SSL profile, an **invalid** state is displayed and the errors are listed in the third column. Once the error is fixed, the SSL profile becomes **valid**.

Examples

- When the certificate ***server.crt*** does not match with the key, the following error message is shown.

```
switch# show management security ssl profile server
 Profile        State       Error
------------ ------------ ----------------------------------------
 server         invalid     Certificate 'server.crt' does not match
                            with key
```

- When a trusted certificate ***ca2.crt*** does not exist, the following error message is shown.

```
switch# show management security ssl profile server
 Profile        State       Error
------------ ------------ -------------------------------------
 server         invalid     Certificate 'ca2.crt' does not exist
```

- When a trusted certificate ***foo.crt*** is not a self-signed root certificate, the following error message is shown.

```
switch# show management security ssl profile server
 Profile        State       Error
------------ ------------ ----------------------------------------
server         invalid     Certificate 'foo.crt' is trusted and not
                           a root certificate
```

- When the certificate ***server.crt*** is expired the following error message is shown.

```
switch# show management security ssl profile server
 Profile        State       Error
------------ ------------ -------------------------------------
 server         invalid     Certificate 'server.crt' has expired
```

- When the certificate chain is missing an intermediate certificate, the following error message is shown.

```
switch# show management security ssl profile server
    Profile        State     Error
------------- ------------ ------------------------------------------------
    server        invalid    Profile has invalid certificate chain
                             Certificate 'intermediate.crt' does not exist
```

9.2.1.3 Rotating Certificate and Key Pair

The certificate and key pair used in SSL profile can be rotated using rotation commands. For example, let's say we want to rotate ***cert.pem*** and ***key.pem*** in the SSL profile ***profile01***.

```
switch01# show running-config section ssl
management security
ssl profile profile01
certificate cert.pem key key.pem
```

Run the **`security pki certificate generate signing-request rotation ssl profile`** command to generate a new key and corresponding signing request for SSL profile ***profile01***. This command also generates a unique rotation ID that can be later used to import the certificate.

```
switch01# security pki certificate generate signing-request rotation ssl
 profile profile01 key generate rsa 2048 parameters common-name switch01
Rotation ID: 2ad7771e8cbc11ebbba37483ef8d9c4b
Certificate Signing Request:
-----BEGIN CERTIFICATE REQUEST-----
MIICZzCCAU8CAQAwEzERMA8GA1UEAwwIc3dpdGNoMDEwggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCy5EsczfEZlAVNZQ8/nfRgEF3bg/tz0XrQJwP/zHhI
UFx1A1VI4O7XhUrYReH1h4OQWhXXX0AHTLTsaClJWHH9m7SXb4iZVo/Y1zXGdyju
1FmnWnNDi72M8f60WXG9gAMtnZK9K53A3lwvrKS+CwJkLCOjlH4xyp1Wsg1+yfay
```

```
AdfXAj+s1Vmg3Rux/XR8iP3N620YVbQ+AfWUQkSNFSsykcTeLvx2WybqX4p4Kids
nqU28ml/NZPS5wEc2OXhagrBn3jHbxdmI33/4SJHN8iNZ6h+gQz+JI18bQrlTHng
RzAx1ENvnz7ZzzeN/n/wh/ArZ6Q9aojrBtAk55aGuY4hAgMBAAGgDzANBgkqhkiG
9w0BCQ4xADANBgkqhkiG9w0BAQsFAAOCAQEAqwQbAsdw6UhpvjDk8OdmXLgCNOSC
jGFLFZe4I67gDmyGQR2lG1brRTQPKp7OphpPxaqr3YvxErEFdQ35gvIUyo9j8qp1
F22yAZGjLqU3prnGLEAZ/I3PcdivNVzL9UJw/JMfHI1CMH6yGtbEI2BXsCTetfxm
JE+N9ujfBlQ/MjUR6IszNxEB2YkFh/DvnVUHoqV0ka+JRmMhGkmTrXwad8bhxYZs
g7cwXktsMLuy2otK21fkFcRvd9OHXssJ2Mf7914ALiDe2sfixHX+35SytR8bahTk
z09HPCkxJmfl+cdhS9SWXrXpHHwXicjwYCj1pqZulBFXtgnVs2Kmd3NnRA==
-----END CERTIFICATE REQUEST-----
```

The complete syntax of the above command is as follows. The **import-timeout** specifies the timeout for this rotation ID. If no certificate is imported within this timeout, the rotation ID expires and will be deleted.

```
switch# security pki certificate generate signing-request rotation ssl profile
 <profile-name>
key generate rsa <2048|3072|4096>
[ import-timeout <minutes> ] (default: 60 mins)
[ digest <sha256|sha384|sha512> ]  (default: sha256) parameters common-name
 <common-name>
[ country <country-code> ]
[ state <state-name> ]
[ locality <locality-name> ]
[ organization <org-name> ]
[ organization-unit <org-unit-name> ]
[ email <email> ]
[ subject-alternative-name [ ip <ip1 ip2 …> ]
[ dns  <nm1 nm2 …> ] [ dns  <nm1 nm2 …> ]
```

Use the **show security pki certificate rotation** command to view the status of rotation IDs.

Example

```
switch# show security pki certificate rotation
Rotation ID                       Profile Name State           Expiry
--------------------------------- ------------ --------------- -------------
------
2ad7771e8cbc11ebbba37483ef8d9c4b  profile01    Import Pending  2021-03-24
 10:15:37
```

Copy the signing request, get it signed by a CA and import the certificate using the **security pki certificate rotation import <rotation-id>** command. Use the rotation ID that was generated with the signing request.

```
switch# security pki certificate rotation import 2ad7771e8cbc11ebbba37483ef
8d9c4b
Enter TEXT certificate. Type 'EOF' on its own line to end.
-----BEGIN CERTIFICATE-----
MIICnTCCAYWgAwIBAgIJANzHst3ljdWfMA0GCSqGSIb3DQEBCwUAMA4xDDAKBgNV
BAMMA2ZvbzAeFw0yMTAzMjQxNjAyMDdaFw0yMjAzMjQxNjAyMDdaMA4xDDAKBgNV
BAMMA2ZvbzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAK2LhqPnQ3Oz
1Pg1PB5toNyCNB60IdCDUVXZcwmyCgS6ifwBYgmw/mCq3iOFncEilaCNIkaFKiWf
b7s43jQd9tmAbnnQw3xUO8jDweus+yCumMNjLLQApbTOZDE4zDonmbWh6kswh8qI
batiz9wR7l5K1bPbbmQx6nO28LrcLCuFSZWrw4R2nprQxdoo5eAotMsGDQdh2vn7
k4yD0CQGVCquVzKI+iVgW7yIfiZ9cwWdFTAlTmkrqQsq+edZmvnuNcOaZm22R5Sb
aPy9osv82oZk8iMX+oDYddY2wMQzLd7ByWlAh4bzCJxNMPIz8hrxU84up0I4srXi
xDVXdL1d2JsCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEADkjfobxF7BAVFdIjyWHL
ID+9D1t96JvCe+PDUyggow6iZE8ROq2fIFHuXhXMrd/neN3WtxqtjvGBnS49t4fa
qIcjerkIPwLaBSwWdpm/1FrIFejYqU0symRE3bKJULLBEdQhyox37D2uqPm71ado
5rXCX9pSu2oNOThd/877QKxtrKa5pekx1acxEa4E0QJ0/YPwkA5nCzM9jy7DZlH2
+cdtCxREeqlhOJUJxQ2354LyykU2fOXe6AGGdVE9hdIOJDnG26VVb+gFt2qaKD5+
3D3/Gd1pm4P3+9aENlhAcr0PUoL3xUApeIdkEf7n8KHiNP+gmlPyVDTCAudwHnwq
```

```
Vg==
-----END CERTIFICATE-----
EOF
Success
```

Commit the rotation ID using the **security pki certificate rotation commit <rotation-id>** command. This will rotate ***cert.pem*** and ***key.pem*** of SSL profile ***profile01*** with new certificate and key.

```
switch# security pki certificate rotation commit 2ad7771e8cbc11ebbba37483ef
8d9c4b
Success
```

Note: If the key is generated outside EOS, then instead of the above workflow, we can rotate the certificate and key pair using a single command as below:

```
switch# security pki certificate rotation commit ssl profile profile01
Enter TEXT private key. Type 'EOF' on its own line to end.
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCti4aj50Nzs9T4
NTwebaDcgjQetCHQg1FV2XMJsgoEuon8AWIJsP5gqt4jhZ3BIpWgjSJGhSoln2+7
ON40HfbZgG550MN8VDvIw8HrrPsgrpjDYyy0AKW0zmQxOMw6J5m1oepLMIfKiG2r
Ys/cEe5eStWz225kMepztvC63CwrhUmVq8OEdp6a0MXaKOXgKLTLBg0HYdr5+5OM
g9AkBlQqrlcyiPolYFu8iH4mfXMFnRUwJU5pK6kLKvnnWZr57jXDmmZttkeUm2j8
vaLL/NqGZPIjF/qA2HXWNsDEMy3ewclpQIeG8wicTTDyM/Ia8VPOLqdCOLK14sQ1
V3S9XdibAgMBAAECggEAEdDMLSD4HTVzDFoBW8mlpQ10G/TNBd1Sk7gY0FV9JCLM
OIPMfzHdeKoB15lcv691DIArP8cQM8A21ab5tKr2JOTuMnDaffXIagyikzb0/tQT
1qhaFeHaHCTFP4yBQKBgQDczahFFYJRP0joT4Hu
iywlkhbyOHV7b9xuPPhqwQxFYqHvEE0qBnmjBzXujbpdb+V18QFGyl0uH4mHr+lt
izcyAbEx5YL/y5Vu08bITZr0mUxS0ZkDXg6n6GKJVIPUH05xSZb/eqtSFIq/DsBQ
YSwu6WzOj4dNpEQAeD8jMmGAlwKBgQDJNWTNyC2JgDYmF039gwNEOY+UuJQ3v/Jo
Ei2IHG4ISxVlZc9lZgLuWHDyS6zNOIeSAYIzDVSsRAGH9sWaK4E4Yno4KHptRC5F
MEbtnrojTO2ANC9JcWo2EgP31r1OJolFpKUiPhOEAzdEYd/sdp9tWEusszTrn8fb
PHvSHUFknQKBgDe8VhByOH4HyoCRqUusp80oDlDAPa+V8f+FtnNEHbPaDORKqh/E
mKm1ZUC9V+DEIRjfaCIVbOX6of21Quga7yjZUoA03hdxrVvXa2Mea9H4bFKvg79c
27g4qb7erZQ6/tML72i370z90HQf5h2kGcIRvBx8EHxhzaSMtetNiV0rAoGAP3Qz
QiJrGf3xFborwlNa6F0uxrwfIiXKkL+K1G4C1WK4cK3W5idxrTD/DaqH6IB3YLhR
E0CU/27C/Nn6H1CxA9MqsCMz2NmzreY3uCBim1dbXx8V+pdl439y+Ooj8U195RSz
b0UcanmJKGulbrFKPfWmh+RMQDK3mJBOjEjlopECgYAr6F+60TZ7ZAvA0vZ9Plrn
tzvY7GhopJgJfAvfi5nBXPS+fkdKtWzOmhW1jon1ka0fEeRQnQjB7DSYB4zldufPKiD+
EXgJtQbhS
qfdtgL7QlhVrpO/s5tUrPE/KRu/yLGtEWruQlDCawpMPA63eP4XER/MHVXBkqbWy
85vx46SisOBAnuEum0yMngru5fARoBKO1aV7G94FI7Eu5rDqeVYsE5jrdnWJTZTg
pHf9RYUOlz8RwwbD/xUs+cKbM1qhaFeHaHCTFP4yBQKBgQDczahFFYJRP0joT4Hu
iywlkhbyOHV7b9xuPPhqwQxFYqHvEE0qBnmjBzXujbpdb+V18QFGyl0uH4mHr+lt
izcyAbEx5YL/y5Vu08bITZr0mUxS0ZkDXg6n6GKJVIPUH05xSZb/eqtSFIq/DsBQ
YSwu6WzOj4dNpEQAeD8jMmGAlwKBgQDJNWTNyC2JgDYmF039gwNEOY+UuJQ3v/Jo
Ei2IHG4ISxVlZc9lZgLuWHDyS6zNOIeSAYIzDVSsRAGH9sWaK4E4Yno4KHptRC5F
MEbtnrojTO2ANC9JcWo2EgP31r1OJolFpKUiPhOEAzdEYd/sdp9tWEusszTrn8fb
PHvSHUFknQKBgDe8VhByOH4HyoCRqUusp80oDlDAPa+V8f+FtnNEHbPaDORKqh/E
mKm1ZUC9V+DEIRjfaCIVbOX6of21Quga7yjZUoA03hdxrVvXa2Mea9H4bFKvg79c
27g4qb7erZQ6/tML72i370z90HQf5h2kGcIRvBx8EHxhzaSMtetNiV0rAoGAP3Qz
QiJrGf3xFborwlNa6F0uxrwfIiXKkL+K1G4C1WK4cK3W5idxrTD/DaqH6IB3YLhR
E0CU/27C/Nn6H1CxA9MqsCMz2NmzreY3uCBim1dbXx8V+pdl439y+Ooj8U195RSz
b0UcanmJKGulbrFKPfWmh+RMQDK3mJBOjEjlopECgYAr6F+60TZ7ZAvA0vZ9Plrn
tzvY7GhopJgGr1GYQPJi38DJ5NR/w64js21t5X2yJ4xcCB3H7R0QWJ9EE+fc+7
nBYFJlaDzSRBbES24yGh4n4Vc6luYW9A+YJR3EaElE6RMWyzIY8J8kV2xuTaK9xe
pdM9x1J1kIm2rA1mcO4Xqw==
-----END PRIVATE KEY-----
EOF
Enter TEXT certificate. Type 'EOF' on its own line to end.
-----BEGIN CERTIFICATE-----
MIICnTCCAYWgAwIBAgIJANzHst3ljdWfMA0GCSqGSIb3DQEBCwUAMA4xDDAKBgNV
```

```
BAMMA2ZvbzAeFw0yMTAzMjQxNjAyMDdaFw0yMjAzMjQxNjAyMDdaMA4xDDAKBgNV
BAMMA2ZvbzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAK2LhqPnQ3Oz
1Pg1PB5toNyCNB60IdCDUVXZcwmyCgS6ifwBYgmw/mCq3iOFncEilaCNIkaFKiWf
b7s43jQd9tmAbnnQw3xUO8jDweus+yCumMNjLLQApbTOZDE4zDonmbWh6kswh8qI
batiz9wR7l5K1bPbbmQx6nO28LrcLCuFSZWrw4R2nprQxdoo5eAotMsGDQdh2vn7
k4yD0CQGVCquVzKI+iVgW7yIfiZ9cwWdFTAlTmkrqQsq+edZmvnuNcOaZm22R5Sb
aPy9osv82oZk8iMX+oDYddY2wMQzLd7ByWlAh4bzCJxNMPIz8hrxU84up0I4srXi
xDVXdL1d2JsCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEADkjfobxF7BAVFdIjyWHL
ID+9D1t96JvCe+PDUyggow6iZE8ROq2fIFHuXhXMrd/neN3WtxqtjvGBnS49t4fa
qIcjerkIPwLaBSwWdpm/1FrIFejYqU0symRE3bKJULLBEdQhyox37D2uqPm71ado
5rXCX9pSu2oNOThd/877QKxtrKa5pekx1acxEa4E0QJ0/YPwkA5nCzM9jy7DZlH2
+cdtCxREeqlhOJUJxQ2354LyykU2fOXe6AGGdVE9hdIOJDnG26VVb+gFt2qaKD5+
3D3/Gd1pm4P3+9aENlhAcr0PUoL3xUApeIdkEf7n8KHiNP+gmlPyVDTCAudwHnwq
Vg==
-----END CERTIFICATE-----
EOF
Success
```

Once the certificate and key pair is rotated, use the **show management security ssl certificate cert.pem** and **show management security ssl key key.pem** commands to display the new contents.

9.2.1.4 Resetting Diffie-Hellman Parameters

The Diffie-Hellman parameters file is used for symmetric key exchange during SSL negotiation. When the system is booted, the system auto generates a Diffie-Hellman parameters file if one does not exist. To reset the auto generated Diffie-Hellman parameters file, use the reset command. The individual features that use SSL profile configuration will decide whether they also use the Diffie-Hellman parameters file. The switch uses 2048-bit Diffie-Hellman parameters with no options to select the size.

Note: Not all features that use SSL profile configuration will use Diffie-Hellman parameters file.

Example

This command resets the Diffie-Hellman parameters file.

```
switch# reset ssl diffie-hellman parameters
```

9.2.1.4.1 Displaying the Diffie-Hellman parameters

The **show management security ssl diffie-hellman** command displays the Diffie-Hellman parameters.

Example

This command displays the Diffie-Hellman parameters.

```
switch# show management security ssl diffie-hellman
Last successful reset on Apr 10 16:18:08 2015
Diffie-Hellman Parameters 1024 bits
 Generator: 2
 Prime:     dc47b5edc0d2b41451432f79f45efab452bba7b1ab118c194d67
1d6752ed1c550
            664ed8f052ad0fdad623c1d54ae5aee5e728d2bd7a6221636b78
7a4c08d1fef8c
```

```
            6dcd10759d38f8b70b47d1c7972d69b0b295a2ee6ab44cfc7352
cb133e85197c8
            9f1fc27aac7e8e02afb4fb01ca1cb05558a7bef505b73a8d06cd
fe403576b
```

9.2.1.5 Configuring the TLS Handshake Settings

During a TLS handshake, both peers send each other a list of the TLS versions they support as a way to agree on and use the highest common version. In a SSL profile the following allowable versions can be configured using the **`tls versions`** command. By default, ***TLSv1***, ***TLSv1.1***, and ***TLSv1.2*** are enabled.

Examples

- This command forces ***TLSv1.2*** to be used. If the other peer does not support this version, the TLS handshake fails.

```
switch# config
switch#(config)# management security
switch(config-mgmt-security)# ssl profile client
switch(config-mgmt-sec-ssl-profile-client)#
switch(config-mgmt-sec-ssl-profile-client)# tls versions 1.2
```

- These commands add support for ***TLSv1.1*** on top of the already configured ***TLSv1.2***.

```
switch(config-mgmt-sec-ssl-profile-client)# tls versions add 1.1
switch(config-mgmt-sec-ssl-profile-client)# tls versions 1.1 1.2
```

Similarly to the TLS version, the cipher suite is negotiated between the client and the server during a TLS handshake. Ideally, the client will send the list of cipher suites it supports and the server will choose a common cipher suite after looking at the clients list as well as its own list of cipher suites. The default cipher-list setting here is an Open SSL cipher string that is **HIGH:!eNULL:!aNULL:!MD5**, which only allows key length larger than 128 bits and forbids cipher suites using MD5. The full list of cipher suites can be expanded using the shell command **openssl ciphers HIGH:!eNULL:!aNULL:!MD5**

Example

This command builds a cipher suite list.

```
switch(config-mgmt-sec-ssl-profile-client)# cipher-list AESGCM
switch(config-mgmt-sec-ssl-profile-client)# cipher-list SHA256:SHA384
switch(config-mgmt-sec-ssl-profile-client)# cipher-list ECDHE-ECDSA-A
ES256-GCM-SHA384
```

9.2.1.5.1 Enabling the Federal Information Processing Standards (FIPS) Mode

Federal Information Processing Standards (FIPS) is a cryptographic standard used to restrict the cryptographic functions and protocol versions that are used by OpenSSL.

Example

This command enables the FIPS mode for a SSL profile.

```
switch(config-mgmt-sec-ssl-profile-client)# fips restrictions
```

9.2.1.6 Syslog with TLS Support

To collect Syslog information on a remote Syslog server define an SSL profile. Traffic to the server is then sent over a TLS connection.

Configuring Syslog with TLS Support

Configure a remote Syslog server with an SSL profile using the following command. It configures a Syslog server with hostname ***test.example.com***. The SSL profile (***test-profile***) is used for communications over port ***1234***.

```
switch(config)# logging host test.example.com 1234 protocol tls ssl-profile
 test-profile
```

SSL Profile Example (Minimal)

The following commands set up a minimal profile to support remote logging over TLS. All relevant configuration on the remote server for TLS communication should also be configured.

```
switch(config-mgmt-security)# ssl profile test-profile
switch(config-mgmt-sec-ssl-profile-test-profile)# certificate clientCert key
 clientKey
switch(config-mgmt-sec-ssl-profile-test-profile)# trust certificate serverCA
```

9.2.1.7 Displaying Certificate and Key Information

- Displaying Certificate Information
- Displaying Key Information

9.2.1.7.1 Displaying Certificate Information

Displaying the Directory Information

The **dir** command displays the directory output of certificate file systems.

Example

This command displays the directory output of certificate: file-system.

```
switch# dir certificate:
Directory of certificate:/
   -rw- 3319 Apr 10 11:50 server.crt
No space information available
```

Displaying the certificate information

The **show management security ssl certificate** command displays the certificate information. To view a specific certificate use the name of the certificate, else all the certificates are displayed.

Example

This command displays the server.crt certificate information.

```
switch# show management security ssl certificate server.crt
Certificate server.crt:
 Version:                    1
 Serial Number:              9
 Issuer:
    Common name:             ca
    Email address:           ca@foo.com
    Organizational unit:     Foo Org
    Organization:            Foo
    Locality:                SC
    State:                   CA
    Country:                 US
 Validity:
    Not before:             Aug 11 21:44:17 2014 GMT
    Not After:              May 14 21:44:17 2069 GMT
 Subject:
    Common name:            server
    Email address:          server@arista.com
    Organizational unit:    Foo Org
    Organization:           Foo
    Locality:               SC
    State:                  CA
    Country:                US
 Subject public key info:
    Encryption Algorithm:   RSA
    Size:                   2048 bits
    Public exponent:        65537
    Modulus:                e04e3ff8e1c64dbcb141fe96133f998e90a322c
671b9f28307bf873
                            2239f69804a77fbb8f146841eb6253b7bb50bf6
c66bbf3097ec695b
                            0d7985cfdd939c9913b4ba4f6cb8655b208ed02
54a269ecab574987
                            ea5ee80085f5216d303cf704372b2fa1aae6275
6c3762441fcc1c04
                            635a831d5ec96d841
```

Displaying Certificate Revocation List (CRL) Information

The **show management security ssl crl** command displays the installed Certificate Revocation List (CRL) information. To view a specific CRL use the name of the CRL, else all the CRLs are displayed.

Example

This command displays the ***intermediate.crl*** information.

```
switch# show management security ssl crl intermediate.crl
CRL intermediate.crl:
   CRL Number: 11
   Issuer:
      Common name: intermediate
      Email address: intermediate@foo.com
      Organizational unit: Foo Org
      Organization: Foo
```

```
      State: CA
      Country: US
   Validity:
      Last Update: Jul 19 19:27:34 2016 GMT
      Next Update: Dec 05 19:27:34 2043 GMT
```

9.2.1.7.2 Displaying Key Information

Displaying the Directory Information

The **dir** command displays the directory output of SSL key file systems.

Example

This command displays the directory output of sslkey: file-system.

```
switch# dir sslkey:
Directory of sslkey:/
   -rw- 1675 Apr 10 12:55 server.key
No space information available
```

Displaying the RSA Key Information

The **show management security ssl key** command displays the RSA key information. To view a specific RSA key use the name of the key, otherwise, all the keys are displayed. For security reasons, only the public part of the key is displayed.

Example

This command displays the server.key key information.

```
switch# show management security ssl key server.key
Key server.key:
 Encryption Algorithm: RSA
 Size:                 2048 bits
 Public exponent:      65537
 Modulus:              e04e3ff8e1c64dbcb141fe96133f998e90a322c
671b9f28307bf873
                       2239f69804a77fbb8f146841eb6253b7bb50bf6
c66bbf3097ec695b
                       0d7985cfdd939c9913b4ba4f6cb8655b208ed02
54a269ecab574987
                       b502f8c3f541fa3bae59743cced6e6ca04f6ca6
c9268744add79c3a
                       f8178d12dd744ddf5db100b33c46b40e53f0a1c
7d49f83488976c5d
```

9.2.1.8 TLS Commands

Configuration Commands

- copy file: certificate:

- copy file: sslkey:
- delete certificate:
- delete sslkey:
- dir certificate:
- dir sslkey:
- reset ssl diffie-hellman parameters
- security pki certificate generate
- security pki key generate
- ssl profile

Show Commands

- show management security ssl certificate
- show management security ssl crl
- show management security ssl diffie-hellman
- show management security ssl key
- show management security ssl profile

9.2.1.8.1 copy file: certificate:

The **`copy file: certificate:`** command copies the certificate to the certificate: file system. The certificate can be copied from any supported source URL of the **`copy`** command.

Command Mode

Global Configuration

Command Syntax

`copy file:` ***file_name*** **`certificate:`**

Parameters

file_name location or the path of the file or the directory where the certificate is saved.

Guidelines

The following requirements apply to copying certificates:

- A single source file can contain multiple PEM encoded entities, but they must all be certificates. If other types such as SSL keys are also included, the copy fails and an error message is displayed as shown.

```
switch(config)#copy file:tmp/ssl/mixed.crt certificate:
% Error copying file:tmp/ssl/mixed.crt to certificate: (Multiple types of
 entities in certificate file not supported)
switch(config)#
```

- The source file must contain valid PEM encoded certificates. If the file contains invalid certificates, the copy fails and an error message is displayed as shown.

```
switch(config)#copy file:tmp/ssl/bad.crt certificate:
% Error copying file:tmp/ssl/bad.crt to certificate: (Invalid certificate)
switch(config)#
```

- Only certificates with RSA public keys are supported. If the certificate does not have an RSA public key, the copy fails and an error message is displayed as shown.

```
switch(config)#copy file:tmp/ssl/dsa.crt certificate:
% Error copying file:tmp/ssl/dsa.crt to certificate: (Certificate does not
 have RSA key)
switch(config)#
```

Example

This command copies a server.crt certificate to the certificate: file system.

```
switch(config)# copy file:/tmp/ssl/server.crt certificate:
Copy completed successfully.
```

9.2.1.8.2 copy file: sslkey:

The **copy file: sslkey:** command copies the SSL key to the sslkey: file system. The key can be copied from any supported source URL of the **copy** command.

Command Mode

Global Configuration

Command Syntax

copy file: ***file_name*** **sslkey:**

Parameters

file_name location or the path of the file or the directory where the key is saved.

Guidelines

The following requirements apply to copying SSL keys:

- Only one PEM encoded key per file is supported. If the source file contains multiple PEM encoded keys, the copy fails and an error message is displayed as shown.

  ```
  switch#copy file:tmp/ssl/multi.key sslkey:
  % Error copying file:tmp/ssl/multi.key to sslkey: (Multiple PEM entities in
   single file not supported)
  ```

- The source file must contain a valid PEM encoded RSA key. If the file contains an invalid RSA key, the copy fails and an error message is displayed as shown.

  ```
  switch#copy file:tmp/ssl/bad.key sslkey:
  % Error copying file:tmp/ssl/bad.key to sslkey: (Invalid RSA key)
  ```

- Password protected keys are not supported. If the source file contains a password protected key, the copy fails and an error message is displayed as shown.

  ```
  switch#copy file:/tmp/ssl/pass.key sslkey:
  % Error copying file:tmp/ssl/pass.key to sslkey: (Password protected keys
   are not supported)
  ```

Example

This command copies an SSL key in the file *server.key* to the sslkey: file system.

```
switch(config)#copy file:/tmp/ssl/server.key sslkey:
Copy completed successfully.
switch(config)#
```

9.2.1.8.3 delete certificate:

The **delete certificate:** command deletes a specified certificate from certificate: file system on the switch.

Command Mode

Global Configuration

Command Syntax

`delete certificate:` ***certificate_name***

Parameters

certificate_name name of the certificate to be deleted.

Example

This command deletes the server.crt certificate from the switch.

```
switch(config)# delete certificate:server.crt
```

9.2.1.8.4 delete sslkey:

The **`delete sslkey:`** command deletes a SSL key from sslkey: file system on a switch.

Command Mode

Global Configuration

Command Syntax

`delete sslkey:` ***key_name***

Parameters

key_name name of the key.

Example

This command deletes the server.key SSL key on the switch.

```
switch(config)# delete sslkey:server.key
```

9.2.1.8.5 dir certificate:

The **`dir certificate:`** command displays the directory output of certificate: file system on the switch.

Command Mode

Global Configuration

Command Syntax

`dir certificate:`

Example

This command shows the directory output of certificate: file system on the switch.

```
switch(config)# dir certificate:
Directory of certificate:/
   -rw- 3319 Apr 10 11:50 server.crt
No space information available
```

9.2.1.8.6 dir sslkey:

The **`dir sslkey:`** command displays the directory output of sslkey: file system on the switch.

Command Mode

Global Configuration

Command Syntax

dir sslkey:

Example

This command shows the directory output of sslkey: file system on the switch.

```
switch(config)# dir sslkey:
Directory of sslkey:/
   -rw- 1675 Apr 10 12:55 server.key
No space information available
```

9.2.1.8.7 reset ssl diffie-hellman parameters

The **reset ssl diffie-hellman parameters** command resets the Diffie-Hellman parameters file after a system reboot.

Command Mode

Global Configuration

Command Syntax

reset ssl diffie-hellman parameters

Example

This command resets the Diffie-Hellman parameters file.

```
switch(config)# reset ssl diffie-hellman parameters
switch(config)#
```

9.2.1.8.8 security pki certificate generate

The **security pki certificate generate** command is used to generate a self-signed certificate or a Certificate Signing Request (CSR) certificate. The generated CSR is displayed on the CLI, whereas a self-signed certificate is saved to the certificate: file system.

Many other parameters can be entered and applied to the certificate as shown in the following examples below.

Command Mode

Global Configuration

Command Syntax

security pki certificate generate {**self-signed** | **signing-request**} ***certificate_name*** **Key** ***key_name***

Parameters

- ***certificate_name*** name of the certificate to generate. Options includes:
  - self-signed request to generate self-signed certificate.
  - signing-request request to generate signing-request.
  - **digest** signs the certificate or key with the following cryptographic hash algorithm (***sha256***, ***sha384***, ***sha512***).
  - ***key_name*** name of the key to modify.
- **parameters** signing request parameters for a certificate. Option includes:
  - ***common-name*** common name for use in subject.
  - ***country*** two-letter country code for use in subject.
  - ***email*** email address for use in subject.

- ***locality*** locality name for use in subject.
- ***organization*** organization name for use in subject.
- ***organization-unit*** organization Unit Name for use in subject.
- ***state*** state for use in subject.
- ***subject-alternative-name*** subject alternative name extension.
- ***rotation*** to generate a unique rotation ID.

- ***validity*** validity of the certificate in days. Value ranges from ***1*** to ***30000*** .

Examples

- This command generates a self-signed certificate or CSR certificate. In the example below an existing private key (***test.key***) is used to generate the certificates.

```
switch(config)# security pki certificate generate self-signed test.crt key
 test.key
```

- This command specifies the digest and the validity (in days) of the certificate or key.

```
switch(config)# security pki certificate generate signing-request key
 test.key digest sha256 validity 365
```

- This command adds the certificate parameters such as common-name, country, email, and others.

```
switch(config)# security pki certificate generate signing-request key
 test.key parameters common-name Test [country US ...]
```

9.2.1.8.9 security pki key generate

The **`security pki key generate`** command generates a RSA key used to validate a specific certificate.

The key generated can be modified and saved by entering the value of the length in generate rsa <length> parameter.

Command Mode

Global Configuration

Command Syntax

`security pki key generate` ***rsa key_name***

Parameters

- *rsa* use Rivest-Shamir-Adleman (RSA) algorithm. Options include.
  - ***2048*** Use 2048-bit keys.
  - ***3072*** Use 3072-bit keys.
  - ***4096*** Use 4096-bit keys.
- ***key_name*** name of the key to generate.

Examples

- This command generates a a 2048-bit long RSA private key(***test.key***) and save it to sslkey:test.key.

```
switch(config)#security pki key generate rsa 2048 test.key
```

- This command modifies the generated RSA key length value.

```
switch(config)# security pki certificate generate self-signed test.crt key
test.key generate rsa 4096
switch(config)# security pki certificate generate signing-request key
 test.key
```

```
generate rsa 2048
```

9.2.1.8.10 show management security ssl certificate

The **show management security ssl certificate** command displays information about the certificate. Provide the name of the certificate if you want to view more information of the certificate. If no name is provided, this command displays information of all the certificates.

Command Mode

EXEC

Command Syntax

show management security ssl certificate [***certificate_name***]

Parameter

certificate_name name of the certificate. This is optional.

Example

This command displays the server.crt certificate information.

```
switch# show management security ssl certificate server.crt
Certificate server.crt:
 Version:                    1
 Serial Number:              9
 Issuer:
    Common name:             ca
    Email address:           ca@foo.com
    Organizational unit:     Foo Org
    Organization:            Foo
    Locality:                SC
    State:                   CA
    Country:                 US
 Validity:
    Not before:             Aug 11 21:44:17 2014 GMT
    Not After:              May 14 21:44:17 2069 GMT
 Subject:
    Common name:            server
    Email address:          server@arista.com
    Organizational unit:    Foo Org
    Organization:           Foo
    Locality:               SC
    State:                  CA
    Country:                US
 Subject public key info:
    Encryption Algorithm:   RSA
    Size:                   2048 bits
    Public exponent:        65537
    Modulus:                e04e3ff8e1c64dbcb141fe96133f998e90a322c671b9f28307bf873
                            2239f69804a77fbb8f146841eb6253b7bb50bf6c66bbf3097ec695b
                            0d7985cfdd939c9913b4ba4f6cb8655b208ed0254a269ecab574987
                            9f54c8c7f0b3a57a7ab826870119083222ad5ee76d40f3fae49d36e
                            b502f8c3f541fa3bae59743cced6e6ca04f6ca6c9268744add79c3a
                            c08af6b451455b4a61071f4c0b3ec3553585312783e9381f65bb0e2
                            ea5ee80085f5216d303cf704372b2fa1aae62756c3762441fcc1c04
                            97ee6190586ed28c0e376f48e53f05a40c7e1f3a65e3c6165bae5df
                            f8178d12dd744ddf5db100b33c46b40e53f0a1c7d49f83488976c5d
                            635a831d5ec96d841
```

9.2.1.8.11 show management security ssl crl

The **show management security ssl crl** command displays the basic information on the installed Certificate Revocation List (CRLs).To view information of a specific CRL provide the name of the CRL. If no name is provided, this command shows information of all the CRLs.



Note: The command only shows basic information and does not show any information on the revocation status of certificates.

Command Mode

EXEC

Command Syntax

show management security ssl crl

Example

This command displays the basic information of the intermediate.crl CRL.

```
switch# show management security ssl crl intermediate.crl
CRL intermediate.crl:
   CRL Number: 11
   Issuer:
      Common name: intermediate
      Email address: intermediate@foo.com
      Organizational unit: Foo Org
      Organization: Foo
      State: CA
      Country: US
   Validity:
      Last Update: Jul 19 19:27:34 2016 GMT
      Next Update: Dec 05 19:27:34 2043 GMT
```

9.2.1.8.12 show management security ssl diffie-hellman

The **show management security ssl diffie-hellman** command displays the Diffie-Hellman parameter information.

Command Mode

EXEC

Command Syntax

show management security ssl diffie-hellman

Example

This command displays the Diffie-Hellman parameter information.

```
switch# show management security ssl diffie-hellman
Last successful reset on Apr 10 16:18:08 2015
Diffie-Hellman Parameters 1024 bits
 Generator: 2
 Prime:     dc47b5edc0d2b41451432f79f45efab452bba7b1ab118c194d671d6752ed1c550
            664ed8f052ad0fdad623c1d54ae5aee5e728d2bd7a6221636b787a4c08d1fef8c
            6dcd10759d38f8b70b47d1c7972d69b0b295a2ee6ab44cfc7352cb133e85197c8
            9f1fc27aac7e8e02afb4fb01ca1cb05558a7bef505b73a8d06cdfe403576b
```

9.2.1.8.13 show management security ssl key

The **show management security ssl key** command displays the RSA key information. To view information of a specific key, provide the name of the key in the command. If no name is provided, this command displays information of all the keys.

Note: For security reasons, only the public part of the key is shown.

Command Mode

EXEC

Command Syntax

show management security ssl key [***key_name***]

Parameter

key_name name of the key. This is optional.

Example

This command displays the server.key key information.

```
switch# show management security ssl key server.key
Key server.key:
Encryption Algorithm: RSA
Size:                 2048 bits
Public exponent:      65537
Modulus:              e04e3ff8e1c64dbcb141fe96133f998e90a322c671b9f28307bf873
                      2239f69804a77fbb8f146841eb6253b7bb50bf6c66bbf3097ec695b
                      0d7985cfdd939c9913b4ba4f6cb8655b208ed0254a269ecab574987
                      9f54c8c7f0b3a57a7ab826870119083222ad5ee76d40f3fae49d36e
                      b502f8c3f541fa3bae59743cced6e6ca04f6ca6c9268744add79c3a
                      c08af6b451455b4a61071f4c0b3ec3553585312783e9381f65bb0e2
                      ea5ee80085f5216d303cf704372b2fa1aae62756c3762441fcc1c04
                      97ee6190586ed28c0e376f48e53f05a40c7e1f3a65e3c6165bae5df
                      f8178d12dd744ddf5db100b33c46b40e53f0a1c7d49f83488976c5d
                      635a831d5ec96d841
```

9.2.1.8.14 show management security ssl profile

The **`show management security ssl profile`** command displays the SSL profile status information. To display information of a specific SSL profile, provide the name of the profile. If no name is provided, this command displays profile status of all the SSL profiles.

If there are any errors in the SSL profile, the state is shown invalid and the errors are listed in the third column as shown in the example below.

Command Mode

EXEC

Command Syntax

`show management security ssl profile` [***profile_name***]

Parameter

profile_name name of the SSL profile, this is optional.

Examples

- This command displays the SSL profile status of profile server.

```
switch# show management security ssl profile server
 Profile      State
------------- -----------
 server       valid
```

- When the certificate ***server.crt*** does not match with the key the following error occurs.

```
switch# show management security ssl profile server
Profile       State         Error
------------- ------------- ----------------------------------------
server        invalid       Certificate 'server.crt' does not match
                            with key
```

- When a trusted certificate ***ca2.crt*** does not exist the following error occurs.

```
switch# show management security ssl profile server
```

```
Profile       State         Error
------------- ------------- ----------------------------------------
server        invalid       Certificate 'ca2.crt' does not exist
```

- When a trusted certificate ***foo.crt*** is not self-signed root certificate the following error occurs.

```
switch# show management security ssl profile server
Profile       State         Error
------------- ------------- ----------------------------------------
server         invalid      Certificate 'foo.crt' is trusted and not
                            a root certificate
```

- When the certificate ***server.crt*** is expired the following error occurs.

```
switch# show management security ssl profile server
Profile       State         Error
------------- ------------- ----------------------------------------
 server       invalid       Certificate 'server.crt' has expired
```

- When the certificate chain is missing an intermediate certificate the following error occurs.

```
switch# show management security ssl profile server
Profile        State         Error
-------------- ------------- --------------------------------------------
server         invalid       Profile has invalid certificate chain
                             Certificate 'intermediate.crt' does not exist
```

9.2.1.8.15 ssl profile

The **`ssl profile`** command places the switch in the SSL profile configuration mode. Various SSL profile management configurations are allowed in this mode. For example, this mode allows to configure a SSL profile with a certificate and its corresponding RSA key.

Similarly, other configurations such as trust certificate, chain certificate, crl, tls, cipher-list can be configured to a SSL profile in this mode.

The no form of the command deletes the SSL profile management configuration from ***running-config***.

Command Mode

Management Security Mode

SSL Profile Mode

Command Syntax

`ssl profile` ***profile_name***

Parameter

profile_name name of the profile.

Examples

- These commands place the switch in SSL profile mode.

```
switch# config
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)#
```

- These commands configure SSL profile server with a certificate and its corresponding RSA key. The **no** command deletes the certificate configuration.

```
switch# config
```

```
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# certificate server.crt key
 server.key
switch(config-mgmt-sec-ssl-profile-server)# no certificate server.crt key
 server.key
```

- These commands configure the trust certificate ca1.crt to an SSL profile. The **no** command deletes a trusted certificate configuration.

```
switch# config
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# trust certificate ca1.crt
switch(config-mgmt-sec-ssl-profile-server)# no trust certificate ca1.crt
```

- These commands configure the intermediate.crt chain certificate to an SSL profile. The **no** command deletes a chain certificate configuration.

```
switch# config
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# certificate server.crt key
 server.key
switch(config-mgmt-sec-ssl-profile-server)# chain certificate intermediate.
crt
switch(config-mgmt-sec-ssl-profile-server)# no chain certificate
 intermediate.crt
```

- These commands provides Certificate Revocation List (CRL) to a SSL profile to check the revocation status of the certificate chain. The **no** command deletes the CRL configuration.

```
switch# config
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# crl intermediate.crl
switch(config-mgmt-sec-ssl-profile-server)# crl ca.crl
switch(config-mgmt-sec-ssl-profile-server)# no crl ca.crl
```

- These commands configure ***TLSv1.2*** to be used in the SSL profile.

```
switch# config
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# tls versions 1.2
```

- These commands build a cipher suite list.
-

```
switch# config
switch(config)# management security
switch(config-mgmt-security)# ssl profile server
switch(config-mgmt-sec-ssl-profile-server)# cipher-list AESGCM
switch(config-mgmt-sec-ssl-profile-server)# cipher-list SHA256:SHA38
switch(config-mgmt-sec-ssl-profile-server)# cipher-list ECDHE-ECDSA-AES256-
GCM-SHA384
```

- This command check that the certificate has an extended key usage attribute.

```
switch(config-mgmt-sec-ssl-profile-client)#certificate requirement extended-
key-usage
```

- These commands check that all the trusted certificates or certificates in the chain have a CA basic constraints set to true.

```
switch(config-mgmt-sec-ssl-profile-client)# trust certificate requirement
 basic-constraints ca true
switch(config-mgmt-sec-ssl-profile-client)# chain certificate requirement
 basic-constraints ca true
```

- This command enables the Federal Information Processing Standards (FIPS) mode for a SSL profile.

```
switch(config-mgmt-sec-ssl-profile-client)# fips restrictions
```

9.2.2 802.1X Port Security

This section explains the basic concepts behind 802.1X port security, including switch roles, how the switches communicate, and the procedure used for authenticating clients.

- 802.1X Port Security Introduction
- 802.1X Port Security Description
- Configuring 802.1X Port Security
- 802.1X AAA Unresponsive VLAN
- 802.1X Web Authentication
- Displaying 802.1X Information
- 802.1X Port Security Commands

9.2.2.1 802.1X Port Security Introduction

802.1X is an IEEE standard protocol that prevents unauthorized devices from gaining access to the network.

802.1X defines three device roles,

- Supplicant (client).
- Authenticator (switch).
- Authentication server (RADIUS).

Before authentication can succeed, switchport is in unauthorized mode and blocks all traffic but, after authentication has succeeded, normal data can then flow through the switchport.

Port security control who can send or receive traffic from an individual switch port. An end node is not allowed to send or receive traffic through a port until the node is authenticated by a RADIUS server.

This prevents unauthorized individuals from connecting to a switch port to access your network. Only designated valid users on a RADIUS server will be allowed to use the switch to access the network.

9.2.2.2 802.1X Port Security Description

802.1X port security controls can send traffic through and receive traffic from the individual switch ports. A supplicant must authenticate itself using EAPoL packets with the switch before it can gain full access to the port. Arista switches act as an authenticator, passing the messages from 802.1X supplicants through to the RADIUS server and vice versa. 802.1X can operate in three different modes:

- Single Host Mode: Once the 802.1X supplicant is authenticated on the port, only the traffic coming from the supplicant's MAC is allowed through the port.
- Multi-Host Mode: Once the 802.1X supplicant is authenticated on the port, traffic coming from any source MAC is allowed through the port.
- Multi-Host authenticated Mode: Multiple 802.1X supplicants are allowed and the traffic coming from all authenticated supplicant’s MAC address is only allowed through the port.

The Single Host and the Multi-Host modes allow only one 802.1X supplicant to be authenticated for one port. Once it is successfully authenticated, no other 802.1X supplicant can be authenticated, unless the current one logs off. However, the Multi-Host authenticated Mode allows multiple 802.1X supplicants to be authenticated and provided access to the network.

Apart from 802.1X authentication, Arista switches also support MAC-Based Authentication (MBA), which allows devices not speaking 802.1X to have access to the network. The authenticator uses the MAC address of such devices as username/password in its RADIUS request packets. Depending on the MAC-Based Authentication configuration on the RADIUS server, it decides whether to authenticate the supplicant or not. Unlike 802.1X supplicants, multiple MBA supplicants are allowed on a single port. The MBA configuration is independent of the 802.1X host modes. MBA supplicants will not be considered to allow or reject unauthenticated traffic, based on the host mode.

Arista switches also support Dynamic VLAN assignment, which allows the RADIUS server to indicate the desired VLAN for the supplicant, using the tunnel attributes with the Access-Accept message. Both 802.1X and MBA supplicants can be assigned a VLAN via the RADIUS server. Note that only one VLAN per port is supported. When the first host authenticates, the authenticator port is put in the respective VLAN (via dynamic VLAN assignment) and subsequently, all other hosts must belong to that VLAN as well.

802.1X features are now supported on 802.1Q trunk ports allowing the user to have Port-Based Network Access Control (PNAC) on such a port. With this feature, traffic coming into an 802.1X enabled port with a VLAN tag can also be authenticated via both 802.1X or MBA.

By default, traffic from any unauthenticated device on an 802.1X enabled port is dropped. By configuring Authentication Failure VLAN on the authenticator switch, 802.1X or MBA supplicants traffic can be put into a specific VLAN, if the supplicant fails to authenticate via the RADIUS server.



Note: Only one configurable VLAN for failure is supported. That is, failure due to server timeout, server unreachable, server AUTH-FAIL, or Quarantine.

9.2.2.2.1 Switch Roles for 802.1X Configurations

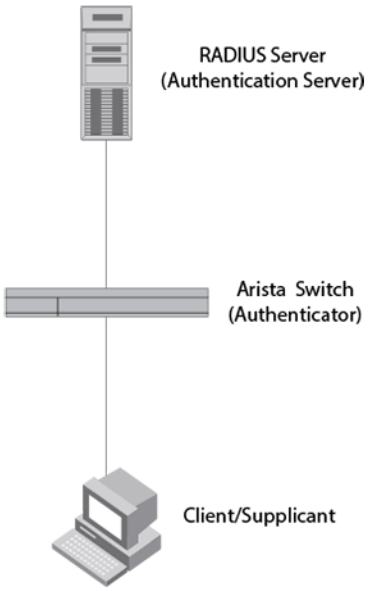
The 802.1X standard specifies the roles of **Supplicant (client)**, **Authenticator**, and **Authentication Server** in a network. Switch Roles for 802.1X Configurations illustrates these roles.

Authentication Server The switch that validates the client and specifies whether or not the client may access services on the switch. The switch supports Authentication Servers running RADIUS.

Authenticator The switch that controls access to the network. In an 802.1X configuration, the switch serves as the Authenticator. As the Authenticator, it moves messages between the client and the Authentication

Server. The Authenticator either grants or does not grant network access to the client based on the identity data provided by the client, and the authentication data provided by the Authentication Server.

Figure 9-3: Authenticator, Supplicant, and Authentication Server in an 802.1X configuration



Supplicant/Client The client provides a username or password data to the Authenticator. The Authenticator sends this data to the Authentication Server. Based on the supplicants information, the Authentication Server determines whether the supplicant can use services given by the Authenticator. The Authentication Server gives this data to the Authenticator, which then provides services to the client, based on the authentication result.

9.2.2.2.2 Authentication Process

The authentication that occurs between a supplicant, authenticator, and authentication server include the following processes.

- Either the authenticator (a switch port) or the supplicant starts an authentication message exchange. The switch starts an exchange when it detects a change in the status of a port, or if it gets a packet on the port with a source MAC address that is not included in the MAC address table.
- An authenticator starts the negotiation by sending an EAP-Request/Identity packet. A supplicant starts the negotiation with an EAPOL-Start packet, to which the authenticator answers with a EAP-Request/ Identity packet.
- The supplicant answers with an EAP-Response/Identity packet to the authentication server via the authenticator.
- The authentication server responds with an EAP-Request packet to the supplicant via the authenticator.
- The supplicant responds with an EAP-Response.
- The authentication server transmits either an EAP-Success packet or EAP-Reject packet to the supplicant.
- If an EAP-Reject is received, the supplicant will receive an EAP-Reject message and their traffic will not be forwarded.

9.2.2.2.3 Communication Between the Switches

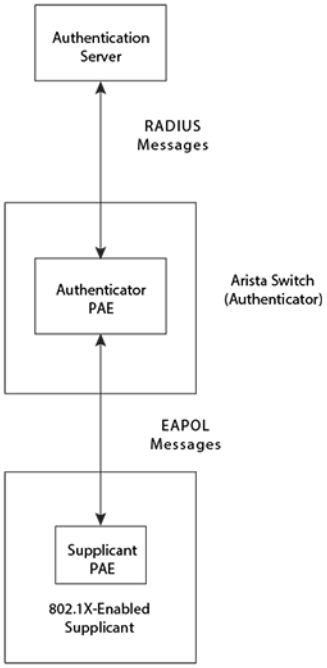
For communication between the switches, 802.1X port security uses the Extensible Authentication Protocol (EAP), defined in ***RFC 2284*** and the RADIUS authentication protocol.

The 802.1X standard defines a method for encapsulating EAP messages so they can be sent over a LAN. This encapsulated kind of EAP is known as EAP over LAN (EAPOL). The standard also specifies

a means of transferring the EAPOL information between the client or Supplicant, Authenticator, and Authentication Server.

EAPOL messages are passed between the Supplicants and Authenticators Port Access Entity (PAE). The figure below shows the relationship between the Authenticator PAE and the Supplicant PAE.

Figure 9-4: Authenticator PAE and Supplicant PAE



Authenticator PAE: The Authenticator PAE communicates with the Supplicant PAE to receive the Supplicants identifying information. Behaving as a RADIUS client, the Authenticator PAE passes the Supplicants information to the Authentication Server, which decides whether to grant the Supplicant access. If the Supplicant passes authentication, the Authenticator PAE allows it access to the port.

Supplicant PAE: The Supplicant PAE provides information about the client to the Authenticator PAE and replies to requests from the Authenticator PAE. The Supplicant PAE may initiate the authentication procedure with the Authenticator PAE, as well as send logoff messages.

9.2.2.2.4 Dot1x Dropped Counters

The Dot1x Dropped Counters count the packets dropped by dot1x interfaces. The dropped counter will not represent all the dropped packets in case of high volume dropping, and the CPU queue drop counter will reflect the rest of the dropped packet counter. This is due to the fact that EOS limits the bandwidth for the packets that get sent to the CPU.

The following counters are supported and increment depending on the ***dot1x interface*** configuration mode:

- EAPOL unauthorized port (indicates the dropped packet number due to the unauthorized EAPOL port when Mac Base Authorization is disabled).
- EAPOL unauthorized host (indicates the dropped packet number due to the unauthorized EAPOL host).
- MBA unauthorized host (counts the dropped packet due to the unauthorized host when Mac Base Authorization is enabled.)

9.2.2.2.5 Enable 802.1X Port Control

To enable 802.1X port authentication on the switch, global command configuration is required:

```
switch(config)# dot1x system-auth-control
```

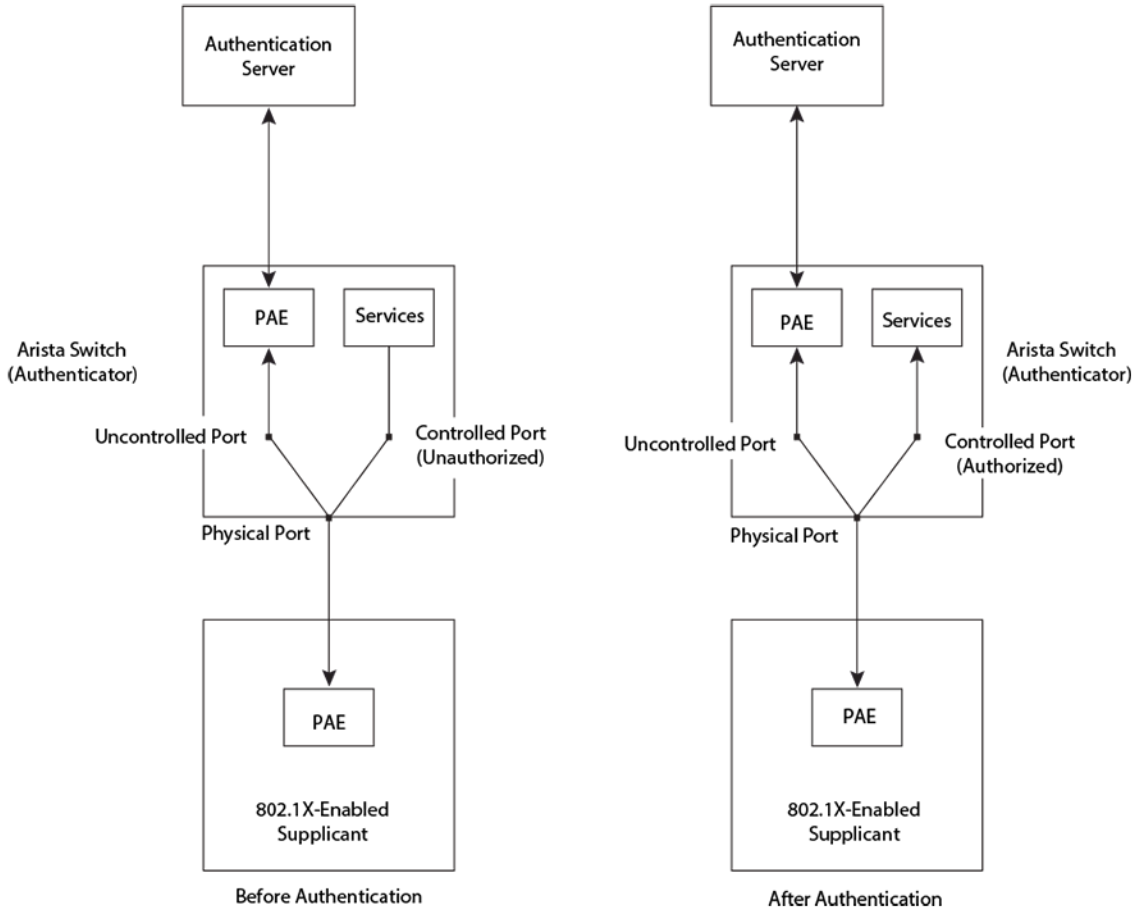
Port mode can be set to access/trunk port and 802.1X port access entity is set to authenticator:

```
switch(config-if-Et1)# switchport mode access
switch(config-if-Et1)# dot1x pae authenticator
```

9.2.2.2.6 Controlled and Uncontrolled Ports

A physical port on the switch used with 802.1X has two virtual access points that include a controlled port and an uncontrolled port. The controlled port grants full access to the network. The uncontrolled port only gives access for EAPOL traffic between the client and the Authentication Server. When a client is authenticated successfully, the controlled port is opened to the client.

Figure 9-5: Ports Before and After Client Authentication



9.2.2.2.6.1 Control Port State

Before the port is authenticated, the port is in an unauthorized state. In this state, only EAPOL packets are processed by 802.1X agent and all other packets are dropped. After the port is successfully authenticated, the port is in the authorized state and all packets are allowed to pass. The state transition is controlled by authentication exchange between supplicant and authentication server. However, the user can control the state by using any one of the following commands:

`dot1x port-control` force-authorized

force-authorized: disables 802.1X authentication and directly put the port to the authorized state. This is the default setting.

`dot1x port-control` force-unauthorized

force-unauthorized: also disables 802.1X authentication and directly put the port to unauthorized state, ignoring all attempts by the client to authenticate.

`dot1x port-control` auto

auto: enables 802.1X authentication and put the port to unauthorized state first. The port state remains in an unauthorized state or transit to authorized state according to authentication result and configuration.

9.2.2.2.6.2 Uncontrolled Port State

The uncontrolled port on the Authenticator is the only one open before a client is authenticated. The uncontrolled port permits only EAPOL frames to be swapped between the client and the Authentication Server. No traffic is allowed to pass through the controlled port in the unauthorized state.

During authentication, EAPOL messages are swapped between the Supplicant PAE and the Authenticator PAE, and RADIUS messages are swapped between the Authenticator PAE and the Authentication Server. If the client is successfully authenticated, the controlled port becomes authorized, and traffic from the client can flow through the port normally.

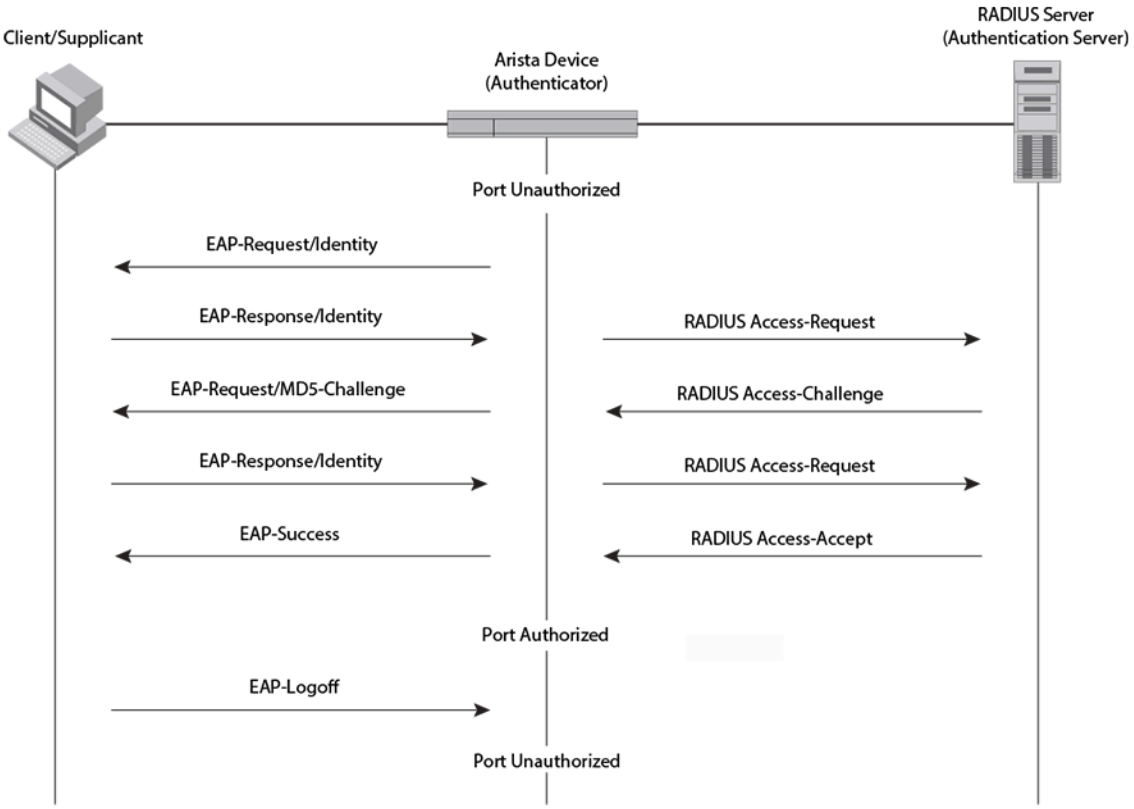
All controlled ports on the switch are placed in the authorized state, allowing all traffic, by default. When authentication is initiated, the controlled port on the interface is initially set in the unauthorized state. If a client connected to the port is authenticated successfully, the controlled port is set in the authorized state.

9.2.2.2.7 Message Exchange During Authentication

The figure below illustrates an exchange of messages between an 802.1X-enabled client, a switch operating as Authenticator, and a RADIUS server operating as an Authentication Server.

Arista switches support MD5-challenge TLS and any other EAP-encapsulated authentication types in EAP Request or Response messages. In other words, the switches are transparent to the authentication scheme used.

Figure 9-6: Message Exchange During Authentication



9.2.2.2.8 Authenticating Multiple Clients Connected to the Same Port

Arista switches support 802.1X authentication for ports with more than one client connected to them. Figure 9: Multiple clients connected to a 802.1X-enabled port illustrates a sample configuration where multiple clients are connected to a single 802.1X port. 802.1X authentication may use multi-host mode, or (on selected switches) single-host mode. In both modes, the port authenticates the packets received from any one client, and the packets received from other clients are dropped, until the connected client is authenticated by the RADIUS server.

9.2.2.2.8.1 Single-host Mode

In single-host mode, once the 802.1X client has been authenticated by the RADIUS server further authentication is not required, but the port accepts packets only from the MAC address of the authenticated client.

9.2.2.2.8.2 Multi-host Mode

In multi-host mode, once the 802.1X client has been authenticated by the RADIUS server, the port is open to accept all packets from any connected client, and these packets do not require any authentication.

9.2.2.2.9 802.1X MAC- Based Authentication

The 802.1X MAC-based authentication allows a set of MAC addresses to be programmed into the RADIUS server. These MAC addresses (MAC-based authentication supplicants) do not connect to 802.1X profiles but

are still allowed access to the network. The authenticator identifies devices that do not support 802.1X and uses the MAC address of these devices as username and password in its RADIUS request packets.

In a MAC-based authentication, every supplicant trying to gain access to the authenticator port is individually authenticated as opposed to authenticating just one supplicant on a given VLAN or port with 802.1X.
The behavior is different for MAC-based authentication supplicants when we have a 802.1.x supplicant authenticated in single host and multi-host 802.1X modes.

To enable Mac-based authentication, use the following command:

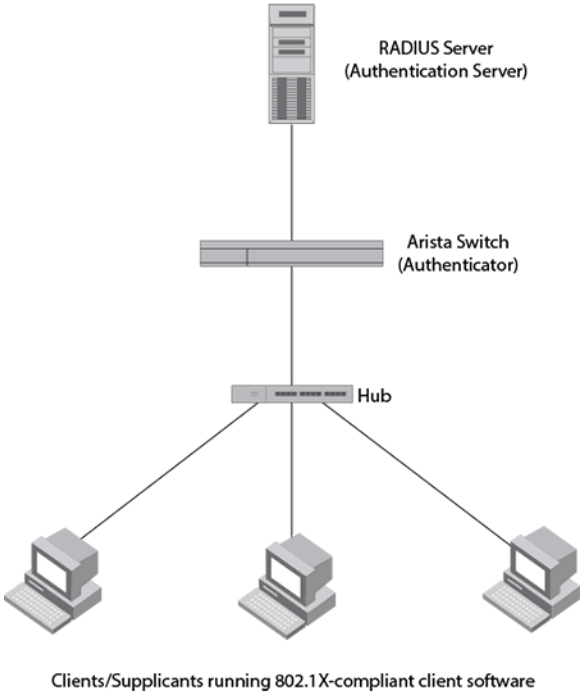
Command syntax

`dot1x mac based authentication`

Note: This command is added to the existing 802.1X configuration on the port, so a typical 802.1X interface configuration with MAC-Based Authentication enabled may look something like this:

```
switch(config-if-Et1/1)# show active
speed forced 1000full
dot1x pae authenticator
dot1x port-control auto
dot1x mac based authentication
```

Figure 9-7: Multiple clients connected to a 802.1X-enabled port



Mac-Based Authentication Delay

Use the **`mac based authentication delay`** command to configure a Mac-based Authentication delay. By default, the delay is triggered after ***5*** seconds.

Comman Syntax

`mac based authentication` delay ***0-300*** seconds

Mac-Based Authentication Hold-Period

When Mac Based Authentication is rejected by a AAA server, there is a default hold period of ***60*** seconds before the Mac Based Authentication is retried again even if the host continues to send traffic. However,

the hold-period can be configured manually using the **mac based authentication hold period** command.

Command Syntax

mac based authentication hold period ***0-300*** seconds

Note: Configuring a low value for the hold-period can significantly increase the load on a AAA server in- case MAC-Based Authentication is not enabled for a host.

9.2.2.3 802.1X AAA Unresponsive VLAN

Overview

Devices connected to 802.1X controlled ports must perform authentication before their generic traffic is allowed into the network. During this process, the switch contacts a configured AAA server that determines if the device's access to the network is accepted or denied. When the AAA server is unresponsive, the default behavior is to deny all authentication attempts. The AAA Unresponsive VLAN feature allows the user to specify different behavior for this case, accepting authentication attempts and assigning devices to the native VLAN or a specified VLAN. As in other failure scenarios, the switch tries to authenticate the supplicant after the quiet period has passed.

9.2.2.3.1 Configuring 802.1X AAA Unresponsive VLAN

The aaa unresponsive action traffic allow vlan command is configured under the dot1x configuration sub-mode to enable the dot1x AAA unresponsive VLAN feature on the switch. When configured, the switch changes the action taken with regards to authentication attempts when the AAA server is unresponsive. The AAA server is considered unresponsive when communication with it times out.

Example

These commands places the switch in the dot1x configuration mode and enables the dot1x AAA unresponsive VLAN feature on the switch.

```
switch(config)# dot1x
switch(config-dot1x)# aaa unresponsive action traffic allow vlan
```

9.2.2.3.2 Limitations

- AAA unresponsive VLAN does not act on devices that tried to authenticate using VLAN-tagged frames.
- When AAA unresponsive VLAN is enabled without a VLAN, devices get assigned to the native VLAN – even phones that would otherwise be assigned to the phone VLAN. If phones should be assigned to the phone VLAN when AAA is unavailable, the knob aaa unresponsive phone vlan action allow should be additionally used.

9.2.2.4 802.1X Web Authentication

802.1X Web authentication feature authenticates a supplicant through a web page, generally referred to as a captive portal, which is why this feature is also known as captive portal authentication. Redirection to captive portal provides support for guest devices/supplicants that neither speak 802.1X nor can be whitelisted in advance or where 802.1X is not sufficient and an additional web based authentication is required.

9.2.2.4.1 Configuring 802.1X Web authentication

A global knob under the 802.1X node is used to enable the 802.1X Web authentication:

Command Syntax

captive portal url ***URL***][ssl profile ***profile***]

Enabling the 802.1X Web authentication starts the redirection agent (Dot1xWeb) and its internal HTTP redirector, and makes 802.1X act on radius web-auth-start VSA's. If a URL is specified, it's used for the redirection when AAA does not provide a specific URL. If a valid SSL profile is specified, the configured certificate and key are used to start 802.1X Web's internal HTTPS redirector.

For ACL based Web Authentication, there is one more configuration knob as follows :

```
switch(config-dot1x)# captive portal access-list ipv4 test-ACL
```

An ACL can be defined locally on the switch and be configured to use for web authentication, for cases, when AAA is not able to send ACL with web auth = start.

Here are the details about the radius VSAs.

AttributeName	Attribute ID	Type	Value
Arista-WebAuth	6	integer	start = 1 complete = 2
Arista-Captive-Portal	10	string	any valid url

Show Commands

The "show" commands that display the state of a host show the new values for WebAuth stage as well.

Example

```
switch(config)# show dot1x hosts
Interface: Ethernet36
Supplicant MAC     Auth Method     State                VLAN Id
--------------     -----------     -----                -------
00:1c:73:73:f9:38  MAC-BASED-AUTH  WEB-AUTH-START
00:1c:73:73:f9:39  MAC-BASED-AUTH  WEB-AUTH-FAILED
```

9.2.2.4.2 Limitations

The following limitations apply to the 802.1X feature.

- Only one device per port is supported (MAC ACLs are not supported), connected in wired fashion.
- HTTPS redirection is only attempted when the connection is to the default TCP port ***443***.
- Limitations present in versions lower than **RIO RELEASE**.
  - HTTPS is not supported.
- Limitations present in versions EOS Release *4.25.0* and *4.25.1*:
  - There is no downloadable ACL support - only implicit ACL support is available. This might not suffice if there is a need to allow multiple intranet websites.
  - There is only support of one Captive portal at a time.
- Limitations in version EOS Release *4.25.0*:
  - IPv4 Management IP needs to be configured on the management interface. If the management ip address is changed, then captive portal configuration needs to be reconfigured.
  - SVI needs to be configured for the VLAN where the host is going to be after the first phase of authentication - be it EAPOL or MBA.

9.2.2.5 Configuring 802.1X Port Security

Basic steps to implementing 802.1X Port-based Network Access Control and RADIUS accounting on the switch:

1. A RADIUS server is required on one or more of your network servers or management stations. 802.1X is not supported with the TACACS+ authentication protocol.
2. You must create supplicant accounts on the RADIUS server:
   - The account for a supplicant connected to an authenticator port must have a username and password combination when set to the 802.1X authentication mode.
     - An account for the supplicant connected to an authenticator port and placed in the MAC address-based authentication mode needs use the MAC address of the node as both the username and password.
     - Connected clients to an 802.1X authenticator port will require 802.1X client software.
3. The RADIUS client must be configured by entering the IP addresses and encryption keys of the authentication servers on your network.
4. The port access control settings must be configured on the switch. This includes the following:
   - Specifying the port roles.
     - Configuring 802.1X port parameters.
       - Enabling 802.1X Port-based Network Access Control.

Guidelines

- Do not set a port that is connected to a RADIUS authentication server to the authenticator role as an authentication server cannot authenticate itself.
- A supplicant connected to an authenticator port set to the 802.1X username and password authentication method must have 802.1X client software.
- To prevent unauthorized individuals from accessing the network through unattended network workstations, end users of 802.1X port-based network access control should always log off when they are finished with a work session.
- The RADIUS client should be configured on the switch before activating port-based access control.

9.2.2.5.1 Configuring 802.1X Authentication Methods

IEEE 802.1X port security relies on external client-authentication methods, which must be configured for use. The method currently supported on Arista switches is RADIUS authentication. To configure the switch to use a RADIUS server for client authentication, use the **aaa authentication dot1x** command.

Example

This command configures the switch to use RADIUS authentication.

```
switch(config)# aaa authentication dot1x default group radius
switch(config)#
```

9.2.2.5.2 Configuring Dot1x Dropped Counters

Use the **statistics packets dropped** command to cofigure the dot1x dropped counters on the switch under ***dot1x*** configuration mode. By default, the dot1x dropped counters is disabled. The **no** form of the command disables the dot1x dropped counters from the running configuration.

Example

These commands places the switch in the ***dot1x*** mode and enables the dot1x dropped counters.

```
switch(config-dot1x)# statistics packets dropped
```

9.2.2.5.3 Globally Enable IEEE 802.1X

To enable IEEE 802.1X port authentication globally on the switch, use the **dot1x system-auth-control** command.

Example

This command enables IEEE 802.1X globally on the switch.

```
switch(config)# dot1x system-auth-control
switch(config)#
```

9.2.2.5.4 Designating Authenticator Ports

To set the port access entity (PAE) type of an Ethernet or management interface to the **authenticator**, use the **dot1x pae authenticator** command.

Example

These commands configure the PAE type to **authenticator** on the Ethernet interface ***1*** to enable IEEE 802.1X on the port.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x pae authenticator
switch(config-if-Et1)#
```

Example

For ports to act as authenticator ports to connected supplicants, those ports must be designated using the **dot1x port-control** command.

The **auto** option of the**dot1x port-control** command designates an authenticator port for immediate use, blocking all traffic that is not authenticated by the AAA server.

Example

This command configures Ethernet ***1*** to immediately begin functioning as an authenticator port.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x port-control auto
switch(config-if-Et1)#
```

The **force-authorized** option of the dot1x port-control command sets the state of the port to **authorized** without authentication, allowing traffic to continue uninterrupted.

Example

These commands designate Ethernet ***1*** as an authenticator port that forwards packets without authentication.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x port-controlforce-authorized
switch(config-if-Et1)#
```

To designate a port as an authenticator but prevent it from authorizing any traffic, use the **force-unauthorized** option of the **dot1x port-control** command.

Example

The **force-unauthorized** option of the **dot1x port-control** command places the specified port in the unauthorized state, which will deny any access requests from users of the ports.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x port-controlforce-authorized
```

```
switch(config-if-Et1)#
```

9.2.2.5.5 Specifying the Authentication Mode for Multiple Clients

By default, Arista switches authenticate in multi-host mode, allowing packets from any source MAC address once 802.1X authentication has taken place. To configure the switch for single-host mode (allowing traffic only from the authenticated clients MAC address), use the **dot1x host-mode** command.

Example

These commands configure Ethernet interface ***1*** to use single-host mode for 802.1X authentication.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x host-mode single-host
switch(config-if-Et1)#
```

9.2.2.5.6 Configuring Re-authentication

The **dot1x reauthentication** command enables re-authentication of authenticator ports with the default values.

The **dot1x timeout reauth-period** command allows to customize the re-authentication period of authenticator ports.

Examples

- These commands configures the configuration mode interface to require re-authentication from clients at regular intervals.

```
switch(config)# interface Ethernet 1
switch(config-if-Eth)# dot1x reauthentication
```

- These commands configure the Ethernet interface ***1*** authenticator to require re-authentication from clients every ***6*** hours (***21600*** seconds).

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x reauthentication
switch(config-if-Et1)# dot1x timeout reauth-period 21600
switch(config-if-Et1)#
```

- These commands deactivate re-authentication on Ethernet interface ***1***.
-

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# no dot1x reauthentication
switch(config-if-Et1)#
```

9.2.2.5.7 Setting the EAP Request Maximum

The **dot1x reauthorization request limit** command configures the number of times the switch retransmits an 802.1X Extensible Authentication Protocol (EAP) request packet before ending the conversation and restarting authentication.

Example

These commands set the number of times the authenticator sends an EAP request packet to the client before restarting authentication.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x reauthorization request limit 4
switch(config-if-Et1)#
```

The default value is ***2***.

9.2.2.5.8 Disabling Authentication on a Port

To disable authentication on an authenticator port, use the **no** form of the **dot1x port-control** command.

Example

These commands disable authentication on Ethernet interface ***1***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# no dot1x port-control
switch(config-if-Et1)#
```

9.2.2.5.9 Setting the Quiet Period

If the switch fails to immediately authenticate the client, the time the switch waits before trying again is specified by the **dot1x timeout quiet-period** command. This timer also indicates how long a client that failed authentication is blocked.

Example

These commands set the 802.1X quiet period for Ethernet interface ***1*** to ***30*** seconds.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x timeout quiet-period 30
```

The default value is ***60*** seconds.

9.2.2.5.10 Setting the Dot1x Timeout Reauth-period

The **dot1x timeout reauth-period** command specifies the time period in seconds that the configuration mode interface waits before requiring re-authentication from clients.

Example

These commands configure the timeout reauth-period to ***21600*** seconds.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x reauthentication
switch(config-if-Et1)# dot1x timeout reauth-period 21600
```

The default value is ***3600*** seconds.

9.2.2.5.11 Setting the Transmission Timeout

Authentication and re-authentication are accomplished by the authenticator sending an Extensible Authentication Protocol (EAP) request to the supplicant and the supplicant sending a reply which the authenticator forwards to an authentication server. If the authenticator doesnt receive a reply to the EAP request, it waits a specified period of time before retransmitting. To configure that wait time, use the **dot1x timeout tx-period** command.

Example

These commands configure Ethernet interface ***1*** to wait ***30*** seconds before retransmitting EAP requests to the supplicant.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x timeout tx-period 30
switch(config-if-Et1)#
```

The default value is ***5*** seconds.

9.2.2.5.12 Enable Authentication Failure VLAN

Configure Authentication Failure VLAN on a dot1x-enabled port using the following CLI command under the ***interface-config*** mode. The CLI command to set ***VLAN10*** as authentication failure VLAN is as follows:

```
switch(config-if-Et1/1)# dot1x authentication failure action traffic allow vlan
 10
```

When **no authentication failure** VLAN is configured on a dot1x-enabled port, the default action is to drop any unauthorized traffic on the port. This behavior can also be specified using the following command:

Example

```
switch(config-if-Et1/1)# dot1x authentication failure action traffic drop
```

9.2.2.5.13 Clearing 802.1X Statistics

The **`clear dot1x statistics`** command resets the 802.1X counters.

Examples

- This command clears the 802.1X counters on all interfaces.

```
switch# clear dot1x statistics all
switch#
```

- This command clears the 802.1X counters on Ethernet interface ***1***.

```
switch# clear dot1x statistics interface ethernet 1
switch#
```

9.2.2.6 Displaying 802.1X Information

You can display information about 802.1X on the switch and on individual ports.

9.2.2.6.1 Displaying 802.1X statistics

Use the **`show dot1x statistics`** command to display 802.1X statistics for the specified port or ports.

Example

- This command displays IEEE 802.1X statistics for Ethernet interface ***5***.

```
switch# show dot1x interface ethernet 5 statistics
Dot1X Authenticator Port Statistics for Ethernet5
-------------------------------------------------
RxStart = 0      RxLogoff = 0    RxRespId = 0
RxResp = 0       RxInvalid = 0   RxTotal = 0
TxReqId = 0      TxReq = 0       TxTotal = 0
RxVersion = 0    LastRxSrcMAC = 0000.0000.0000 switch#
```

- This command displays the dot1x dropped counters for all the dot1x interfaces.

```
switch# show dot1x all statistics
Dot1X Authenticator Port Statistics for Ethernet51/1
-------------------------------------------------
RX start = 1     RX logoff = 0   RX response ID = 1
RX response = 10         RX invalid = 0  RX total = 12
TX request ID = 2        TX request = 11         TX total = 13
```

```
RX version = 2   Last RX src MAC = ded6.404b.ec94
Data packet drop counters:
EAPOL unauthorized port = 2
EAPOL unauthorized host = 1
MBA unauthorized host = 0

Dot1X Authenticator Port Statistics for Ethernet49
--------------------------------------------------
RX start = 1     RX logoff = 0   RX response ID = 1
RX response = 10         RX invalid = 0  RX total = 12
TX request ID = 2        TX request = 11         TX total = 13
RX version = 2   Last RX src MAC = ded6.404b.ec94
Data packet drop counters:
EAPOL unauthorized port = 2
EAPOL unauthorized host = 1
MBA unauthorized host = 0
```

9.2.2.6.2 Displaying 802.1X supplicant information

Use the **show dot1x hosts** command to display information for all the supplicants.

Example

This command displays 802.1X supplicant information.

```
switch# show dot1x hosts
    Interface: Ethernet1/1
    Supplicant MAC       Auth Method      State     VLAN Id
    --------------       -----------      -----     -------
    e2:29:cb:11:2f:4a    EAPOL            SUCCESS   300
    e2:29:cb:11:2f:4b    MAC-BASED-AUTH   SUCCESS   300
```

9.2.2.6.3 Displaying Mac-address Tables

Use the **show mac address-table** command to display the MAC address of the supplicants allowed to pass the traffic through the port.

Example

```
switch# show mac address-table
    Mac Address Table
    ------------------------------------------------------------------

    Vlan    Mac Address       Type        Ports      Moves   Last Move
    ----    -----------       ----        -----      -----   ---------
    300     e229.cb11.2f4a    STATIC      Et1/1
    300     e229.cb11.2f4b    STATIC      Et1/1
    Total Mac Addresses for this criterion: 2
```

9.2.2.6.4 Displaying Port Security Configuration Information

The **show dot1x** command shows information about the 802.1X configuration on the specified port or ports.

Example

This commands displays IEEE 802.1X configuration information for Ethernet interface ***5***.

```
switch# show dot1x interface ethernet 5
Dot1X Information for Ethernet5
-------------------------------------------
PortControl             : auto
```

```
QuietPeriod              : 60 seconds
TxPeriod                 : 5 seconds
ReauthPeriod             : 3600 seconds
MaxReauthReq             : 2 switch#
```

9.2.2.6.5 Displaying the Status of the 802.1X Attributes for each Port

Use the **`show dotx1 interface`** ***interface-id*** command to display the status of the 802x1 attributes for each port.

Example

```
switch(config-if-Et1/1)# show dot1x interface ethernet1/1
       Dot1X Information for Ethernet1
       ---------------------------------------
       PortControl              : force-authorized
       HostMode                 : multi-host
       QuietPeriod              : 60 seconds
       TxPeriod                 : 5 seconds
       ReauthPeriod             : 0 seconds
       MaxReauthReq             : 2
       ReauthTimeoutIgnore      : No
       AuthFailVlan             : 10
```

9.2.2.6.6 Displaying 802.1X Information for all Ports

Use the **show dot1x all brief** command to display IEEE 802.1X status for all ports.

Example

The following commands display a summary of IEEE 802.1X status.

```
switch# show dot1x all brief
Interface   Client   Status
----------  -------- -------------
Ethernet5   None     Unauthorized switch#
```

9.2.2.6.7 Displaying VLANS

Use the **`show vlan`** command to display if a VLAN has been dynamically assigned to the port.

Example

```
switch# show vlan
    VLAN  Name          Status    Ports
    ----- ------------- --------- ---------------------------------
    1     default        active
    2     VLAN0002       active    Et7, Et17, Et18, Et41
    300*  VLAN0300       active    Et1/1, Et6, Et19, Et20, Et29
                                   Et30, Et31, Et32, Et42, Et43, Et44

* indicates a Dynamic VLAN
```

9.2.2.6.8 Displaying EAPOL Fallback to MBA Authentication and MBA Timeout Information

Use the **`show dotx1 interface`** ***interface ID*** **details** command to display information about the EAPOL fallback to MBA authentication and MBA timeout details.

Example

```
switch(config-if-Et1)# show dot1x interface Ethernet1 details
Dot1X Information for Ethernet1
-------------------------------------------
Port control: auto
Host mode: multi-host authenticated
Quiet period: 60 seconds
TX period: 5 seconds
Maximum reauth requests: 2
Ignore reauth timeout: No
Auth failure VLAN: 101
Unauthorized access VLAN egress: Yes
Unauthorized native VLAN egress: Yes
EAPOL: enabled
MAC-based authentication: disabled
EAPOL authentication failure fallback: MBA, timeout 200 seconds

Dot1X Authenticator Client

Port status: Authorized
Supplicant MAC  Reauth Period (in seconds)
--------------  --------------------------
0022.0100.0001  120
```

9.2.2.7 802.1X Port Security Commands

Global Configuration Commands

- dot1x system-auth-control

Dot1x Configuration Commands

- aaa unresponsive action traffic allow vlan
- captive portal
- dot1x mac based authentication delay
- dot1x mac based authentication hold period

Interface Configuration Commands Ethernet Interface

- dot1x host-mode
- dot1x mac based authentication
- dot1x pae authenticator
- dot1x port-control
- dot1x reauthentication
- dot1x reauthorization request limit
- dot1x timeout quiet-period
- dot1x timeout reauth-period
- dot1x timeout tx-period
- statistics packets dropped

Privileged EXEC Commands

- clear dot1x statistics
- show dot1x
- show dot1x all brief
- show dot1x hosts

- show dot1x statistics

9.2.2.7.1 aaa unresponsive action traffic allow vlan

The **aaa unresponsive action traffic allow vlan** enables the the dot1x AAA unresponsive VLAN feature on the switch.

The **no aaa unresponsive action traffic allow vlan** command disbales the dot1x AAA unresponsive VLAN feature from the ***running-config***.

Command Mode

Dot1x Configuration Mode

Command Syntax

aaa unresponsive action traffic allow vlan ***VLAN-ID***

no unresponsive action traffic allow vlan

Parameters

- **unresponsive** Configure AAA timeout options.
- **action** Set action for supplicant when AAA times out.
- **traffic** Set action for supplicant traffic when AAA times out.
- **allow** Allow traffic when AAA times out.
- **vlan** Allow traffic in VLAN when AAA times out.
- ***VLAN-ID*** Identifier for a Virtual LAN. Value ranges from ***1*** to ***4094***.

Example

These commands places the switch in the dot1x configuration mode and enables the dot1x AAA unresponsive VLAN feature on the switch.

```
switch(config)# dot1x
switch(config-dot1x)# aaa unresponsive action traffic allow vlan 50
```

9.2.2.7.2 captive portal

The **captive portal** command enables the 802.1X Web Authentication on the switch.

The **no captive portal** command removes the 802.1X Web Authentication configuration from the ***running-config***.

Command Mode

Dot1x Configuration Mode

Command Syntax

captive portal url ***URL*** ssl profile ***profile*** access-list ipv4 ***ACL name***

no captive portal url ***URL*** ssl profile ***profile*** access-list ipv4 ***ACL name***

Parameters

- **url** Configure captive portal URL.
- **ssl** Configure SSL related option.
- **access-list** Configure access control list.

Example

- This command enables 802.1X Web Authentication on the switch.

```
switch(config)# dot1x
```

```
switch(config-dot1x)# captive portal ssl profile test-ssl_profile
```

- This command enables the ACL based Web authentication.

```
switch(config)# dot1x
switch(config-dot1x)# captive portal access-list ipv4 test-ACL
```

9.2.2.7.3 clear dot1x statistics

The **clear dot1x statistics** command resets the 802.1X counters on the specified interface or all interfaces.

Command Mode

Privileged EXEC

Command Syntax

clear dot1x statistics **INTERFACE_NAME**

Parameters

INTERFACE_NAME Interface type and number. Options include:

- **all** Display information for all interfaces.
- **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **interface loopback *l_num*** Loopback interface specified by***l_num***.
- **interface management *m_num*** Management interface specified by ***m_num***.
- **interface port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **interface vlan *v_num*** VLAN interface specified by ***v_num***.

Example

This command resets the 802.1X counters on all interfaces.

```
switch# clear dot1x statistics all
switch#
```

9.2.2.7.4 dot1x mac based authentication

The **dot1x mac based authentication** command enables MAC-based authentication on the existing 802.1X authenticator port.

The **no dot1x mac based authentication** and the **default dot1x mac based authentication** commands restore the switch default by disabling the corresponding dot1x mac based authentication command for the specific 802.1X authenticator port.

Command Mode

Interface-Ethernet Configuration

Command Syntax

dot1x mac based authentication

no dot1x mac based authentication

default dot1x mac based authentication

Related Command

show dot1x hosts

Example

These commands configure MAC-based authentication on ***Ethernet interface 1***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x mac based authentication
switch(config-if-Et1)#
```

9.2.2.7.5 dot1x mac based authentication delay

The **dot1x mac based authentication delay** command enables MAC-based authentication delay. By default, the delay is triggered after ***5*** seconds.

The **no dot1x mac based authentication delay** and the **default dot1x mac based authentication delay** commands restore the switch default by disabling the corresponding dot1x mac based authentication delay command.

Command Mode

Dot1x Configuration

Command Syntax

dot1x mac based authentication delay *delay-time* seconds

no dot1x mac based authentication delay

default dot1x mac based authentication delay

Parameter

- ***delay-time*** Delay in seconds. The value is from ***0*** to ***300***.
- **seconds** Unit in seconds.

Example

These commands configure a MAC-based authentication delay of ***30*** seconds on a switch.

```
switch(config)# dot1x
switch(config-dot1x)# mac based authentication delay 30 seconds
```

9.2.2.7.6 dot1x mac based authentication hold period

The **dot1x mac based authentication hold period** command enables MAC-based authentication hold period. By default, the hold period is ***60*** seconds.

The **no dot1x mac based authentication hold period** and the **default dot1x mac based authentication hold period** commands restore the switch default by disabling the corresponding dot1x mac based authentication hold period command.

Command Mode

Dot1x Configuration

Command Syntax

dot1x mac based authentication hold period ***hold period-time*** seconds

no dot1x mac based authentication hold period

default dot1x mac based authentication hold period

Parameter

- ***hold period-time*** Hold period in seconds. The value is from ***1*** to ***300*** in seconds.
- **seconds** Unit in seconds.

Example

These commands configure a MAC-based authentication hold period of ***100*** seconds on a switch.

```
switch(config)# dot1x
switch(config-dot1x)# mac based authentication hold period 100 seconds
```

9.2.2.7.7 dot1x pae authenticator

The **dot1x pae authenticator** command sets the port access entity (PAE) type of the configuration mode interface to **authenticator**, which enables IEEE 802.1X on the port. IEEE 802.1X is disabled on all ports by default.

The **no dot1x pae authenticator** and **default dot1x pae authenticator** commands restore the switch default by deleting the corresponding **dot1x pae authenticator** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

dot1x pae authenticator

no dot1x pae authenticator

default dot1x pae authenticator

Examples

- These commands configure ***interface ethernet 2*** as a port access entity (PAE) authenticator, enabling IEEE 802.1X on the port.

```
switch(config-if-Et1)# interface ethernet 2
switch(config-if-Et1)# dot1x pae authenticator
switch(config-if-Et1)#
```

- These commands disable IEEE 802.1X authentication on ***interface ethernet 2***.

```
switch(config-if-Et1)# interface ethernet 2
switch(config-if-Et1)# no dot1x pae authenticator
switch(config-if-Et1)#
```

9.2.2.7.8 dot1x reauthentication

The **dot1x reauthentication** command configures the configuration mode interface to require re-authentication from clients at regular intervals. The interval is set by the **dot1x timeout reauth-period** command.

The **no dot1x reauthentication** and **default dot1x reauthentication** commands restore the default setting by deleting the corresponding **dot1x reauthentication** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

dot1x reauthentication

no dot1x reauthentication

default dot1x reauthentication

Example

These commands configure the ***interface Ethernet 1*** authenticator to require periodic re-authentication from clients.

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x reauthentication
switch(config-if-Et1)#
```

9.2.2.7.9 dot1x reauthorization request limit

The **dot1x reauthorization request limit** command configures how many times the switch retransmits an 802.1X Extensible Authentication Protocol (EAP) request packet before ending the conversation and restarting authentication.

The **no dot1x reauthorization request limit** and **default dot1x reauthorization request limit** commands restore the default value of 2 by deleting the corresponding **dot1x reauthorization request limit** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

dot1x reauthorization request limit attempts

no dot1x reauthorization request limit

default dot1x reauthorization request limit

Parameters

attempts Maximum number of attempts. Values range from ***1*** to ***10***; default value is ***2***.

Examples

- This command sets the 802.1X EAP-request retransmit limit to ***6***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x reauthorization request limit 6
switch(config-if-Et1)#
```

- This command restores the default request repetition value of ***2***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# no dot1x reauthorization request limit
switch(config-if-Et1)#
```

9.2.2.7.10 dot1x system-auth-control

The **dot1x system-auth-control** command enables 802.1X authentication on the switch.

The **no dot1x system-auth-control** and **default dot1x system-auth-control** commands disables 802.1X authentication by removing the **dot1x system-auth-control** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

dot1x system-auth-control

no dot1x system-auth-control

default dot1x system-auth-control

Examples

- This command enables 802.1X authentication on the switch.

```
switch(config)# dot1x system-auth-control
switch(config)#
```

- This command disables 802.1X authentication on the switch.

```
switch(config)# no dot1x system-auth-control
switch(config)#
```

9.2.2.7.11 dot1x timeout quiet-period

If the switch fails to immediately authenticate the client, the time the switch waits before trying again is specified by the **dot1x timeout quiet-period** command. This timer also indicates how long a client that failed authentication is blocked.

The **no dot1x timeout quiet-period** and **default dot1x timeout quiet-period** commands restore the default quiet period of ***60*** seconds by removing the corresponding **dot1x timeout quiet-period** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

dot1x timeout quiet-period ***quiet_time***

no dot1x timeout quiet-period

default dot1x timeout quiet-period

Parameter

quiet_time Interval in seconds. Values range from ***1*** to ***65535***. Default value is ***60***.

Example

These commands set the 802.1X quiet period for Ethernet interface ***1*** to ***30*** seconds.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x timeout quiet-period 30
switch(config-if-Et1)#
```

9.2.2.7.12 dot1x timeout reauth-period

The **dot1x timeout reauth-period** command specifies the time period that the configuration mode interface waits before requiring re-authentication from clients.

The **no dot1x timeout reauth-period** and **default dot1x timeout reauth-period** commands restore the default period of ***60*** minutes by removing the corresponding **dot1x timeout reauth-period** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

dot1x timeout reauth-period ***reauth_time***

no dot1x timeout reauth-period

default dot1x timeout reauth-period

Parameter

reauth_time The number of seconds the interface passes traffic before requiring re-authentication. Values range from ***1*** to ***65535***. Default value is ***3600***.

Example

These commands configure theinterface Ethernet ***1*** authenticator to require re-authentication from clients every ***6*** hours (***21600*** seconds).

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x reauthentication
switch(config-if-Et1)# dot1x timeout reauth-period 21600
switch(config-if-Et1)#
```

9.2.2.7.13 dot1x timeout tx-period

Authentication and re-authentication are accomplished by the authenticator sending an Extensible Authentication Protocol (EAP) request to the supplicant and the supplicant sending a reply which the authenticator forwards to an authentication server. If the authenticator does not get a reply to the EAP request, it waits a specified period of time before retransmitting. The **dot1x timeout tx-period** command configures that wait time.

The **no dot1x timeout tx-period** and **default dot1x timeout tx-period** commands restore the default wait time by removing the corresponding **dot1x timeout tx-period** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

dot1x timeout tx-period ***tx_time***

no dot1x timeout tx-period

default dot1x timeout tx-period

Parameter

tx_time Values range from ***1*** to ***65535***. Default value is ***5***.

Example

These commands configure interface Ethernet ***1*** to wait ***30*** seconds before retransmitting EAP requests to the supplicant.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x timeout tx-period 30
switch(config-if-Et1)#
```

9.2.2.7.14 dot1x host-mode

When multiple clients are connected to an Ethernet interface providing 802.1X authentication, the port can accept packets from all MAC addresses once the supplicant has been authenticated (multi-host mode), or it can accept only those packets originating from the MAC address of the authenticated client (single-host mode) or ultiple authenticated clients (multi-host authenticated mode) . The **dot1x host-mode** command specifies the host mode for authentication of multiple clients on the configuration mode interface.

The **no dot1x host-mode** and **default dot1x host-mode** commands restore the switch default (multi-host mode) by removing the corresponding dot1x host-mode command for the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

dot1x host-mode [multi-host | single-host | multi-host authenticated]

no dot1x host-mode

default dot1x host-mode

Parameters

- **multi-host** Configures the interface to use multi-host mode (the default).
- **single-host** Configures the interface to use single-host mode.
- **multi-host authenticated** Configures the interface to use multi-host authenticated mode.

Example

These commands configure ***interface Ethernet 1*** to use single-host mode for 802.1X authentication.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# dot1x host-mode single-host
switch(config-if-Et1)#
```

9.2.2.7.15 dot1x port-control

The **dot1x port-control** command configures the configuration mode interface as an authenticator port and specifies whether it will authenticate traffic.

The **no dot1x port-control** and **default dot1x port-control** commands configure the port to pass traffic without authorization by removing the corresponding **dot1x port-control** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

dot1x port-control STATE

no dot1x port-control

default dot1x port-control

Parameters

STATE Specifies whether the interface will authenticate traffic. The default value is **force-authorized**. Options include:

- **auto** Configures the port to authenticate traffic using Extensible Authentication Protocol messages.
- **force-authorized** Configures the port to pass traffic without authentication.

- **force-unauthorized** Configures the port to block all traffic regardless of authentication.

Examples

- These commands configure ***interface Ethernet 1*** to pass traffic without authentication. This is the default setting.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x port-control force-authorized
switch(config-if-Et1)#
```

- These commands configure ***interface Ethernet 1*** to block all traffic.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x port-control force-unauthorized
switch(config-if-Et1)#
```

- These commands configure ***interface Ethernet 1*** to authenticate traffic using EAP messages.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# dot1x port-control auto
switch(config-if-Et1)#
```

9.2.2.7.16 show dot1x all brief

The **`show dot1x all brief`** command displays the IEEE 802.1X status for all ports.

Command Mode

EXEC

Command Syntax

`show dot1x all brief`

Example

This command displays the IEEE 802.1X status.

```
switch# show dot1x all brief
Interface           Client          Status
-----------------------------------------------
Ethernet5           None            Unauthorized switch#
```

9.2.2.7.17 show dot1x hosts

The **`show dot1x hosts`** command displays 802.1X information for all the supplicants.

Command Mode

EXEC

Command Syntax

`show dot1x hosts` [ethernet]

Parameter

ethernet *e_num* Ethernet interface specified by ***e_num***.

Related Command

`dot1x mac based authentication`

Example

This command displays 802.1X information for all the supplicants.

```
switch# show dot1x hosts
Interface: Ethernet1/1
    Supplicant MAC       Auth Method       State      VLAN Id
    --------------       -----------       -----      -------
    e2:29:cb:11:2f:4a    MAC-BASED-AUTH    SUCCESS     300
```

9.2.2.7.18 show dot1x statistics

The **show dot1x statistics** command displays 802.1X statistics for the specified port or ports.

Command Mode

EXEC

Command Syntax

show dot1x **INTERFACE_NAME** statistics

Parameters

- **INTERFACE_NAME** Interface type and number. Options include:
  - **all** Display information for all interfaces.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  -
- **Output Fields**
  - **RxStartNumber** of EAPOL-Start frames received on the port.
  - **TxReqIdNumber** of EAP-Request/Identity frames transmitted on the port.
  - **RxVersionVersion** number of the last EAPOL frame received on the port.
  - **RxLogoffNumber** of EAPOL-Logoff frames received on the port.
  - **RxInvalidNumber** of invalid EAPOL frames received on the port.
  - **TxReqNumber** of transmitted EAP-Request frames that were not EAP-Request/Identity.
  - **LastRxSrcMAC** The source MAC address in the last EAPOL frame received on the port.
  - **RxRespId** The number of EAP-Response/Identity frames received on the port.
  - **RxTotal** The total number of EAPOL frames transmitted on the port.
  - **TxTotal** The total number of EAPOL frames transmitted on the port.

Example

This command displays the 802.1X statistics for ***interface ethernet 5***.

```
switch# show dot1x interface ethernet 5 statistics
Dot1X Authenticator Port Statistics for Ethernet5
-------------------------------------------------
RxStart = 0      RxLogoff = 0    RxRespId = 0
RxStart= 0       RxInvalid = 0   RxTotal = 0
TxReqId = 0      TxReq = 0       TxTotal = 0
RxVersion = 0    LastRxSrcMAC = 0000.0000.0000 switch#
```

9.2.2.7.19 show dot1x

The **show dot1x** command displays 802.1X information for the specified interface.

Command Mode

EXEC

Command Syntax

`show dot1x` **INTERFACE_NAME INFO**

Parameters

- **INTERFACE_NAME** Interface type and number. Options include:
  - **all** Display information for all interfaces.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
- **INFO** Type of information the command displays. Values include:
  - ***no parameter*** displays summary of the specified interface.
  - **detail** displays all 802.1X information for the specified interface.

Examples

- This command displays 802.1X summary information for ***interface ethernet 5***.

```
switch# show dot1x interface ethernet 5
Dot1X Information for Ethernet5
-----------------------------------------
PortControl             : auto
QuietPeriod             : 60 seconds
TxPeriod                : 5 seconds
ReauthPeriod            : 3600 seconds
MaxReauthReq            : 2 switch#
```

- This command displays detailed 802.1X information for ***interface ethernet 5***.

```
switch# show dot1x interface ethernet 5 detail
Dot1X Information for Ethernet5
-----------------------------------------
PortControl             : auto
QuietPeriod             : 60 seconds
TxPeriod                : 5 seconds
ReauthPeriod            : 3600 seconds
MaxReauthReq            : 2

Dot1X Authenticator Client

Port Status             : Unauthorized switch#
```

9.2.2.7.20 statistics packets dropped

The **`statistics packets dropped`**command to cofigure the dot1x dropped counters on the switch under ***dot1x*** configuration mode. By default, the dot1x dropped counters is disabled. The **no** form of the command disables the dot1x dropped counters from the running configuration.

The **`no statistics packets dropped`** command disables the dot1x dropped counters from the running configuration.

Command Mode

Dot1x Configuration

Command Syntax

`statistics packets dropped`

`no statistics packets dropped`

Example

These commands places the switch in the ***dot1x*** mode and enables the dot1x dropped counters.

```
switch(config-dot1x)# statistics packets dropped
```

9.3 Data Plane Security

This section contains the following topics:

- IP NAT
- Media Access Control Security
- Internet Protocol Security (IPsec)
- Macro-Segmentation Service (CVX)

9.3.1 IP NAT

Network Address Translation (NAT) is a router process that modifies address information of IP packets in transit. NAT is typically used to correlate address spaces between a local network and a remote, often public, network. Static NAT defines a one-to-one map between local and remote IP addresses. Static maps are configured manually through CLI commands. An interface can support multiple NAT commands, but each command must specify a unique local IP address-port location.

NAT is configured on routers that have interfaces connecting to the local networks and interfaces connecting to a remote network.

Inside and Outside Addresses

In NAT configurations, IP addresses are placed into one of two categories: inside or outside. Inside refers to IP addresses used within the organizational network. Outside refers to addresses on an external network outside the organizational network.

9.3.1.1 Static IP NAT

Static NAT configurations create a one-to-one mapping and translate a particular address to another address. This type of configuration creates a permanent entry in the NAT table as long as the configuration is present, and it enables both inside and outside hosts to initiate a connection.

Static NAT options include source NAT, destination NAT, and twice NAT.

- Source NAT modifies the source address in the IP header of a packet exiting the interface, and can optionally change the source port referenced in the TCP/UDP headers.
- Destination NAT modifies the destination address in the IP header of a packet entering the interface, and can optionally change the destination port referenced in the TCP/UDP headers.
- Twice NAT modifies both the source and destination address of packets entering and exiting the interface, and can optionally change the L4 port information in the TCP/UDP headers. Twice NAT is generally used when inside network addresses overlap or otherwise conflict with outside network addresses. When a packet exits the interface, local source and destination addresses are translated to global source and destination addresses. When a packet enters the interface, global source and destination addresses are translated to local source and destination addresses.

9.3.1.1.1 Configuring Static NAT

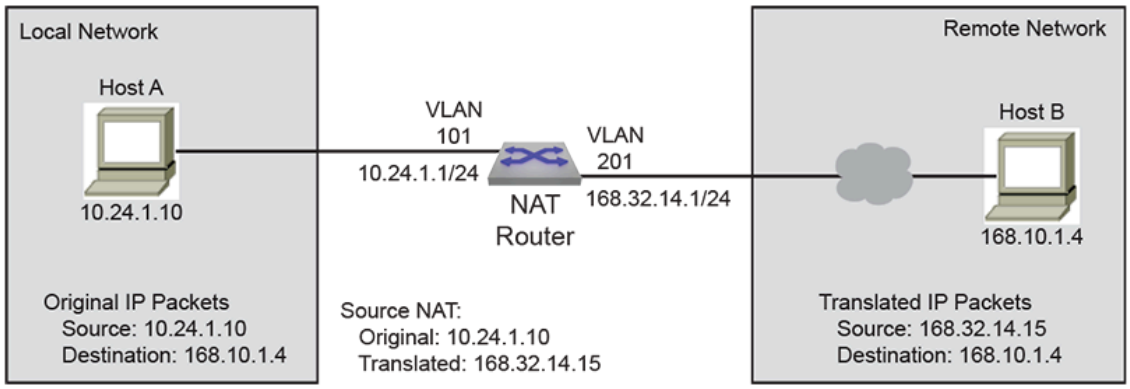
Configuring Source NAT

Network address translation of a source address (source NAT) is enabled by the ip nat source static command for the configuration mode interface. Applying source NAT to interfaces that connect to local hosts shields the IP address of the host when sending IP packets to remote destinations.

This command installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

Note: The switch uses a common NAT table for the entire switch, not a per interface one. For example, if a customer has the same inside local address translated to different inside global addresses depending on which interface it exits. It might be translated to exit interface B's inside global address even though it exits through interface A. A way to avoid this is to use an access list that differentiates based on the destination IP address.

Figure 9-8: Source NAT Example



Example

These commands configure ***VLAN 201*** to translate source address ***10.24.1.10*** to ***168.32.14.15***.

```
switch(config)# interface vlan 201
switch(config-if-Vl201)# ip nat source static 10.24.1.10 168.32.14.15
switch(config-if-Vl201)#
```

The **`ip nat source static`** command may include an ACL to limit packet translation. Only packets whose source IP address matches the ACL are cleared. ACLs configured for source NAT must specify a source IP address of **any**. Source port or protocol matching is not permitted. The destination may be an IP subnet. Commands referencing nonexistent ACLs are accepted by the CLI but not installed in hardware until the ACL is created. Modifying a referenced ACL causes the corresponding hardware entries to be replaced by entries that match the new command.

Example

These commands configure ***VLAN 101*** to translate the source address ***10.24.1.10*** to ***168.32.14.15*** for all packets with IP destination addresses in the ***168.10.1.1/32*** subnet.

```
switch(config)# ip access-list ACL1
switch(config-acl-ACL1)# permit ip any 168.10.1.0/24
switch(config-acl-ACL1)# exit
```

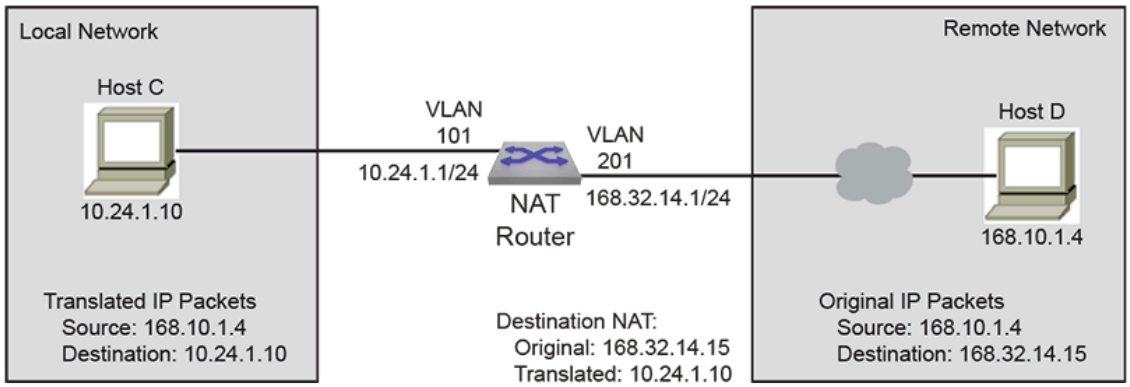
```
switch(config)# interface vlan 101
switch(config-if-Vl101)# ip nat source static 10.24.1.10 access-list
 ACL1 168.32.14.15
switch(config-if-Vl101)#
```

Configuring Destination NAT

Network address translation of a destination address (destination NAT) is enabled by the ip nat destination static command for the configuration mode interface. Applying destination NAT to interfaces that connect to remote hosts shields the IP address of the recipient host when receiving IP packets from remote destinations.

This command installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

Figure 9-9: Detination NAT Example



Example

These commands configure ***VLAN 201*** to translate destination address ***168.32.14.15*** to ***10.24.1.10***.

```
switch(config)# interface vlan 201
switch(config-if-Vl201)# ip nat destination static 168.32.14.15
 10.24.1.10
switch(config-if-Vl201)#
```

The ip nat destination static command may include an ACL to limit packet translation. Only packets whose source IP address matches the ACL are cleared. ACLs configured for destination NAT must specify a destination IP address of **any**. Destination port or protocol matching is not permitted. The source may be an IP subnet. Commands referencing nonexistent ACLs are accepted by the CLI but not installed in hardware until the ACL is created. Modifying a referenced ACL causes the corresponding hardware entries to be replaced by entries that match the new command.

Example

These commands configure ***VLAN 201*** to translate the destination address ***168.32.14.15*** to ***10.24.1.10*** for all packets with IP source addresses in the ***168.10.1.4/32*** subnet.

```
switch(config)# ip access-list ACL2
switch(config-acl-ACL2)# permit ip 168.10.1.4/32 any
```

```
switch(config-acl-ACL2)# exit
switch(config)# interface vlan 201
switch(config-if-Vl201)# ip nat destination static 168.32.14.15
 access-list ACL2
10.24.1.10
switch(config-if-Vl201)#
```

Configuring Twice NAT

Network address translation of both source and destination addresses on the same interface (twice NAT) is enabled by creating one source NAT rule and one destination NAT rule on the same interface and associating them through a NAT group using the ip nat source static and ip nat destination static commands.

The **ip nat source static** command translates the actual local source address to a source address which can be used outside the local network to reference the source. The **ip nat destination static** command translates an internally used destination address to the actual IP address that is the destination of the packet.

The source and destination NAT rules must reference the same NAT group, and both should either specify only IP addresses or specify both IP addresses and L4 port information. If L4 port information is configured in one rule but not in the other, an error message will be displayed.

Each NAT rule installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. Twice NAT does not support the use of access control lists to filter packets for translation.

Example

These commands configure ***interface ethernet 2*** to translate the local source address ***10.24.1.10*** to the global source address ***168.32.14.15***, and to translate the local destination address ***10.68.104.3*** to the global destination address ***168.25.10.7*** for all packets moving through the interface. The use of NAT ***group 3*** is arbitrary, but must be the same in both rules.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# ip nat source static 10.24.1.10 168.32.14.15
 group 3
switch(config-if-Et2)# ip nat destination static 10.68.104.3
 168.25.10.7 group 3
```

9.3.1.1.2 Static NAT Configuration Considerations

Egress VLAN Filter for Static NAT

When a static source NAT is configured on an interface, the source IP translation happens only for those packets that is going 'out' of this interface. If a packet is egressing on an interface which does not have NAT configured, then the source IP is not translated.

When there are two interfaces on which static SNAT is configured then the translation specified for one interface can be applied to a packet going out on the other interface.

Examples

- In this example, the packets with source IP ***20.1.1.1*** going out of ***E1*** will still have the source IP translated to ***172.1.1.1*** even though the rule is configured in ***E2*** and not on ***E1***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# ip nat source static 10.1.1.1 171.1.1.1
switch(config)# interface ethernet 2
switch(config-if-Et2)# ip nat source static 20.1.1.1 172.1.1.1
```

- To prevent this, use an ACL to filter the traffic that needs NAT on the interfaces.

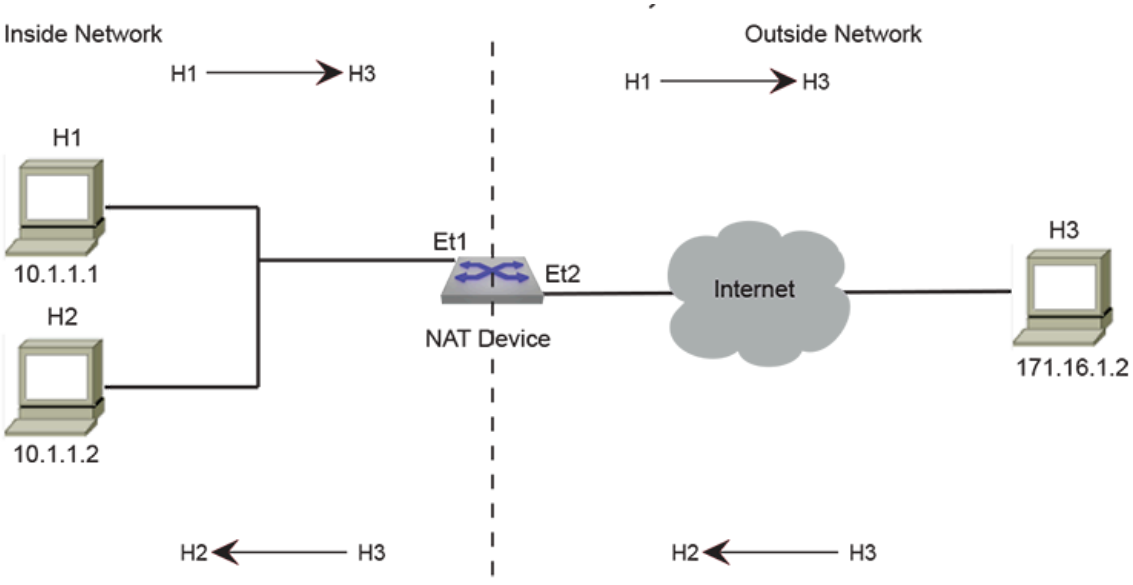
```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# permit ip any 171.1.1.0/24
switch(config)# ip access-list acl2
switch(config-acl-acl2)# permit ip any 172.1.1.0/24
switch(config)# interface ethernet 1
switch(config-if-Et1)# ip nat source static 10.1.1.1 access-list
 acl1 171.1.1.1
switch(config)# interface ethernet 2
switch(config-if-Et2)# ip nat source static 20.1.1.1 access-list
 acl2 172.1.1.1
```

- ACL filtering is not supported when using twice NAT.

9.3.1.2 Dynamic NAT

Dynamic NAT can be used when fewer addresses are accessible than the number of hosts to be translated. A NAT table entry is created when the host starts a connection and establishes a one-to-one mapping between addresses. The mapping can vary and is dependent upon the registered addresses in the pool at the time of the communication. Dynamic NAT sessions are only allowed to be initiated only from inside networks. NAT should be configured on a Layer 3 interface, either a routed port or Switch Virtual Interface (SVI). If the host doesn't communicate for a specific period, dynamic NAT entries are removed from the translation table. The address will then returned to the pool for use by another host

Figure 9-10: Dynamic NAT Scenario



Dynamic NAT options:

- Many-to-Many NAT

Maps local addresses to a global address that is selected from a pool of global addresses. After pool is configured, the first available address from the pool is picked dynamically on receiving the first packet.

- Many-to-One NAT (PAT)

  PAT is a form of dynamic NAT where multiple local addresses are mapped to a single global address (many-to-one) using different source ports. This method is also called NAT Overloading, Network and Port address translation (NAPT), and Masquerade. The global address can be the IP address configured on the outside interface.

  Hardware entries that translate packets are created when the CLI command is processed. Entries for forward and reverse traffic are created for unicast traffic. The hardware entry for reverse traffic is not created for multicast traffic.

  Commands may include ACLs to filter packets that are cleared. Source NAT use ACLs to filter packets based on destination IP address. Destination NAT use ACLs to filter packets based on source IP address. Upon using NAT, inside usually refers to a private network while outside usually refers to a public network.

  A switch with NAT configured translates forwarded traffic between inside and outside interfaces, and the flow that matches the criteria specified for translation.

  The same IP address can't be used for the NAT static configuration and in the pool for dynamic NAT configurations. Public IP addresses must be unique. The global addresses used in static translations aren't excluded with dynamic pools containing the same global addresses.

  Hardware entries that translate packets are created when the CLI command is processed. Entries for forward and reverse traffic are created for unicast traffic. The hardware entry for reverse traffic is not created for multicast traffic.

  Commands may include ACLs to filter packets that are cleared. Source NAT use ACLs to filter packets based on destination IP address. Destination NAT use ACLs to filter packets based on source IP address. When using NAT, inside usually refers to a private network while outside usually refers to a public network.

  A switch with NAT configured translates forwarded traffic between inside and outside interfaces, and the flow that matches the criteria specified for translation.



Note: The same IP address can't be used for the NAT static configuration and in the pool for dynamic NAT configurations. Public IP addresses must be unique. The global addresses used in static translations aren't excluded with dynamic pools containing the same global addresses.



Note: Dynamic NAT with ACL destination port is not supported on 7050SX3.

9.3.1.2.1 Configuring Dynamic NAT

Prerequisites

- Configure an ACL to specify IP addresses allowed to be translated.
- Determine if you should use an IP address as the translated source address.
- Decide on a public IP address pool for address translation.

Configure the Address Pool

The addresses used for translation are configured by issuing the ip nat pool command in global configuration mode.

Example

This command configures the pool of addresses using start address, and end address.

```
switch(config)# ip nat pool p1 10.15.15.15 10.15.15.25
```

```
switch(config)#
```

Set the IP Address

The ip address command configures ***VLAN 201*** with an IP address.

Examples

- This command configures an IPv4 address for ***VLAN 201***.

```
switch(config)# interface vlan 201
switch(config-if-Vl201)# ip address 10.0.0.1/24
switch(config-if-Vl201)#
```

- This command configures the dynamic NAT source address and sets the NAT overload for pool ***P2***.

```
switch(config-if-Vl201)# ip nat source dynamic access-list ACL2 pool
 p2
switch(config-if-Vl201)#
```

Configuring Dynamic NAT Priority

For each Dynamic NAT configuration, you can specify the priority from lowest to highest in an interface mode. The ip nat source dynamic command allows you to configure dynamic NAT priority from the source IP address. Multiple dynamic NAT configurations have the same priority irrespective of the order. If priority is not specified in NAT rule, by default the priority is ***0*** (lowest priority).

Service FTP dynamic NAT rules with a single IP in the pool are considered to be of highest priority.

Note: Priorities in address-only and non-address-only NAT rules are independent of each other.

Example

This command configures the dynamic NAT priority of the access-list in the pool with the order ***a5 > a4 > a3 > a2 > a1 > a0***.

```
switch(config)# interface vlan 201
switch(config-if-Vl201)# ip address 10.0.0.1/24
switch(config-if-Vl201)# ip nat source dynamic access-list a0 pool p0
switch(config-if-Vl201)# ip nat source dynamic access-list a1 pool p1
 priority 1
switch(config-if-Vl201)# ip nat source dynamic access-list a2 pool p2
 priority 2
switch(config-if-Vl201)# ip nat source dynamic access-list a3 pool p3
 priority 3
switch(config-if-Vl201)# ip nat source dynamic access-list a4 pool p4
 priority 4
switch(config-if-Vl201)# ip nat source dynamic access-list a5 pool p5
 priority 5
switch(config-if-Vl201)#
```

Configuring Dynamic NAT with Overload

The following configures a dynamic NAT profile with overload.

Example

This command configures the dynamic NAT for overload.

```
ip nat profile patName
   ip nat source dynamic access-list accessList1 overload
!

ip access-list accessList1
   20 permit ip host 1.1.1.2 any log
```

Define the NAT Source Address for Translation

The ip nat source dynamic command specifies a dynamic translation from the source IP address to the pool and to overload the pool address (or addresses).

Example

This command configures the dynamic NAT source address and sets the pool ***P2*** NAT overload.

```
switch(config)# interface ethernet 3/1
switch(config-if-Et3/1)# ip nat source dynamic access-list ACL2 pool
 p2
switch(config-if-Et3/1)#
```

Specify the Timeout Values

The ip nat translation tcp-timeout or ip nat translation udp-timeout commands alter the translation timeout period for NAT translation table entries.

Examples

- This command globally sets the timeout for TCP to **600** seconds.

```
switch(config)# ip nat translation tcp-timeout 600
 switch(config)#
```

- This command globally sets the timeout for UDP to **800** seconds.

```
switch(config)# ip nat translation udp-timeout 800
 switch(config)#
```

9.3.1.2.2 Verify the NAT Configuration

Display the Address Pools

The show ip nat pool command displays the configuration of the address pool.

Example

This command displays all the address pools configured on the switch.

```
switch# show ip nat pool

Pool      StartIp           EndIp           Prefix
p1        10.15.15.15       10.15.15.25     24
p2        10.10.15.15       10.10.15.25     22
p3        10.12.15.15       10.12.15.25     12 switch#
```

9.3.1.2.3 Clearing IP NAT Table Entries

Use the clear ip nat flow translation command to remove all or the specified NAT table entries.

Example

This command clears all dynamic entries from the NAT table.

```
switch# clear ip nat flow translation
switch#
```

9.3.1.2.4 Dynamic NAT Configuration Considerations

Configuring Dynamic NAT Using Pools in a L2 Adjacent Network

When many-to-one dynamic NAT is configured using a NAT pool, and the next hop router for the NAT device is on the same network (L2 adjacent), then you must configure the IP addresses in the NAT pool as a secondary address on the interface.

Example:

The IP addresses in the NAT pool are configured as the secondary address on the interface.

```
switch(config)# ip nat pool p1 10.1.1.1 10.1.1.4 prefix-length 24
switch(config)# interface ethernet 1
switch(config-if-Et1)# ip nat source dynamic access-list a1 pool p1
switch(config-if-Et1)# ip address 10.1.1.1/24 secondary
switch(config-if-Et1)# ip address 10.1.1.2/24 secondary
switch(config-if-Et1)# ip address 10.1.1.3/24 secondary
switch(config-if-Et1)# ip address 10.1.1.4/24 secondary
```

Configuring Dynamic NAT Using Pool in a L3 Network

If the next hop of the NAT device is on a different subnet, then you should configure a static Null route for the IP addresses in the NAT pool. Redistribute the static route using BGP/OSPF.

Examples

- Outside Interface

```
switch(config)# interface port-channel 319
switch(config-if-Po319)# ip nat source dynamic access-list dynamic-
nat-m2m pool
```

```
natpl-dynamic-nat-m2m
switch(config)# ip access-list dynamic-nat-m2m
switch(config-acl-dynamic-nat-m2m)# 10 permit ip 192.168.93.0/24 any
switch(config)# ip nat pool natpl-dynamic-nat-m2m prefix-length 24
switch(config-natpool-p1)# range 11.3.3.2 11.3.3.10
```

- Static Null Route for Virtual IP

```
switch(config)# ip route 11.0.0.0/8 Null0
switch(config)# router ospf 1
switch(config-router-ospf)# redistribute static
```

Configuring Dynamic NAT Using Overload with ECMP Routes

Dynamic many-to-one NAT using overload (PAT) should not be configured on interfaces that form an ECMP group. When one interface in the group goes down, the return packet for connections that are already established will continue to go to the IP address of the interface that went down and will not be forwarded to the inside host. For this type of scenario, use Dynamic NAT with pool configurations.

9.3.1.2.5 Dynamic NAT Peer State Synchronization

The NAT peer state synchronization provides redundancy and resiliency for dynamic NAT across a pair of devices to avoid single NAT device failure. Both devices in redundant pair are active and they track new sessions and create or delete NAT entries dynamically. Essentially, an active NAT entry is maintained on both devices irrespective of who created the NAT entry.

Configuring Dynamic NAT Peer State Synchronization

The following prerequisites should be fulfilled before configuring NAT peer state synchronization on devices in a redundant pair.

- Both devices in redundant pair must be reachable across an IP address within the same subnet.
- NAT version on both devices in redundant pair must be compatible.
- Dynamic NAT configuration must be identical across both devices in redundant pair.

The following configuration output indicates a valid running configuration of the NAT peer state synchronization on one device.

```
ip nat pool POOL61 prefix-length 24
  range 170.24.0.2 170.24.0.200

ip access-list NatACL61
  10 permit ip 61.0.0.0/16 any


interface Port-Channel5
  mtu 9214
  no switchport
  ip address 10.0.0.1/31
  ip nat source dynamic access-list NatACL61 pool POOL61


ip nat synchronization
  peer-address 11.11.11.1
  local-interface Vlan1111
  port-range 1024 2048
```

The following limitations are applicable during NAT peer state synchronization.

- While configuring dynamic NAT peer state synchronization across peer switches, the port range values of the switches should always be disjoint to avoid virtual IP conflict.
- NAT peer state synchronization does not support asymmetrical TCP setup (SYN - SYNACK - ACK should always be hashed to the same peer.)
- The connection is only synchronized with a peer if the TCP state is established.

The following command specifies the description of the device itself.

```
switch(config)#ip nat synchronization
switch(config-nat-synchronization)#description <description>
```

The following command specifies seconds the switch needs to wait before timing out existing connections.

```
switch(config)#ip nat synchronization
switch(config-nat-synchronization)#expiry-interval 6
```

The following command specifies seconds the IP address of the peer device from where the synchronization is coming.

```
switch(config)#ip nat synchronization
switch(config-nat-synchronization)#peer address 202.1.1.2
```

The following command displays the details of connections, which are advertised to peer device.

```
switch(config)#show ip nat synchronization advertised-translations
Source IP  Destination IP  Translated IP  TGT  Type  Interface/Profile
------------------------------------------------------------------------
-------------------
10.1.3.10:21800  191.1.1.10:80  139.1.1.1:21800  SRC  DYN  Port-Channel100
10.1.2.10:13750  191.1.1.10:80  139.1.1.1:13750  SRC  DYN  Port-Channel100
10.1.2.10:33757  191.1.1.10:80  139.1.1.1:5951   SRC  DYN  Port-Channel100
10.1.5.10:37111  191.1.1.10:80  139.1.1.1:7561   SRC  DYN  Port-Channel100
```

The following command displays the details of connections, which has been advertised by the peer device.

```
switch(config)#show ip nat synchronization discovered-translations
Source IP  Destination IP  Translated IP  TGT  Type  Interface/Profile
------------------------------------------------------------------------
-------------------
10.1.3.10:28606  191.1.1.10:80  139.1.1.1:28606  SRC  DYN  Port-Channel100
10.1.6.10:39697  191.1.1.10:80  139.1.1.1:39697  SRC  DYN  Port-Channel100
10.1.6.10:20583  191.1.1.10:80  139.1.1.1:31683  SRC  DYN  Port-Channel100
10.1.6.10:28419  191.1.1.10:80  139.1.1.1:28419  SRC  DYN  Port-Channel100
```

9.3.1.3 Applying NAT profile on a Tunnel Interface

The following commands apply the configured NAT profile on a tunnel interface.

Example

This command applies the NAT configuration profile ***natNameProfile*** to the tunnel ***Tunnel0***.

```
interface Tunnel0
   ip address 10.1.1.1/24
   tunnel source 2.1.1.1
   tunnel destination 2.1.1.2
```

```
ip nat service-profile <natNameProfile>
```

9.3.1.4 IP NAT Commands

IP NAT Commands

- clear ip nat flow translation
- ip address
- ip nat destination static
- ip nat pool
- ip nat source dynamic
- ip nat source static
- ip nat translation counters
- ip nat translation low-mark
- ip nat translation max-entries
- ip nat translation tcp-timeout
- ip nat translation udp-timeout
- show ip nat access-list interface
- show ip nat pool
- show ip nat synchronization advertised-translations
- show ip nat synchronization discovered-translations
- show ip nat synchronization peer
- show ip nat translation

9.3.1.4.1 clear ip nat flow translation

The **`clear ip nat flow translation`** command clears all or the specified NAT table entries.

Command Mode

Privileged EXEC

Command Syntax

`clear ip nat flow translation` [**HOST_ADDR** [**DEST_ADDR**]][**INTF**][**PROT_TYPE**]

Parameters

DEST_ADDR must immediately follow **HOST_ADDR**. All other parameters, including **HOST_ADDR** may be placed in any order.

- **HOST_ADDR** Host address to be modified. Options include:
  - ***no parameter*** All packets with specified destination address are cleared.
  - **address *local_ipv4*** IPv4 address.
  - **address *local_ipv4 local_port*** IPv4 address and port (port value ranges from **`1`** to **`65535`**).
- **DEST_ADDR** Destination address of translated packet. Destination address can be entered only when the **HOST_ADDR** is specified. Options include:
  - ***no parameter*** All packets with specified destination address are cleared.
  - ***global_ipv4*** IPv4 address.
  - ***global_ipv4 global_port*** IPv4 address and port (port value ranges from **`1`** to **`65535`**).
- **INTF** Route source. Options include:
  - ***no parameter*** All packets with specified destination address are cleared.
  - **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **interface loopback *l_num*** Loopback interface specified by ***l_num***.

- **interface management *m_num*** Management interface specified by ***m_num***.
- **interface port-channel *p_num*** Port-channel interface specified by ***p_num***.
- **interface vlan *v_num*** VLAN interface specified by ***v_num***.

- **PROT_TYPE** Filters packets based on protocol type. Options include:
  - ***no parameter*** All packets with specified destination address are cleared.
  - **tcp** TCP packets with specified destination address are cleared.
  - **udp** UDP packets with specified destination address are cleared.

Examples

- This command clears all dynamic entries from the NAT translation table.

```
switch# clear ip nat flow translation
switch#
```

- This command clears a specific NAT IP address ***172.22.30.52***.

```
switch# clear ip nat flow translation address 172.22.30.52
switch#
```

- This command clears the inside entry that maps the private address ***10.10.10.3*** to Internet address ***172.22.30.52***.

```
switch# clear ip nat flow translation address 172.22.30.52
 10.10.10.3
switch#
```

9.3.1.4.2 ip address

The **`ip address`** command configures the IPv4 address and connected subnet on the configuration mode interface. Each interface can have one primary address and multiple secondary addresses.

The **`no ip address`** and **`default ip address`** commands remove the IPv4 address assignment from the configuration mode interface. Entering the command without specifying an address removes the primary and all secondary addresses from the interface. The primary address cannot be deleted until all secondary addresses are removed from the interface.

Removing all IPv4 address assignments from an interface disables IPv4 processing on that port.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip address` [***ipv4_subnet***][**PRIORITY**]

`no ip address` [***ipv4_subnet***][**PRIORITY**]

`default ip address` [***ipv4_subnet***][**PRIORITY**]

Parameters

- ***ipv4_subnet*** IPv4 and subnet address (CIDR or address-mask notation). ***Running-config*** stores value in CIDR notation.
- **PRIORITY** interface priority. Options include:
  - ***no parameter*** The address is the primary IPv4 address for the interface.
  - **secondary** The address is the secondary IPv4 address for the interface.

Guidelines

The **`ip address`** command is supported on routable interfaces.

Example

This command configures an IPv4 address for ***VLAN 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ip address 10.0.0.1/24
switch(config-if-Vl200)#
```

9.3.1.4.3 ip nat destination static

The **`ip nat destination static`** command enables NAT of a specified destination address for the configuration mode interface. This command installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

When configuring twice NAT, an arbitrary NAT group number is used to associate the source NAT and destination NAT rules. This number must be the same in both rules.

The **`no ip nat destination static`** and **`default ip nat destination static`** commands disables NAT translation of the specified destination address by removing the corresponding **`ip nat destination static`** command from ***running_config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip nat destination static` **ORIGINAL** [**FILTER**] **TRANSLATED** [***PROT_TYPE***][group ***group_number***]

`no ip nat destination static` **ORIGINAL** [**FILTER**] **TRANSLATED** [***PROT_TYPE***] [group ***group_number***]

`default ip nat destination static` **ORIGINAL** [**FILTER**] **TRANSLATED** [***PROT_TYPE***][group ***group_number***]

Parameters

- **ORIGINAL** Destination address to be modified. Options include:
  - ***local_ipv4*** IPv4 address.
  - ***local_ipv4 local_port*** IPv4 address and port (port value ranges from **`1`** to **`65535`**)
- **FILTER** Access control list that filters packets. Options include:
  - ***no parameter*** All packets with specified destination address are cleared.

    - **access-list *list_name*** List that specifies the packets that are cleared. Not supported when configuring twice NAT.
- **TRANSLATED** Destination address of translated packet. Options include:
    - ***global_ipv4*** IPv4 address.
    - ***global_ipv4 global_port*** IPv4 address and port (port value ranges from **1** to **65535**). When configuring twice NAT, source and destination NAT rules must either both specify a port translation or both not specify a port translation.
- **PROT_TYPE** Filters packets based on protocol type. Options include:
    - ***no parameter*** All packets with specified destination address are cleared.
    - **protocol tcp** TCP packets with specified destination address are cleared.
    - **protocol udp** UDP packets with specified destination address are cleared.
- **group*group_number*** Used only when configuring twice NAT, the NAT group number associates a source NAT rule with a destination NAT rule on the same interface. The group number (values range from **1** to **255**) is arbitrary, but must be the same in both rules.

Examples

- These commands configure ***VLAN 201*** to translate destination address ***10.24.1.10*** to ***168.32.14.15***.

```
switch(config)# interface vlan 201
switch(config-if-Vl201)# ip nat destination static 10.24.1.10
 168.32.14.15
switch(config-if-Vl201)#
```

- These commands configure ***VLAN 201*** to translate the source address ***10.24.1.10*** to ***168.32.14.15*** for all packets with IP destination addresses in the ***168.10.1.1/32*** subnet.

```
switch(config)# ip access-list ACL2
switch(config-acl-ACL2)# permit ip 168.10.1.1/32 any
switch(config-acl-ACL2)# exit
switch(config)# interface vlan 201
switch(config-if-Vl201)#
switch(config-if-Vl201)#
```

- These commands configure ***interface Ethernet 2*** to translate the local source address ***10.24.1.10*** to the global source address ***168.32.14.15***, and to translate the local destination address ***10.68.104.3*** to the global destination address ***168.25.10.7*** for all packets moving through the interface. The use of NAT ***group 3*** is arbitrary, but must be the same in both rules.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# ip nat source static 10.24.1.10 168.32.14.15
 group 3
switch(config-if-Et2)# ip nat destination static 10.68.104.3
 168.25.10.7 group 3
```

9.3.1.4.4 ip nat pool

The **ip nat pool** command identifies a pool of addresses using start address, end address, and either netmask or prefix length. If its starting IP address and ending IP address are the same, there is only one address in the address pool.

The **no ip nat pool** removes the **ip nat pool**command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

`ip nat pool` ***pool_name*** [***ADDRESS_SPAN***] ***SUBNET_SIZE***

`no ip nat pool` ***pool_name***

`default ip nat pool` ***pool_name***

Parameters

- ***pool_name*** Name of the IP address pool.
- ***ADDRESS_SPAN*** Options include:
  - ***start_addr*** The first IP address in the address pool (IPv4 addresses in dotted decimal notation).
  - ***end_addr*** The last IP address in the address pool. (IPv4 addresses in dotted decimal notation).
- ***SUBNET_SIZE*** This functions as a sanity check to ensure it is not a network or broadcast network. Options include:
  - **netmask** ***ipv4_addr*** The netmask of the address pool's network (dotted decimal notation).
  - **prefix-length** **`0`** to **`32`** The number of bits of the netmask (of the address pool's network) that are ones (how many bits of the address indicate network).

Examples

- This command configures the pool of addresses using start address, end address, and prefix length of **`24`**.

```
switch(config)# ip nat pool poo1 10.15.15.15 10.15.15.25 prefix-
length 24
switch(config)
```

- This command removes the pool of addresses.

```
switch(config)# no ip nat pool poo1 10.15.15.15 10.15.15.25 prefix-
length 24
 switch(config)
```

9.3.1.4.5 ip nat source dynamic

The **`ip nat source dynamic`** command enables NAT of a specified source address for packets sent and received on the configuration mode interface. This command installs hardware translation entries for forward and reverse traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

The **`no ip nat source dynamic`** and **`default ip nat source dynamic`** commands disables NAT translation of the specified destination address by removing the corresponding **`ip nat source dynamic`** command from ***running_config*** .

Note: Ethernet and Port-channel interfaces should be configured as routed ports.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ip nat source dynamic access-list ***acl_name*** **POOL_TYPE**

no ip nat source dynamic access-list ***acl_name***

default ip nat source dynamic access-list ***acl_name***

Parameters

- ***acl_name*** Access control list that controls the internal network addresses eligible for NAT.
- **POOL_TYPE** Options include:
  - **overload** Translates multiple local addresses to a single global address. When overloading is enabled, conversations using the same IP address are distinguished by their TCP or UDP port number.
  - **pool** ***pool_name*** The name of the IP address pool. The pool is defined using the ip nat pool command.

    The pool option is required even if the pool has just one address. NAT uses that one address for all of the translations.
- ***pool_fullcone*** Enables full cone NAT where all requests from the same internal IP address and port are mapped to the same external IP address and port.

Examples

- This command configures the dynamic NAT source address and sets the NAT overload for pool ***P2***.

```
switch(config)# interface ethernet 3/1
switch(config-if-Et3/1)# ip nat source dynamic access-list ACL2 pool
 p2
switch(config-if-Et3/1)#
```

- This command disables the NAT source translation on interface Ethernet ***3/1***.

```
switch(config)# interface ethernet 3/1
switch(config-if-Et3/1)# no ip nat source dynamic access-list ACL2
switch(config-if-Et3/1)#
```

9.3.1.4.6 ip nat source static

The **ip nat source static** command enables NAT of a specified source address for the configuration mode interface. This command installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

When configuring twice NAT, an arbitrary NAT group number is used to associate the source NAT and destination NAT rules. This number must be the same in both rules.

The **no ip nat source static** and **default ip nat source static** commands disables NAT translation of the specified source address by removing the corresponding **ip nat source** command from ***running_config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ip nat source static **ORIGINAL** [**FILTER**] **TRANSLATED** [**PROT_TYPE**] [group *group_number*]

`no ip nat source static` **ORIGINAL** [**FILTER**] **TRANSLATED** [**PROT_TYPE**] [group *group_number*]

`default ip nat source static` **ORIGINAL** [**FILTER**] **TRANSLATED** [**PROT_TYPE**] [group *group_number*]

Parameters

- **ORIGINAL** Source address to be modified. Options include:
  - ***original_ipv4*** IPv4 address.
  - ***original_ipv4 original_port*** IPv4 address and port (port value ranges from `1` to `65535`).
- **FILTER** Access control list that filters packets. Options include:
  - ***no parameter*** All packets with specified source address are cleared.
  - **access-list *list_name*** List that specifies the packets that are cleared. Not supported when configuring twice NAT.
- **TRANSLATED** Source address of translated packet. Options include:
  - ***translated_ipv4*** IPv4 address.
  - ***translated_ipv4 translated_port*** IPv4 address and port (port value ranges from `1` to `65535`). When configuring twice NAT, source and destination NAT rules must either both specify a port translation or both not specify a port translation.
- **PROT_TYPE** Filters packets based on protocol type. Options include:
  - ***no parameter*** All packets with specified source address are cleared.
  - **protocol tcp** TCP packets with specified source address are cleared.
  - ***protocol udp*** UDP packets with specified source address are cleared.
- **group *group_number*** Used only when configuring twice NAT, the NAT group number associates a source NAT rule with a destination NAT rule on the same interface. The group number (values range from `1` to `255`) is arbitrary, but must be the same in both rules.

Restrictions

- If **ORIGINAL** includes a port, **TRANSLATED** must also include a port.
- If **ORIGINAL** does not include a port, **TRANSLATED** cannot include a port.

Examples

- These commands configure ***VLAN 101*** to translate source address ***10.24.1.10*** to ***168.32.14.15***.

```
switch(config)# interface vlan 101
switch(config-if-Vl101)# ip nat source static 10.24.1.10
 168.32.14.15
switch(config-if-Vl101)#
```

- These commands configure ***VLAN 101*** to translate the source address ***10.24.1.10*** to access-list ACL1 ***168.32.14.15*** for all packets with IP destination addresses in the ***168.10.1.1/32*** subnet.

```
switch(config)# ip access-list ACL1
switch(config-acl-ACL1)# permit ip any 168.10.1.1/24
switch(config-acl-ACL1)# exit
switch(config)# interface vlan 101
switch(config-if-Vl101)# ip nat source static 10.24.1.10 access-list
 ACL1
168.32.14.15
switch(config-if-Vl101)#
```

- These commands configure Ethernet interface ***2*** to translate the local source address ***10.24.1.10*** to the global source address ***168.32.14.15***, and to translate the local destination

address ***10.68.104.3*** to the global destination address ***168.25.10.7*** for all packets moving through the interface. The use of NAT ***group 3*** is arbitrary, but must be the same in both rules.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# ip nat source static 10.24.1.10 168.32.14.15
 group 3
switch(config-if-Et2)# ip nat destination static 10.68.104.3
 168.25.10.7 group 3
```

9.3.1.4.7 ip nat translation counters

The **ip nat translation counters** command enables the feature to count packets that are translated by static and twice NAT rules in hardware. Once this feature is enabled, all current rules in hardware and new rules that are configured after running this command receive policers for counting packets.

The **no ip nat translation counters** and **default ip nat translation counters** commands disable the packet counter feature for static and twice NAT connections.

Command Mode

Global Configuration

Command Syntax

ip nat translation counters

no ip nat translation counters

default ip nat translation counters

Guidelines

The **ip nat translation counters** command is supported on the DCS-7150 series switches only. This command is solely intended to debug static and twice NAT translation failures in hardware. Disable this feature after completing troubleshooting. If this feature remains enabled even when the count of static connections exceed ***275***, it can cause unpredictable behavior including restart of FocalPointV2 agent. The restart of FocalPointV2 agent results in traffic disruption.

Example

The **ip nat translation counters** command enables the packet counter feature for static and twice NAT connections. Using the **show ip nat translation hardware detail** and the **show ip nat translation twice hardware detail** commands, you can verify the packet count.

```
switch(config)# ip nat translation counters
switch(config)# show ip nat translation hardware detail

Source IP        Destination IP  Translated IP  TGT Type Intf   Proto  Packets  Packets Reply
-------------------------------------------------------------------------------------------
192.168.10.2:0    -              20.1.10.2:0    SRC STAT Vl2640 -      2         1
192.168.110.2:0   -              20.1.110.2:0   SRC STAT Vl2640 -      2         1

switch(config)# show ip nat translation twice hardware detail
Source IP     Destination IP  Translated     Translated    Intf   Group   Packets  Packets
                              Src IP         Dst IP                Proto           Reply
-------------------------------------------------------------------------------------------
192.16.50.2:0  10.1.50.2:0     20.1.50.2:0    10.1.60.2:0   Vl2922 2  -    2        1
19.16.150.2:0  10.1.150.2:0    20.1.150.2:0   10.1.160.2:0  Vl2922 12 -    2
```

9.3.1.4.8 ip nat translation low-mark

The **ip nat translation low-mark** command configures the minimum threshold that triggers the resumption of programming new NAT translation connections.

The ip nat translation max-entries command specifies the maximum number of NAT translation connections that can be stored. When this limit is reached, new connections are dropped instead of being programmed in hardware or software. At this point no new connections will be programmed until the number of stored entries drop below the configured low-mark, expressed as a percentage of the max-entries value. The default low mark value is 90%.

The **`no ip nat translation low-mark`** and **`default ip nat translation low-mark`** commands restores the default low-mark value by removing the **`ip nat translation low-mark`** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

`ip nat translation low-mark` ***threshold***

`no ip nat translation low-mark`

`default ip nat translation low-mark`

Parameters

threshold Percentage of maximum connection entries. Value ranges from **`1`** to **`99`**. Default is **`90`**.

Example

This command globally sets the translation low mark of ***93%***.

```
switch(config)# ip nat translation low-mark 93
switch(config)#
```

9.3.1.4.9 ip nat translation max-entries

The **`ip nat translation max-entries`** command specifies maximum number of NAT translation connections. After this threshold is reached, new connections are dropped until the number of programmed connections is reduced below the level specified by the ip nat translation low-mark command.

The **`no ip nat translation max-entries`** and **`default ip nat translation max-entries`** commands removes the maximum connection limit and resets the parameter value to zero by removing the **`ip nat translation max-entries`** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

`ip nat translation max-entries` ***connections***

`no ip nat translation max-entries`

`default ip nat translation max-entries`

Parameters

- ***connections*** The maximum number of NAT translation connections. Value ranges from **`0`** to **`4294967295`**. Default value is **`0`**, which removes the connection limit.

Example

This command limits the number of NAT translation connections the switch can store to ***3000***.

```
switch(config)# ip nat translation max-entries 3000
switch(config)#
```

9.3.1.4.10 ip nat translation tcp-timeout

The **ip nat translation tcp-timeout** command specifies the translation timeout period for translation table entries. The timeout period specifies the interval during which the switch will attempt to reuse an existing TCP translation for devices specified by table entries.

The **no ip nat translation tcp-timeout** and **default ip nat translation tcp-timeout** commands reset the timeout to its default by removing the corresponding **ip nat translation tcp-timeout** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

ip nat translation tcp-timeout ***period***

no ip nat translation tcp-timeout

default ip nat translation tcp-timeout

Parameters

period Time-out period in seconds for port translations. Value ranges from **0** to **4294967295**. Default value is **86400** (24 hours).

Examples

- This command sets the TCP timeout for translations to ***600*** seconds.

```
switch(config)# ip nat translation tcp-timeout 600
switch(config)#
```

- This command removes the TCP translation timeout.

```
switch(config)# no ip nat translation tcp-timeout
switch(config)#
```

9.3.1.4.11 ip nat translation udp-timeout

The **ip nat translation udp-timeout** command specifies the translation timeout period for translation table entries. The timeout period specifies the interval the switch attempts to establish a UDP connection with devices specified by table entries.

The **no ip nat translation udp-timeout** and **default ip nat translation udp-timeout** commands disables NAT translation of the specified destination address by removing the corresponding **ip nat translation udp-timeout** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

ip nat translation udp-timeout ***period***

no ip nat translation udp-timeout

default ip nat translation udp-timeout

Parameters

period Value ranges from **0** to **4294967295**. Default value is **300** (5 minutes).

Examples

- This command globally sets the timeout for UDP to ***800*** seconds.

```
switch(config)# ip nat translation udp-timeout 800
```

- This command removes the timeout for UDP.

```
switch(config)# no ip nat translation udp-timeout
```

9.3.1.4.12 show ip nat access-list interface

The **show ip nat acl interface** command displays the access control lists (ACLs) that are configured as source NAT or destination NAT filters. The display indicates ACL rules that do not comply with these NAT requirements:

- Source IP address is any.
- Destination IP address may use any mask size.
- Source port matching is not allowed.
- Protocol matching is not allowed.

Command Mode

EXEC

Command Syntax

show ip nat access-list [**INTF**][**LISTS**]

Parameters

- **INTF** Filters NAT statements by interface. Options include:
  - ***no parameter*** Includes all statements on all interfaces.
  - **interface ethernet *e_num*** Statements on specified Ethernet interface.
  - **interface loopback *l_num*** Statements on specified Loopback interface.
  - **interface management *m_num*** Statements on specified Management interface.
  - **interface port-channel *p_num*** Statements on specified Port-Channel Interface.
  - **interface vlan *v_num*** Statements on specified VLAN interface.
  - **interface vxlan *vx_num*** Statements on specified VXLAN interface.
- **LISTS** ACLs displayed by command. Options include:
  - ***no parameter*** All ACLs.
  - ***acl_name*** Specifies individual ACL.

Example:

These commands display the NAT command usage of the ACL1 and ACL2 access control lists.

```
switch> show ip nat acl ACL1

acl ACL1
```

```
        (0.0.0.0/0, 168.10.1.1/32)
Interfaces using this ACL for Nat:
        Vlan100

switch> show ip nat acl ACL2
acl ACL2
        (168.10.1.1/32, 0.0.0.0/0)
Interfaces using this ACL for Nat:
        Vlan201 switch>
```

9.3.1.4.13 show ip nat pool

The **show ip nat pool** command displays the configuration of the address pool.

Command Mode

EXEC

Command Syntax

show ip nat pool POOL_SET

Parameters

- ***pool_name*** The name of the pool.
- **POOL_SET** Options include:
  - ***no parameter*** All configured port channels.
  - ***pool_name*** The name of the pool.

Examples

- This command displays all the address pools configured on the switch.

```
switch# show ip nat pool
Pool                 StartIp               EndIp
 Prefix
p1                   10.15.15.15           10.15.15.25           24
p2                   10.10.15.15           10.10.15.25           22
p3                   10.12.15.15           10.12.15.25           12
 switch#
```

- These commands display specific information for the address pools configured on the switch.

```
switch# show ip nat pool p1
Pool                 StartIp               EndIp
 Prefix
p1                   4.1.1.1               4.1.1.2               24
                     1.1.1.1               1.1.1.2               24
                     3.1.1.1               3.1.1.2               24
switch# show ip nat pool p2
Pool                 StartIp               EndIp
 Prefix
p2                   10.1.1.1              10.1.1.2              16
 switch#
```

9.3.1.4.14 show ip nat synchronization advertised-translations

The **show ip nat synchronization advertised-translations** command displays the detailed status of devices that are advertised to a peer device.

Command Mode

EXEC

Command Syntax

show ip nat synchronization advertised-translations

Example

This command displays details of devices that are advertised to a peer device.

```
switch# show ip nat synchronization advertised-translations

Source IP         Destination IP   Translated IP          TGT  Type
 Intf
-------------------------------------------------------------
------------
61.0.0.15:6661    100.0.0.2:80     192.170.230.171:6661    SRC  DYN
 Et5
61.0.0.41:2245    100.0.0.2:80     192.170.230.170:2245    SRC  DYN
 Et5
61.0.0.48:22626   100.0.0.2:80     192.170.230.169:22626   SRC  DYN
 Et5
61.0.0.41:22601   100.0.0.2:80     192.170.230.170:22601   SRC  DYN
 Et5
61.0.0.41:16798   100.0.0.2:80     192.170.230.170:16798   SRC  DYN
 Et5
61.0.0.18:22605   100.0.0.2:80     192.170.230.177:22605   SRC  DYN
 Et5
61.0.0.16:2256    100.0.0.2:80     192.170.230.166:2256    SRC  DYN
 Et5
```

9.3.1.4.15 show ip nat synchronization discovered-translations

The **show ip nat synchronization discovered-translations** command displays details of what has been advertised from a peer device.

Command Mode

EXEC

Command Syntax

show ip nat synchronization discovered-translations

Example

This command displays details of devices that are advertised to a peer device.

```
switch# show ip nat synchronization discovered-translations

Source IP         Destination IP    Translated IP          TGT  Type
 Intf
-------------------------------------------------------------
-------------
```

```
61.0.2.229:63     100.0.0.2:63     170.24.86.180:63        SRC  DYN
 Et5
61.0.15.51:63     100.0.0.2:63     170.24.73.90:63         SRC  DYN
 Et5
61.0.6.68:63      100.0.0.2:63     170.24.110.128:63       SRC  DYN
 Et5
61.0.7.163:63     100.0.0.2:63     170.24.104.35:63        SRC  DYN
 Et5
```

9.3.1.4.16 show ip nat synchronization peer

The **`show ip nat synchronization peer`** command displays the detailed status of a peer device.

Command Mode

EXEC

Command Syntax

`show ip nat synchronization peer`

Example:

This command displays details of a peer device with an IP address of ***11.11.11.0*** and interface ***VLAN 1111*** that is used to connect to the peer device.

```
switch# show ip nat synchronization peer
Description : Value
Peer : 11.11.11.0
Connection Port : 4532
Connection Source : 0.0.0.0
Kernel Interface : vlan1111
Local Interface : Vlan1111
Established Time : 1969-12-31 16:00:00
Connection Attempts : 0
Oldest Supported Version : 1
Newest Supported Version : 1
Version Compatible : True
Connection State : connected
Shutdown State : False
Status Mount State : mountMounted
Version Mount State : mountMounted
Recover Mount State : mountMounted
Reboot Mount State : mountMounted
```

9.3.1.4.17 show ip nat translation

The **`show ip nat translation`** command displays configured NAT statements in the switch hardware.

Command Mode

EXEC

Command Syntax

`show ip nat translation` [address | address-only | destination | detail | dynamic | hardware | interface | kernel | max-entries | source | static | summary | twice]

Command position of all parameters are interchangeable.

Parameters

- ***no parameter*** Displays all NAT connections installed in software.
- **address *ipv4_addr*** Displays NAT connections of the specified IPv4 host address.
- **address-only *ipv4_addr*** Displays address-only NAT connections of the specified IPv4 host address.
- **destination** Displays destination NAT connections installed in software.
- **detail** Displays detailed output of all NAT connections.
- **dynamic** Displays dynamic NAT connections.
- **hardware** Displays NAT connections installed in hardware.
- **interface** Filters NAT connections by interface. Options include:
  - **interface ethernet *e_num*** Displays NAT connections of the specified ethernet interface.
  - **interface port-channel *p_num*** Displays NAT connections of the specified port-channel interface.
  - **interface vlan *v_num*** Displays NAT connections of the specified VLAN interface.
- **kernel** Displays NAT connections installed in kernel.
- **max-entries** Displays the configured NAT connection limits of a hardware.
- **source** Displays source NAT connections installed in software.
- **static** Displays static NAT connections.
- **summary** Displays summary of all NAT connections.
- **twice** Displays twice NAT connections.

Examples

- This command displays all configured NAT translations.

```
switch> show ip nat translation

Source IP          Destination IP   Translated IP         TGT Type Intf
-------------------------------------------------------------------------
192.168.1.10:62822 172.22.22.40:53  172.17.254.161:62822  SRC DYN  Vl3925
192.152.1.10:20342 172.22.22.40:80  172.17.254.161:22222  SRC STAT Vl3945 switch#
```

- This command displays NAT connections of the specified ethernet interface.

```
switch> show ip nat translation dynamic interface Ethernet 26

Source IP          Destination IP    Translated IP       TGT Type Intf
-----------------------------------------------------------------------
192.168.1.2:8080   10.1.1.5:600      20.1.1.5:8080       SRC DYN  Et26
```

- This command displays the configured NAT connection limits of a hardware.

```
switch> show ip nat translation max-entries

Global connection limit                          100
Global connection limit low mark                 90(90%)
Hosts connection limit                           20
Hosts connection limit low mark                  18(90%)
Total number of connections                       1

Host           Max-Entries           Low-Mark              Connections
-------------------------------------------------------------------
10.1.1.1       10                    9(90%)                0
```

9.3.2 Media Access Control Security

This section explains the basic concepts about Media Access Control Security (MACsec) including overview, configuration, and the different MACsec commands that are used.

- MACsec Overview
- Configuring MACsec
- Displaying MACsec Information

- MACsec Key Retirement Immediate
- MACsec EAP-FAST Support
- MACsec Proxy For VXLAN
- MACsec Fallback to Unprotected Traffic
- MACsec Commands

9.3.2.1 MACsec Overview

Media Access Control Security (MACsec) is an industry standard encryption mechanism that protects all traffic flowing on the Ethernet links. MACsec is based on IEEE 802.1X and IEEE 802.1AE standards.

The major benefits of MACsec are:

- MACsec supports packet authentication by providing integrity checking so that packet data is not altered during a packet flow.
- MACsec provides secure encryption at Layer 2 level by ensuring complete data confidentiality.
- A high density MACsec solution for Cloud Data Centers is integrated with 7500R for highest density and performance in a modular platform.
- Cost and performance is optimized for Data Center Interconnect to transport massive volumes of traffic through metro or long haul networks.
- Secure transport of data over distance is made possible with MACsec encryption eliminating additional intermediate devices.

9.3.2.1.1 MACsec Terminology

MACsec Key Agreement Protocol (MKA) is the key agreement protocol for discovering MACsec peers and negotiating keys between MACsec peers (IEEE 802.1X-REV).

A **Connectivity Association (CA)** is a security relationship between MACsec-capable devices (endpoints). Endpoints in the same CA share a Connectivity Association Key (CAK). Arista implementation allows 2 endpoints.

A **Connectivity Association Key (CAK)** is a master key that is used to generate all other keys that are used for MACsec. Endpoints in the same secure Connectivity Association (CA) share a CAK. This key can either be a static pre-shared key, or dynamically derived with the use of 802.1X authentication.

Primary Key- It is ideally the CAK for the MKA session in progress. The Primary key is a combination of key name and the actual key. For example, when a configuration uses ***0abcd1 0 1234abcd*** as a primary key, in this the ***0abcd1*** is the hex key name, while ***1234abcd*** is the actual key. Note, a key name must be in hex format too.



Note: That the operator ***0*** means the key being entered here is unencrypted (or unhashed), vs. ***7*** meaning the hashed version of this key is being entered (in cases where the configuration is being replayed onto the switch).

Fallback Key- In case the primary configured key does not establish its connection, the fallback key is used, so as to ensure no loss of traffic.

Secure Association Key (SAK) - The SAK is derived from the CAK and is the key used by the network device ports to encrypt traffic for a given session.

Key Server - One of the MACsec peers in the CA becomes the Key Server. The main role of the Key Server is to create and distribute Secure Association Keys (SAKs), which are used in actual data encryption.

A **Static Secure Association Key (SAK)** is an SAK configured directly on a switch and used with unidirectional links, in situations where the MKA protocol is not feasible. Static SAKs require the use of eXtended Packet Numbering (XPN) cipher suites.

9.3.2.1.2 MACsec Limitations

The limitations of MACsec are:

- MACsec is supported only on point-to-point links, unless static SAK is enabled.
- When MACsec is enabled on an interface for the first time, interface flapping occurs for MACsec to take effect.
- If static SAK is not enabled, the port does not forward any traffic until the MKA protocol converges and negotiates encryption keys. This occurs initially when MACsec is configured on a port.

9.3.2.1.3 MACsec Licensing

MACsec encryption is a EOS licensed feature. A valid MACsec license must be configured on a switch. MACsec licenses are tied to a switch serial number and the licensee. Every switch running MACsec requires a separate license of its own.

There are two ways of configuring MACsec license:

1. Use the command **`license`** ***licensee_name license_value*** in MACsec mode. The license value is an 8 digit hexadecimal number. This method of license configuration is no longer being used except for backward compatibility.
2. Use the command **`license`** **import** ***license_file_path*** in Global configuration mode. All new licenses generated on the license portal are JSON-based.

Contact your system engineer to acquire the required license codes before attempting to configure MACsec.

9.3.2.1.4 MACsec in FIPS mode

Federal Information Processing Standards (FIPS) are a set of standards defined by the United States federal government related to the processing of data in computer systems by non-military government agencies and government contractors. These standards define specific requirements for various purposes such as ensuring computer security and interoperability within and across the computer networking industry.

Arista devices are compliant with FIPS 140. The FIPS 140 enforces the use of a "FIPS Crypto Module". This both ensures that the algorithms are correct and restricts the set of allowed algorithms to those approved by the FIPS standard. These are the FIPS supported algorithms AES-128/256, SHA-256/512, RSA with 2048 bit keys, a subset of EcDSA. MACsec has both the AES-128-GCM and AES-256-GCM algorithms certified for the data plane. The FIPS mode is enabled using the **`fips restrictions`** command which when enabled filters out any unapproved algorithms and warns you if you try to set them.

9.3.2.1.5 VLAN Tagged MACsec

Media Access Control Security (MACsec) is configured on subinterfaces using the **`mac security profile`** command. Since subinterfaces are logical interfaces that send and receive VLAN tagged traffic, encryption/ decryption is applied per VLAN tag.

9.3.2.1.6 MACsec Using Static Secure Association Key

MAC security uses the MACsec Key Agreement (MKA) protocol to negotiate between peers using keys (CAKs and CKNs) which are either pre-shared or derived from an 802.1X session, and derives a Secure Association Key (SAK) based on the MKA negotiation. This SAK is then programmed in hardware and used for encrypting and decrypting data traffic. In cases where MKA negotiation is not feasible but encryption and decryption of traffic is required (such as unidirectional links), MACsec can instead be configured to use static Secure Association Keys (SAK) configured separately on transmitting and receiving peers. Each peer can have up to four receiving secure keys and one transmitting key.

9.3.2.2 Configuring MACsec

These sections describe basic MACsec configuration steps:

- Enabling MACsec
- Configuring MACsec for MKA
- Configuring the FIPS mode
- Configuring MACsec Profile on a Subinterface
- Configuring MACsec Using Static SAK
- Configuring MACsec Proxy For VXLAN
- Configuring MAC Security Dynamic Key Derivation
- Configuring MACsec Fallback to Unprotected Traffic

9.3.2.2.1 Enabling MACsec

MACsec requires a configuration profile. Use the **mac security** command to enable MACsec and enter MAC Security Configuration Mode, then use the **profile** command to create a profile and enter MAC Security Profile Configuration Mode, where the following commands are available for detailed configuration:

- cipher
- key (MACsec)
- mka key-server
- mka session
- replay
- sci
- traffic unprotected allow

Example

These commands enable MACsec and enter MAC Security Configuration Mode, then create a profile named "MACsec_test" and enter MAC Security Profile Configuration Mode.

```
switch(config)# mac security
switch(config-mac-security)# profile MACsec_test
switch(config-mac-security-profile-MACsec_test)#
```

9.3.2.2.2 Configuring MACsec for MKA

By default, MAC security (MACsec) uses the MACsec Key Agreement (MKA) protocol to negotiate and exchange encryption keys among peers. To complete a typical MACsec configuration, use the **cipher** command to select a valid encryption standard. Then use the **key** command to enter a Connectivity Association Key (CAK). You can use the **fallback** option to add a fallback CAK to be used if the primary CAK fails.

The key server is responsible for generating and distributing encryption keys. Run the **mka key-server priority** command on a peer to change its priority. The peer with the lowest priority is elected as the key server. If multiple peers have the same priority, the one with the lowest MAC address is chosen. Priority values range from 0 to 255 and the default priority is 16.

Configure the period at which the Secure Association Key (SAK) is refreshed with the **mka session rekey-period** command. MACsec uses an SAK for encrypting data traffic, and this SAK is derived from the CAK. Rekey-period values range from ***30*** to ***100000*** seconds. By default, there is no session rekey period, and the SAK will not be refreshed periodically.

To improve the randomness of the numbers used to generate MACsec's cryptographic keys, add a source of entropy with the **entropy source** command in Management Security Configuration Mode.

Examples

These commands configure MACsec to use the AES256-GCM-XPN cipher and add a key and fallback key. For MKA with pre shared key configuration, keys with any length are allowed to work. However, we should have keys with 64 hexadecimal digits in length for a 256-bit cipher.

```
switch(config-mac-security-profile-test)# cipher aes256-gcm-xpn
switch(config-mac-security-profile-test)# key 0abc12340def56780abc12340def5678
 7 06070E234E4D0A48544540585F507E
switch(config-mac-security-profile-test)# key 0def56780abc12340def56780abc1234
 7 09484A0C1C0311475E5A527D7C7C70 fallback
```

These commands give the switch a key-server priority of 10 and an MKA session rekey period of 600 seconds.

```
switch(config-mac-security-profile-test)# mka key-server priority 10
switch(config-mac-security-profile-test)# mka session rekey-period 600
```

These commands add an entropy source for more random cryptographic keys.

```
switch(config-mac-security-profile-test)# management security
switch(config-mgmt-security)# entropy source hardware
```

These commands apply the "test" profile to Ethernet interface 5/3/1.

```
switch(config-mgmt-security)# interface ethernet 5/3/1
switch(config-if-Et5/3/1)# mac security profile test
switch(config-if-Et5/3/1)#
```

9.3.2.2.3 Configuring the FIPS mode

To configure the FIPS mode on the MACsec protocol, use the **FIPS** command.

Example

This command configures the FIPS mode on the MACsec protocol.

```
switch(config)# mac security
switch(config-mac-security) fips restrictions
```

9.3.2.2.4 Configuring MACsec Profile on a Subinterface

Following are the commands used to configure a MACsec profile on a subinterface.

Example

- The following example enables MAC security on a subinterface with a predefined MACsec profile **test-profile**.

```
switch(config)# interface ethernet1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# interface ethernet1.10
switch(config-if-Et1.10)# encapsulation dot1q vlan 20
switch(config-if-Et1.10)# mac security profile test-profile
```

9.3.2.2.5 Configuring MACsec Using Static SAK

Static SAK is configured for receive (Rx) and transmit (Tx) directions separately. In the Rx direction, multiple SAKs can be configured. For the Tx direction, only one SAK is allowed at a time. An SAK configured for Rx on the local peer should match the SAK configured for Tx on the connected peer, and vice versa. The Rx

direction should be configured first on all the MACsec peers, and then the Tx direction should be configured. Use the **cipher** command to select a cipher suite. You must choose an eXtended Packet Number (XPN) cipher suite, such as AES128-GCM-XPN or AES256-GCM-XPN. Static SAK will not work with a non-XPN cipher.

Examples

- These commands select the AES256-GCM-XPN cipher suite for the MACsec profile ***rx_test*** on the receiving peer (Rx).

```
switch(config)# mac security
switch(config-mac-security)# profile rx_test
switch(config-mac-security-profile-rx_test)# cipher aes128gcm-xpn
switch(config-mac-security-profile-rx_test)#
```

- This command configures the key source as static SAK.

```
switch(config-mac-security-profile-rx_test)# key source sak static
switch(config-mac-security-profile-rx_test-sak-static)#
```

- These commands configure a secure channel identifier (SCI) on the receiving peer. The SCI is a MAC address with six hexadecimal octets and a decimal port number.

```
switch(config-mac-security-profile-rx_test-sak-static)# secure channel rx
switch(config-mac-security-profile-rx_test-sak-static-rx)# identifier
 01:02:03:04:05:06::1234
switch(config-mac-security-profile-rx_test-sak-static-rx)#
```

- This command configures an SAK and assigns it an association number (AN) of ***0***.

```
switch(config-mac-security-profile-rx_test-sak-static-rx)# an 0 key 0
 11112222333344445555666677778888
switch(config-mac-security-profile-rx_test-sak-static-rx)#
```

- This command configures another SAK and its association number. Up to four associations can be configured.

```
switch(config-mac-security-profile-rx_test-sak-static-rx)# an 1 key 0
 9999aaaabbbbccccddddeeeeffff0000
switch(config-mac-security-profile-rx_test-sak-static-rx)#
```

- These commands configure the secure channel on a transmitting peer using the profile ***tx_test***. Only one SAK can be configured per transmitting peer. This will encrypt traffic in the Tx direction, so the receiving peer must be configured with a matching SAK to decrypt this traffic.

```
switch(config-mac-security-profile-tx_test-sak-static)# secure channel tx
switch(config-mac-security-profile-tx_test-sak-static-tx)# identifier
 01:02:03:04:05:07::1235
switch(config-mac-security-profile-tx_test-sak-static-tx)# an 0 key 0
 22223333444455556666777788889999
switch(config-mac-security-profile-tx_test-sak-static-tx)#
```

9.3.2.2.6 Configuring MACsec Proxy For VXLAN

The switch platforms which use this feature are:

- 7280SRAM-48C6
- 7280CR2M-30
- 7500R2M-36CQ-LC

The mandatory steps to configure a MACsec proxy sub-interface on an Arista switch are:

1. Configure the parent interface to be a routed port.
2. Create a L3 sub-interface on the parent interface. This is the MACsec proxy sub-interface.
3. Create a L2 sub-interface on the parent interface. This is the MACsec patch sub-interface.
4. Configure and enable the MACsec proxy port on a sub-interface.
5. Configure the VXLAN tunnel.
6. Assign the forwarding VLAN ID for the MACsec patch sub-interface and VXLAN tunnel.

Example Configurations

a. Configure a ***100g*** MACsec interface as a routed port.

```
switch(config)# interface et49/1
switch(config-if-Et49/1)# no switchport
```

b. Create a new L3 sub-interface - ***et49/1.1*****.**

```
switch(config-if-Et49/1)# interface et49/1.1
```

c. Create a new L2 sub-interface - ***et49/1.2***.

```
switch(config-if-Et49/1)# interface et49/1.2
```

d. Configure the MACsec proxy port, and enable MACsec on the proxy port.

```
switch(config)# interface et49/1.1
switch(config-if-Et49/1.1)# mac security proxy patch Ethernet49/1.2
switch(config-if-Et49/1.1)# mac security profile test1
switch(config-if-Et49/1.1)# ip address 2.2.2.1/24
```

e. Configure the VXLAN tunnel. The remote VTEP is provided as the flood VTEP.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan source-interface Loopback0
switch(config-if-Vx1)# vxlan udp-port 4789
switch(config-if-Vx1)# vxlan vlan 20 vni 20
switch(config-if-Vx1)# vxlan vlan 20 flood vtep 100.100.100.2
```

f. Configure the L2 MACsec patch interface to be in the same VLAN as VXLAN.

```
switch(config)# interface et49/1.2
switch(config-if-Et49/1.2)# vlan id 20
```

9.3.2.2.7 Configuring MAC Security Dynamic Key Derivation

9.3.2.2.8 Configuring MACsec Fallback to Unprotected Traffic

This feature is supported on all MACsec capable cards except for 7500E-6CFPX-LC.

The MACsec Fallback to Unprotected Traffic feature is configured under MACsec profile mode using the **`traffic unprotected allow`** command. The **no** form of the command removes the configuration from the switch. This configuration must be present in both the peers for the unprotected traffic to flow between them successfully.

Example

```
switch(config-mac-security-profile-sampleProfile)# no traffic unprotected allow
```

9.3.2.3 Displaying MACsec Information

The following sections provide information about MACsec on a switch.

This section contains the following topics:

- Displaying MACsec Information
- Displaying MACsec detailed information
- Displaying MACsec participants
- Displaying MACsec participants detailed information
- Displaying MACsec MKA Counters
- Displaying MACsec Security Counters Detailed Information
- Displaying MACsec Security Counters
- Displaying MACsec MKA Counters detailed information
- Displaying MACsec FIPS Status
- Displaying Information for MACsec Using Static Secure Association Key

9.3.2.3.1 Displaying MACsec Information

The **`show mac security interface`** command shows information about the MACsec on the interface.

Example

```
switch# show mac security interface
Interface SCI Controlled Port Key in Use
Ethernet4/1/1 28:99:3a:82:6f:82::605 True 9d5bc0d3076ea4a08b99b9d9:1
Ethernet4/3/1 28:99:3a:82:6f:85::613 True 9d5bc0d3076ea4a08b99b9d9:1
```

9.3.2.3.2 Displaying MACsec detailed information

Use the **`show mac security interface detail`** command to display detailed information about MACsec.

Example

```
switch# show mac security interface detail
Interface: Ethernet4/1/1
    SCI: 28:99:3a:82:6f:82::605
    SSCI: 00000002
    Controlled port: True
    Key server priority: 16
    Session rekey period: 0
    Traffic: Protected
    Key in use: 9d5bc0d3076ea4a08b99b9d9:1
    Latest key: None
    Old key: 9d5bc0d3076ea4a08b99b9d9:1(RT)

Interface: Ethernet4/3/1
    SCI: 28:99:3a:82:6f:85::613
    SSCI: 00000001
    Controlled port: True
    Key server priority: 16
    Session rekey period: 0
    Traffic: Protected
    Key in use: 9d5bc0d3076ea4a08b99b9d9:1
    Latest key: None
    Old key: 9d5bc0d3076ea4a08b99b9d9:1(RT)
```

About the Output:

- **`Interface`**: Name of the interface.

- **`Secure Channel Identifier (SCI)`**: Combination of MAC address and port number. Used to uniquely identify a Mac Security port.
- **`Controlled Port`**: Indicates if Mac Security is enabled on the port. A value of True indicates that encryption is enabled on the port.
- **`Key In Use`**: The SAK identifier currently in use. Combination of Key Servers message identifier (see below) and key number.
- **`Key Server Priority`**: Configured key server priority.
- **`Session Rekey Period`**: Configured session rekey period.
- **`Latest Key`**: Latest SAK being negotiated by Mac Security Key Agreement Protocol (MKA)
- **`Old Key`**: The last SAK negotiated by Mac Security Key Agreement Protocol (MKA)



Note: Latest and Old key are MKA protocol specific terminology and are used to refer to the last two keys in use. For all practical purposes, Key In Use field is used to identify the current key.

9.3.2.3.3 Displaying MACsec participants

Use the **`show mac security participants`** command to display information about the MACsec participants.

Example

```
switch# show mac security participants
Interface: Ethernet4/1/1
    CKN: abcd
      Message ID: 9d5bc0d3076ea4a08b99b9d9
      Elected self: True
      Success: True
      Principal: True
      Default: False

    CKN: dead
      Message ID: 4ef4cf009161bd551b5e7434
      Elected self: True
      Success: True
      Principal: False
      Default: True

Interface: Ethernet4/3/1
    CKN: abcd
      Message ID: c79ad8882c2dd3a8e838a691
      Elected self: False
      Success: True
      Principal: True
      Default: False

    CKN: dead
      Message ID: 3dfd4486b5f68a81014a37ec
      Elected self: False
      Success: True
      Principal: False
      Default: True
```

9.3.2.3.4 Displaying MACsec participants detailed information

Use the **`show mac security participants detail`** command to display detailed information about the MACsec participants.

Example

```
switch# show mac security participants detail
```

```
Interface: Ethernet4/1/1
    CKN: abcd
      Message ID: 9d5bc0d3076ea4a08b99b9d9
      Elected self: True
      Success: True
      Principal: True
      Default: False
      KeyServer SCI: 28:99:3a:82:6f:82::605
      SAK transmit: True
      LLPN exhaustion: 0
      Distributed key identifier: 9d5bc0d3076ea4a08b99b9d9:1
      Live peer list: ['c79ad8882c2dd3a8e838a691']
      Potential peer list: []

    CKN: dead
      Message ID: 4ef4cf009161bd551b5e7434
      Elected self: True
      Success: True
      Principal: False
      Default: True
      KeyServer SCI: 28:99:3a:82:6f:82::605
      SAK transmit: False
      LLPN exhaustion: 0
      Distributed key identifier: None
      Live peer list: ['3dfd4486b5f68a81014a37ec']
      Potential peer list: []

Interface: Ethernet4/3/1
    CKN: abcd
      Message ID: c79ad8882c2dd3a8e838a691
      Elected self: False
      Success: True
      Principal: True
      Default: False
      KeyServer SCI: 28:99:3a:82:6f:82::605
      SAK transmit: True
      LLPN exhaustion: 0
      Distributed key identifier: 9d5bc0d3076ea4a08b99b9d9:1
      Live peer list: ['9d5bc0d3076ea4a08b99b9d9']
      Potential peer list: []

    CKN: dead
      Message ID: 3dfd4486b5f68a81014a37ec
      Elected self: False
      Success: True
      Principal: False
      Default: True
      KeyServer SCI: 28:99:3a:82:6f:82::605
      SAK transmit: False
      LLPN exhaustion: 0
      Distributed key identifier: None
      Live peer list: ['4ef4cf009161bd551b5e7434']
      Potential peer list:
```

About the Output

- **`Connectivity Association Key Name (CKN)`**: Configured name of the key in use.
- **`Message ID`**: A random 92 bit string used as an identifier for an MKA participant.
- **`Elected Self`**: True if this participant is the elected key server.
- **`Success`**: True if this participant is live and has at least one live peer.
- **`Principal`**: True if this participant is the principal participant elected to distribute SAKs or if participant receives SAKs from key server.

- **Default**: True if this participant is a fallback/backup participant (spawned when a fallback key is configured in a Mac Security profile).
- **Key Server SCI**: The SCI of the key server.
- **SAK Transmit**: True if the participant is ready to use the negotiated key for transmit.
- **LLPN Exhaustion**: Increments if the number of data packets sent using the current key exceeds a certain threshold.
- **Distributed Key Identifier**: Message ID + key number of the most recently generated SAK.
- **Live Peer List**: Message IDs of all the live peers of the participant.
- **Potential Peer List**: Message IDs of all the potential peers of the participant. These are peers which havent yet established mutual liveness but have sent out at least one control packet.

9.3.2.3.5 Displaying MACsec MKA Counters

Use the **show mac security mka counters** command to display information about the MACsec MKA counters.

Example

```
switch# show mac security mka counters
Interface       Rx Success      Rx Failure      Tx Success      Tx Failure
Ethernet4/1/1   287             0               288             0
Ethernet4/3/1   288             0               287             0
```

9.3.2.3.6 Displaying MACsec Security Counters Detailed Information

Use the **show mac security counters detail** command to display detailed information about the MACsec security counters.

Example

```
switch# show mac security counters detail
Ethernet4/1/1     Counter Name        Count
-----------------------------------------------------
                  outPktsEncrypted    112
                  outOctetsEncrypted  11984
                  outPktsUntagged     0
                  outPktsTooLong      0
                  outPktCtrl          224
                  inPktsDecrypted     2
                  inOctetsDecrypted   214
                  inPktsUnchecked     0
                  inPktsOK            2
                  inPktsNotValid      0
                  inPktsNotUsingSA    0
                  inPktsCtrl          223
                  inPktsNoTag         8
                  inPktsTagged        0
                  inPktsBadTag        0
                  inPktsNoSCI         0
                  inPktsLate          0

Ethernet4/3/1     Counter Name        Count
-----------------------------------------------------
                  outPktsEncrypted    2
                  outOctetsEncrypted  214
                  outPktsUntagged     0
                  outPktsTooLong      0
                  outPktCtrl          223
                  inPktsDecrypted     111
                  inOctetsDecrypted   11877
```

```
                    inPktsUnchecked     0
                    inPktsOK            111
                    inPktsNotValid      0
                    inPktsNotUsingSA    0
                    inPktsCtrl          224
                    inPktsNoTag         9
                    inPktsTagged        0
                    inPktsBadTag        0
                    inPktsNoSCI         0
                    inPktsLate          0
```

9.3.2.3.7 Displaying MACsec Security Counters

Use the **`show mac security counters`** command to display information about the MACsec security counters.

Example

```
switch# show mac security counters
Port     InPktsDecrypted  InOctetsDecrypted  OutPktsEncrypted OutOctetsEncrypted
Et4/1/1                2                214               109             11663
Et4/3/1              109              11663                 2               214
```

9.3.2.3.8 Displaying MACsec MKA Counters detailed information

Use the **`show mac security mka counters detail`** command to display detailed information about the MACsec MKA counters.

Example

```
switch# show mac security mka counters detail
Interface: Ethernet4/1/1
    Tx packet success: 290
    Tx packet failure: 0
        Tx invalid: 0
    Rx packet success: 289
    Rx packet failure: 0
        Rx invalid: 0
        Rx eapol error: 0
        Rx basic parameter set error: 0
        Rx unrecognized CKN error: 0
        Rx ICV validation error: 0
        Rx live peer list error: 0
        Rx potential peer list error: 0
        Rx SAK use set error: 0
        Rx distributed SAK set error: 0
        Rx distributed CAK set error: 0
        Rx ICV Indicator error: 0
        Rx unrecognized parameter set error: 0

Interface: Ethernet4/3/1
    Tx packet success: 289
    Tx packet failure: 0
        Tx invalid: 0
    Rx packet success: 290
    Rx packet failure: 0
        Rx invalid: 0
        Rx eapol error: 0
        Rx basic parameter set error: 0
        Rx unrecognized CKN error: 0
        Rx ICV validation error: 0
        Rx live peer list error: 0
```

```
        Rx potential peer list error: 0
        Rx SAK use set error: 0
        Rx distributed SAK set error: 0
        Rx distributed CAK set error: 0
        Rx ICV Indicator error: 0
        Rx unrecognized parameter set error: 0
```

9.3.2.3.9 Displaying MACsec FIPS Status

Use the **show mac sec status** command to display information about the MACsec FIPS status.

Example

```
switch(config)# mac security
switch(config-mac-security)# show mac sec status
Active Profiles:          1
FIPS Mode:                Yes
Secured Interfaces:       2
```

9.3.2.3.10 Displaying Information for MACsec Using Static Secure Association Key

If MACsec is configured to use static SAKs, these commands will show additional information related to static SAKs:

- **show active**

  In MAC Security configuration mode, the **show active** command displays the MAC security key source. If one or more static SAKs are configured, this key source will be shown as "key source sak static."
- **show mac security interface**

  With a static SAK configured, the **show mac security interface** command shows the association numbers for SAKs which are programmed for Rx and Tx. **Show** commands never display actual SAK values.

  If a unidirectional link is configured with a static SAK, the Rx side will show the SCI as "00:00:00:00:00:00::0," and only the Rx AN will be shown. On the Tx side, the configured SCI and Tx AN will be shown.
- **show mac security sak**

  If one or more SAKs are configured in the switch, the **show mac security sak** command will show SAK-related details.

9.3.2.4 MACsec Key Retirement Immediate

The MACsec uses the concept of configuring two keys for MKA negotiation: Primary and Fallback (as a backup). Given a mac security profile configured on an interface, there is an actor created per key which is responsible for MKA negotiation with the other peer. When a new primary key is configured, old primary keys actor is retained in the system till the time MKA session becomes successful with the configured new primary key. Same holds good for fallback key as well. When **key retirement immediate** command is used it removes the actor corresponding to old key, be it primary or fallback, from the system immediately.

MACsec Key Retirement Immediate Operations

- If a new primary key is configured in a mac security profile, old primary keys actor is deleted from the system immediately.
- If a new fallback key is configured in a mac security profile, old fallback keys actor is deleted from the system immediately.
- Removing the feature configuration from mac security profile will just prevent cleaning up of old keys immediately when new keys are configured. It will not create old actor again.

Note: The **`key retirement immediate`** command only deletes the actor corresponding to old key. It does not clean up the SAK programmed in the hardware until a new SAK is available to be programmed. However, as a side effect of deletion of actor, a new principal actor will be chosen (if an eligible actor is available) over which a new SAK will be distributed subsequently.

MACsec Key Retirement Immediate Feature Interactions

MACsec EAP-FAST Support

If Dynamic MAC Security keys is used with key retirement immediate, then on every new primary key derived from 802.1X, old primary keys actor will be deleted from the system. This will usually happen based on the reauth time interval configuration for 802.1X.

MACsec Fallback to Unprotected Traffic Support

The key retirement immediate is configured with Fallback to Unprotected Traffic feature, transition between unprotected traffic and protected traffic may become more frequent. This is because with Key Retirement Immediate feature, whenever a new key is configured, existing successful MKA session corresponding to the old key are not maintained, which might bring down the number of successful MKA sessions to zero, which eventually moves the interface to unprotected traffic state as per Fallback to Unprotected Traffic feature functionality.

9.3.2.4.1 MACsec Key Retirement Immediate Configuration

The **`show dot1x supplicant`** command is configured in mac security profile mode, the configuration needs to be present on both key server and non key server peers. Since key server decides the principal actor for SAK distribution, it is recommended that this configuration is present in key server for triggering the re-election of principal actor immediately.

If key retirement immediate is configured only on key server, non key server will still try to negotiate MKA over old primary key unnecessarily utilizing some system resources and some time even when not required.

If key retirement immediate is configured only on non key server, it will take ***6*** seconds (MKA Lifetime) for triggering any re-election on key server as a result of session failure.

```
switch(config-mac-security-profile-sampleProfile)# [no] key retirement
 immediate
```

Configuration Scenarios

When both Primary Key and Fallback Key configured: without configuring key retirement immediate, when a new primary is configured, the actor corresponding to the old actor will stay active till MKA session on the new primary becomes successful. With key retirement immediate, the actor corresponding to the old primary is deleted immediately. Since fallback is also configured, key server will choose it as the new principal actor, if eligible. Once a new principal actor is chosen, new SAK is distributed which will eventually get programmed and used for encryption & decryption.

When only Primary Key is configured: the behavior is same as above except the fact that no other actor will become principal until the new primary becomes successful. Till then hardware will continue to use SAK generated with old primary.

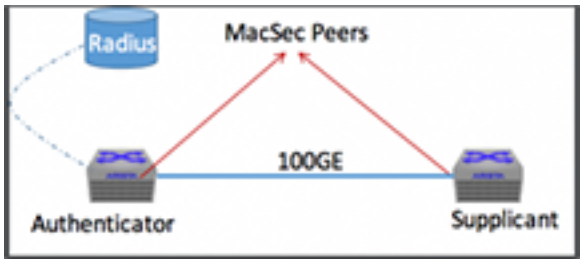
When Fallback is Principal actor: without key retirement immediate, when a new fallback key is configured, old fallback will stay in the system till the time new fallback becomes active or primary becomes active. With key retirement immediate, old fallback actor is deleted immediately. Till the time a new principal actor is elected, hardware will continue to use SAK generated with old fallback.

The **show mac security participants** command shows all the participants present in the system. When key retirement immediate is configured, the actor corresponding to old keys will no longer list up in the output of the above **show** command.

9.3.2.5 MACsec EAP-FAST Support

The Media Access Control Security (MACsec) with static keys feature brings support for dynamic Mac Security keys. To derive Mac Security keys dynamically, both peers must be configured for 802.1X authentication. One peer must be configured to be the Authenticator and the other peer to be the Supplicant. Upon a successful 802.1X authentication sequence between the peers, keying material is generated by both the authenticator and the supplicant. This keying material is then used to derive Mac Security keys to establish a MACsec Key Agreement (MKA) protocol session. This feature brings in support for Arista devices to act as the supplicant to derive Mac Security keys in a bidirectional fashion.

The following diagram illustrates a typical Mac Security + 802.1X topology:



9.3.2.5.1 Configuring MAC Security Dynamic Key Derivation

9.3.2.5.1.1 802.1X Authenticator Configuration

A new option is added to 802.1X authenticator configuration to make the authenticator more strong to unreliable authentication servers. By default, when an authentication server is unreachable, the authenticator blocks all traffic on the port and keeps the port as Unauthorized until it gets replies from the authentication server. The following option changes the behavior and maintains the port in its current state if the authentication server is not reachable:

Example

```
switch(config-if-Et1)# dot1x timeout reauth-timeout-ignore always
```

9.3.2.5.1.2 802.1X Supplicant Configuration

The 802.1X supplicant configurations are done through MACsec profiles. MACsec profile contain all the credentials necessary for 802.1X authentication to succeed.

Following are the steps to configure an 802.1X supplicant profile:

1. Use **dot1x** command to enter the dot1x mode to configure a supplicant profile.

   ```
   switch(config)# dot1x
   switch(config-dot1x)#
   ```

2. Use **supplicant profile** command to configure a 802.1X supplicant profile.

   ```
   switch(config)# supplicant profile <profileName>
   ```

   The following mandatory commands must be configured for a supplicant profile to be operational:

3.

4. An Extensible Authentication Protocol (EAP) method must be configured for the profile. The only method supported by Arista supplicants is EAP-FAST.

```
switch(config-dot1x-supp-profile-test)# eap-method fast
```

   a. Configure EAP Identity which is used to authenticate the supplicant with the Radius server:

```
switch(config-dot1x-supp-profile-test)# identity <user-identity>
```

   b. Configure EAP pass-phrase the password used to authenticate the supplicant with the Radius server:

```
switch(config-dot1x-supp-profile-test)# passphrase <options>
```

Example

This is an sample 802.1X supplicant profile:

```
switch(config-dot1x-supp-profile-test)# show active
dot1x
   supplicant profile test
      identity arista
      passphrase 7 070E334D5D1D0B04
```

Apply the supplicant profile by enabling it on the Mac Security interface:

```
switch(config-if-Et6/1)# dot1x pae supplicant test
```

Mac Security

Mac Security configuration remains the same as described in the configuration guide with a single important difference. Instead of configuring manual keys, a Mac security profile must instead be configured to use dynamic keys:

```
switch(config-mac-security-profile-test)# key source dot1x
```

9.3.2.5.2 Displaying 802.1X Supplicant Status

The **`show dot1x supplicant`** command displays the 802.1X supplicant status.

```
switch #show dot1x supplicant
Interface: Ethernet6/1
    Identity: arastra
    EAP method: fast
    Status: success
    Supplicant MAC: 44:4c:a8:34:bf:20
    Authenticator MAC: 00:1c:73:e0:d3:76
```

About the Output

- **`Interface`**: The port on which the supplicant is running.
- **`Identity`**: Configured supplicant identity.
- **`EAP method`**: Configured EAP method (Currently just EAP-FAST).
- **`Status`**: Supplicant Status. Can be one of the following:
  - Success Authentication has been successful.
  - Down Authentication sequence has not begun.
  - Failed Authentication has failed.
  - Connecting Authentication is in progress.

- Unused Supplicant is uninitialized.
- **`Supplicant MAC`**: MAC address of the supplicant.
- **`Authenticator MAC`**: MAC address of the authenticator (peer).
- **`Existing Mac Security`**: Show commands can be used to look at Mac Security status.

9.3.2.6 MACsec Proxy For VXLAN

The MACsec Proxy for VXLAN feature enables the MACsec service over VXLAN. MACsec over VXLAN is provided by mapping a Visual Networking Index (VNI), Remote VXLAN Tunnel Endpoint (VTEP) IP to a MACsec proxy sub interface.

Any packets routed to the MACsec proxy sub interface is encrypted and tunneled to the remote VTEP. On the receiving path the packets are decrypted, then decapsulated and forwarded. MKA negotiates and renews the encryption keys, for this purpose a MACsec capable front panel port has to be dedicated and cannot be plugged in as it will be used to recycle packets being encrypted and decrypted.

9.3.2.6.1 Configuring MACsec Proxy For VXLAN

The switch platforms which use this feature are:

- 7280SRAM-48C6
- 7280CR2M-30
- 7500R2M-36CQ-LC

The mandatory steps to configure a MACsec proxy sub-interface on an Arista switch are:

1. Configure the parent interface to be a routed port.
2. Create a L3 sub-interface on the parent interface. This is the MACsec proxy sub-interface.
3. Create a L2 sub-interface on the parent interface. This is the MACsec patch sub-interface.
4. Configure and enable the MACsec proxy port on a sub-interface.
5. Configure the VXLAN tunnel.
6. Assign the forwarding VLAN ID for the MACsec patch sub-interface and VXLAN tunnel.

Example Configurations

a. Configure a ***100g*** MACsec interface as a routed port.

```
switch(config)# interface et49/1
switch(config-if-Et49/1)# no switchport
```

b. Create a new L3 sub-interface - ***et49/1.1*****.**

```
switch(config-if-Et49/1)# interface et49/1.1
```

c. Create a new L2 sub-interface - ***et49/1.2***.

```
switch(config-if-Et49/1)# interface et49/1.2
```

d. Configure the MACsec proxy port, and enable MACsec on the proxy port.

```
switch(config)# interface et49/1.1
switch(config-if-Et49/1.1)# mac security proxy patch Ethernet49/1.2
switch(config-if-Et49/1.1)# mac security profile test1
switch(config-if-Et49/1.1)# ip address 2.2.2.1/24
```

e. Configure the VXLAN tunnel. The remote VTEP is provided as the flood VTEP.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan source-interface Loopback0
switch(config-if-Vx1)# vxlan udp-port 4789
```

```
switch(config-if-Vx1)# vxlan vlan 20 vni 20
switch(config-if-Vx1)# vxlan vlan 20 flood vtep 100.100.100.2
```

f. Configure the L2 MACsec patch interface to be in the same VLAN as VXLAN.

```
switch(config)# interface et49/1.2
switch(config-if-Et49/1.2)# vlan id 20
```

9.3.2.6.2 Displaying MACsec Proxy For VXLAN Information

Use **`show mac security interface`** command to display the proxy sub-interface information.

Examples

- Use **`show mac security mka counters`** command to display the MACsec counters and detailed values.

```
switch(config)# show mac security interface
Interface     SCI                      Controlled Port  Key in Use
Ethernet4/1/1 28:99:3a:82:6f:82::605  True             9d5bc0d3076ea4a08b99b9d9:1
Ethernet4/3/1 28:99:3a:82:6f:85::613  True             9d5bc0d3076ea4a08b99b9d9:1
```

-

```
switch(config)# show mac security mka counters
Interface      Rx Success  Rx Failure  Tx Success   Tx Failure
Ethernet4/1/1  287         0            288          0
Ethernet4/3/1  288         0            287          0

switch(config)# show mac security mka counters ethernet 49/1.1 detail
Interface: Ethernet49/1.1
    Tx packet success: 84
    Tx packet failure: 0
        Tx invalid: 0
    Rx packet success: 82
    Rx packet failure: 0
        Rx invalid: 0
        Rx eapol error: 0
        Rx basic parameter set error: 0
        Rx unrecognized CKN error: 0
        Rx ICV validation error: 0
        Rx live peer list error: 0
        Rx potential peer list error: 0
        Rx SAK use set error: 0
        Rx distributed SAK set error: 0
        Rx distributed CAK set error: 0
        Rx ICV Indicator error: 0
        Rx unrecognized parameter set error: 0
```

9.3.2.6.3 Limitations

When this feature is in use, following limitations can be noticed:

- An interface while moving from allowing unprotected traffic to allowing only protected traffic can experience a traffic disruption of up to ***4*** seconds.
- If the key server interface manages to establish a MKA session with its old credentials (CKN/CAK pair) while unprotected traffic was allowed, then traffic disruption for a duration of up to ***6*** seconds can be noticed in addition to the duration mentioned in the above point.

9.3.2.7 MACsec Fallback to Unprotected Traffic

When MACsec is enabled on an interface, it tries to establish MACsec Key Agreement (MKA) session(s) with its peer. If no MKA sessions is successfully established, then the interface can continue to protect the traffic with the last known negotiated key, and if such a key does not exist then it blocks the traffic. The MACsec Fallback to Unprotected Traffic feature introduces an optional configuration which, if provided, allows unprotected traffic whenever there is no successful MKA session with the peer in the following scenarios:

- If MACsec is enabled on an interface with this feature configured, then the interface allows unprotected traffic immediately without waiting for MKA session establishment.
- If a MACsec enabled interface was blocking traffic as no MKA sessions were established and its corresponding MACsec profile is changed to enable this feature, the interface will start allowing unprotected traffic immediately.
- If a MACsec enabled interface was allowing unprotected traffic and its corresponding MACsec profile is changed to disable this feature, the interface will block traffic immediately.
- While an interface is allowing unprotected traffic, it will stop doing so when a new Secure Association Key (SAK) is generated (if this interface is key server) or when a SAK is received from the key-server (if this interface is not the key server).
- If MACsec Fallback to Unprotected Traffic is configured and all MKA sessions between the peers fail, the peers will switch to unprotected traffic. If not configured, protected traffic could have continued with last known negotiated key.

To protect traffic between pairs, primary MKA session derived keys are given priority over Fallback MKA session. With this feature enabled, the priority order of traffic between peers is -

1. Protected using derived keys from primary MKA sessions.

2. Protected using derived keys from Fallback MKA sessions.

3. Unprotected traffic.



Note: Arista allows a primary and a Fallback Connectivity Association Key (CAK) and Connectivity Association Key Name (CKN) pair to be configured on an interface. And interfaces tries to establish a MKA session with its peer corresponding to each CAK/CKN pair.

9.3.2.7.1 MACsec Fallback to Unprotected Traffic Feature Interaction

This feature interacts with other related features in following way:

- **MACsec EAP-FAST Support**: If dynamic MAC Security keys (derived from 802.1X authentication) are used, then the feature configuration has no effect.
- **MACsec Proxy Interfaces**: This feature does not work with MACsec proxy sub interfaces.
- **Key Retirement Immediate**: If this feature is configured with Key Retirement Immediate feature on an interface, transition between unprotected traffic and protected traffic may become more frequent. This is because with Key Retirement Immediate feature, whenever a new key is configured, existing successful MKA session corresponding to the old key is not maintained.

9.3.2.7.2 Limitations

When this feature is in use, following limitations can be noticed:

- An interface while moving from allowing unprotected traffic to allowing only protected traffic can experience a traffic disruption of up to ***4*** seconds.
- If the key server interface manages to establish a MKA session with its old credentials (CKN/CAK pair) while unprotected traffic was allowed, then traffic disruption for a duration of up to ***6*** seconds can be noticed in addition to the duration mentioned in the above point.

9.3.2.7.3 Configuring MACsec Fallback to Unprotected Traffic

This feature is supported on all MACsec capable cards except for 7500E-6CFPX-LC.

The MACsec Fallback to Unprotected Traffic feature is configured under MACsec profile mode using the **`traffic unprotected allow`** command. The **no** form of the command removes the configuration from the switch. This configuration must be present in both the peers for the unprotected traffic to flow between them successfully.

Example

```
switch(config-mac-security-profile-sampleProfile)# no traffic unprotected allow
```

9.3.2.7.4 Displaying MACsec Fallback to Unprotected Traffic Information

The **show mac security interface detail** command can be used to verify if the interface is currently allowing unprotected traffic.

```
switch# show mac security interface Ethernet 6/1/1 detail
Interface: Ethernet4/1/1
    SCI: 28:99:3a:82:6f:82::605
    SSCI: 00000002
    Controlled port: True
    Key server priority: 16
    Session rekey period: 0
    Traffic: Unprotected
    Key in use: 9d5bc0d3076ea4a08b99b9d9:1
    Latest key: None
    Old key: 9d5bc0d3076ea4a08b99b9d9:1(RT)

Interface: Ethernet4/3/1
    SCI: 28:99:3a:82:6f:85::613
    SSCI: 00000001
    Controlled port: True
    Key server priority: 16
    Session rekey period: 0
    Traffic: Protected
    Key in use: 9d5bc0d3076ea4a08b99b9d9:1
    Latest key: None
    Old key: 9d5bc0d3076ea4a08b99b9d9:1(RT)
```

9.3.2.8 MACsec Commands

MACsec Configuration Commands

- an (MACsec)
- cipher
- entropy source hardware
- identifier (MACsec)
- key (MACsec)
- key retirement immediate
- license (Global Mode)
- license (MACsec)
- mac security
- mka key-server
- mka session
- profile (MACsec)
- replay
- sci
- secure channel (MACsec)
- traffic unprotected allow

MACsec Profile on a Subinterface

- mac security profile

MACsec Show Commands

- show mac security counters
- show mac security counters detail
- show mac security interface
- show mac security interface detail
- show mac security mka counters
- show mac security participants
- show mac security participants detail
- show mac security profile
- show mac security sak
- show mac security status

MACsec EAP FAST Support Commands

- dot1x
- dot1x pae supplicant
- dot1x timeout reauth-timeout-ignore always
- show dot1x supplicant
- supplicant profile

9.3.2.8.1 an (MACsec)

The **an** command defines an Association Number (AN) and a Secure Association Key (SAK) for use in the selected channel in MACsec. Up to 4 SAKs can be configured in the Rx direction, with ANs ranging from 0 to 3. The Tx channel can only have one AN and one SAK. The **no an** and **default an** commands remove the specified AN and its SAK from ***running-config***.

Command Mode

MAC Security Profile SAK Static Secure Channel Configuration

Command Syntax

an ***an_number*** **key** ***key_type key_string***

no an ***an_number***

default an ***an_number***

Parameters

- ***an_number*** The Association Number. For the Rx channel, values range from 0 to 3. For the Tx channel, the only allowed value is 0. There is no default value.
- ***key_type*** The type of string specifying the SAK. There are three valid key types:
  - **0** indicates that the key string which follows is not encrypted.
  - **7** indicates that the key string which follows is hidden or obfuscated.
  - **8a** The following key is encrypted with AES-256-GCM.
- ***key_string*** The Secure Association Key itself, in hexadecimal octets.

Example

These commands add a static SAK with ***AN 1*** to the Rx channel for profile ***test***.

```
switch(config)# mac security
switch(config-mac-security)# profile test
switch(config-mac-security-profile-test)# key source sak static
switch(config-mac-security-profile-test-sak-static)# secure channel rx
switch(config-mac-security-profile-test-sak-static-rx)# an 1 key 0
 11112222333344445555666677778888
```

```
switch(config-mac-security-profile-test-sak-static-rx)#
```

9.3.2.8.2 cipher

The **cipher** command configures the cipher authentication for MAC security on the switch.

Command Mode

MACsec Profile

Command Syntax

cipher ***encryption_standard***

Parameters

encryption_standard The cipher authentication options.

- **aes128-gcm-xpn** Advanced Encryption Standard (128 bit, Galois/Counter mode, Extended Packet Numbering).
- **aes256-gcm-xpn** Advanced Encryption Standard (256 bit, Galois/Counter mode, Extended Packet Numbering).

Example

The following command configures the **cipher aes128-gcm-xpn** for MAC security on the switch for the MACsec profile called test.

```
switch(config-mac-security-profile-test)# cipher aes128-gcm-xpn
switch(config-mac-security-profile-test)#
```

9.3.2.8.3 dot1x pae supplicant

The **dot1x pae supplicant** command applies the supplicant profile by enabling it on the Mac Security interface.

Command Mode

Interface Configuration

Command Syntax

dot1x pae supplicant

Example

The following command applies the supplicant profile test on the ***MACsec interface 6/1***.

```
switch(config-if-Et6/1)# dot1x pae supplicant test
```

9.3.2.8.4 dot1x timeout reauth-timeout-ignore always

The **dot1x timeout reauth-timeout-ignore always** command retains the current port state without blocking it irrespective of when the authentication server is unreachable or in-case of supplicant time outs.

Command Mode

Interface Configuration

Command Syntax

dot1x timeout reauth-timeout-ignore always

Example

The following command retains the current port status of ***interface Ethernet 6/1*** when there is authentication server timeout.

```
switch(config-if-Et6/1)# dot1x timeout reauth-timeout-ignore always
```

9.3.2.8.5 dot1x

The **dot1x** command places the switch in the dot1x mode. In this mode user is allowed to configure various MACsec configurations.

Command Mode

Global Configuration

Command Syntax

dot1x

Example

The following command places the switch in the ***dot1x*** mode.

```
switch(config)# dot1x
switch(config-dot1x)#
```

9.3.2.8.6 entropy source hardware

The **entropy source hardware** command generates the cryptographic keys to strengthen the random number generator used by MACsec.

Command Mode

Management Configuration

Command Syntax

entropy source hardware

Example

The following command configures the entropy source hardware and generates the cryptographic keys.

```
switch(config)# management security
switch(config-mgmt-security)# entropy source hardware
```

9.3.2.8.7 identifier (MACsec)

The **identifier** command defines a Secure Channel Identifier (SCI) for the Rx or Tx secure channel for use with MACsec static SAKs. The SCI is a MAC address in the format H:H:H:H:H:H::P, where H is a hexadecimal octet and P is a decimal integer. The **no identifier** and **default identifier** commands remove the channel's SCI from ***running-config***.

Command Mode

MAC Security Profile SAK Static Secure Channel Configuration

Command Syntax

identifier ***MAC_address***

no identifier

default identifier

Parameters

- ***MAC_address*** The MAC address identifying the secure channel.

Example

These commands add the SCI 01:02:03:04:05:06::1234 to the Rx channel for profile "test".

```
switch(config)# mac security
switch(config-mac-security)# profile test
switch(config-mac-security-profile-test)# key source sak static
switch(config-mac-security-profile-test-sak-static)# secure channel rx
switch(config-mac-security-profile-test-sak-static-rx)# identifier
 01:02:03:04:05:06::1234
switch(config-mac-security-profile-test-sak-static-rx)#
```

9.3.2.8.8 key (MACsec)

The **key** command configures the primary key so that the MACsec profile is activated.

Note: Optionally a fallback CAK can also be configured on a profile. This CAK is picked up by MACsec to negotiate keys if the primary CAK fails. A CAK can be configured as a backup key using the fallback keyword with the key command.

Command Mode

MACsec Profile Configuration

Command Syntax

key <options>

Parameters

- **CKN** Connectivity association key name in hex octets. Options include:
  - ***0*** Specifies that an UNENCRYPTED key will follow.
  - ***7*** Specifies that an HIDDEN key will follow.
  - **CAK** Connectivity association key in hex octets.
  - **fallback** Configure the key as a fallback.
- **retirement** Retire the key. Options include:
  - **immediate** Retire the key immediately.
- **source** List of sources to derive MAC security keys. Options include:
  - **dot1xDerive** MAC security keys from IEEE 802.1X based port authentication
  - **group-cak** Derive MAC security keys from Group CAK Distribution.
  - **sak static** Enter

Examples

- The following example configures the primary key for the profile called sample profile for MAC security on the switch.

```
switch(config)# mac security
switch(config-mac-security)# profile sample_Profile
switch(config-mac-security-profile-sample_Profile)# key 0abcd1 0 1234abcd
```

- The following example configures the fallback CAK on a profile.

```
switch(config)# mac security
switch(config-mac-security)# profile sample_Profile
switch(config-mac-security-profile-sample_Profile)# key 0abcd1 0 1234abcd
 fallback
```

9.3.2.8.9 key retirement immediate

The **key retirement immediate** command configures the key retirement feature on the key server and assists the key server to decide the principal actor for SAK distribution by triggering the re-election of principal actor immediately. It is recommended that the key retirement is configured on both key server and non key server peers.

The **no key retirement immediate** command disable the key retirement function by removing the key retirement immediate command from the ***running-config***.

Command Mode

MACsec Profile

Command Syntax

key retirement immediate

Example

The following commands configures the key retirement immediate feature on a switch for a MACsec profile called ***sample***.

```
switch(config)# mac security
switch(config-mac-security)# profile sample
switch(config-mac-security-profile-sample)# key retirement immediate
```

9.3.2.8.10 license (Global Mode)

The **license** command configures EOS licenses on the switch under the global configuration mode. These licenses include the MACsec license.



Note: Contact your system engineer to acquire the required license codes before attempting to configure MACsec.

Command Mode

Global Configuration

Command Syntax

license {**import** ***URL*** | **update**}

Parameters

- **import** Import license from a URL.
- ***URL*** The URL from which to import a license.
- **update** Trigger a check for licenses.

Example

The following example configures the MACsec license on the switch using a JSON file as shown.

```
switch# license import flash:EOSLic-1.json
switch#
```

9.3.2.8.11 license (MACsec)

The **license** command configures the MACsec license on the switch under the MAC Security configuration mode using a hex key.

The **no license** and **default license** commands delete the current license from ***running-config***.



Note: This method of license configuration is no longer being used except for backward compatibility.

Command Mode

MAC Security

Command Syntax

`license` ***licensee_name license_value***

Parameters

- ***licensee_name*** Name of the licensee.
- ***license_value*** 8 digit hexadecimal key to authorize MAC security.

Example

The following example configures the MACsec license on the switch using an 8 digit hexadecimal key.

```
switch(config)# mac security
switch(config-mac-security)# license Test-LICNC AABBCCDD
switch(config-mac-security)#
```

9.3.2.8.12 mac security

The **`mac security`** command enables MAC security provision on the switch.

The **`no mac security`** and **`default mac security`** commands restore the switch to its default state by removing the corresponding mac security command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`mac security`

`no mac security`

`default mac security`

Example

The following command places the switch in MAC security mode.

```
Switch(config)# mac security
Switch(config-mac-security)#
```

9.3.2.8.13 mac security profile

The **`mac security profile`** command applies a MACsec profile to an interface or subinterface.

The **`no mac security profile`** and **`default mac security profile`** commands remove the MACsec profile, disabling MACsec on the configuration-mode interface.

Command Mode

Interface Ethernet Configuration Mode

Command Syntax

`mac security profile` ***profile-name***

`no mac security profile` ***profile-name***

`default mac security profile` ***profile-name***

Parameter

- ***profile-name*** the MACsec profile name.

Example

- The following commands enable MACsec on Ethernet subinterface ***1.10*** by applying the MACsec profile called ***test-profile***.

```
switch(config)# interface ethernet1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# interface ethernet1.10
switch(config-if-Et1.10)# encapsulation dot1q vlan 20
switch(config-if-Et1.10)# mac security profile test-profile
```

9.3.2.8.14 mka key-server

The **`mka key-server`** command configures key server among the MACsec peers.

Command Mode

MACsec Profile Configuration

Command Syntax

`mka key-server` priority ***value***

Parameters

- **priority** MKA key server priority.
- ***value*** Key server priority value. Value ranges from ***0*** to ***255***.

Example

The following example configures the key server value of ***10*** among the MACsec peers.

```
Switch(config)# mac security
Switch(config-mac-security)# profile sample_Profile
Switch(config-mac-security-sample_Profile)# mka key-server priority 10
```

9.3.2.8.15 mka session

The **`mka session`** command configures period at which the SAK is refreshed .

Command Mode

MACsec Profile Configuration

Command Syntax

`mka session` rekey-period ***value***

Parameter

- **rekey-period** Sets MKA session re-key period.
- ***value*** Session re-key period in seconds. Value ranges from ***30*** to ***100000***.

Example

The following example configures the mka session rekey-period time of ***10*** seconds at which the SAK is refreshed.

```
Switch(config)# mac security
Switch(config-mac-security)# profile sample_Profile
Switch(config-mac-security-sample_Profile)# mka session rekey-period 10
```

9.3.2.8.16 profile (MACsec)

The **profile** command places the switch in MAC Security Profile configuration mode and creates a MACsec profile if a profile of the specified name does not already exist. MACsec profiles contain the configuration information needed to establish a MACsec connection, and are applied to interfaces using the **mac security profile** command.

Command Mode

MAC Security Configuration

Command Syntax

profile ***profile-name***

Parameter

profile-name Name of the MACsec profile.

Commands Available in MAC Security Profile Configuration Mode

- cipher
- key (MACsec)
- mka key-server
- mka session
- replay
- sci
- traffic unprotected allow

Example

The following commands create a MACsec profile called ***test*** and place the switch in MAC Security Profile configuration mode for that profile.

```
switch(config)# mac security
switch(config-mac-security)# profile test
switch(config-mac-security-profile-test)#
```

9.3.2.8.17 replay

The **replay** command configures the action to be taken when packets received are not in order, based on their packet numbers. The window size in replay protection specifies the window size within which out-of-order packets are allowed. This command is configured under the MACsec Profile configuration mode.

The **no** and **default** form of the command removes all the configurations related to `replay` command from the running configuration on the switch.

Command Mode

MACsec Profile

Command Syntax

replay protection {**disabled** | **window** ***window_size***}

no replay protection {**disabled** | **window** ***window_size***}

default replay protection {**disabled** | **window** ***window_size***}

Parameters

- protection Specifies the action to be taken when packets received are not in order, based on their packet numbers..
- **disabled** Disables replay protection.
- **window** Specifies the allowable window within which an out-of-order packet can be received.

- ***window_size*** The allowable value ranges from ***0*** tthrough **4294967295**.

Example

The following commands configures a MACsec profile called TEST and a replay protection with a window size of ***100*** is configured on the switch.

```
switch(config)# mac security
switch(config-mac-security)# profile TEST
switch(config-mac-security-profile-TEST)# replay protection window 100
```

9.3.2.8.18 sci

The **sci** command add a Secure Channel Identifier (SCI) in data packets for MACsec on the switch. Each MACsec device has a Secure Channel (SC) used to send traffic to other device. Each channel has an 8-byte Secure Channel Identifier (SCI). The first 6 bytes match the MAC address of the device transmitting through that channel. The remaining 2 bytes are a Port Identifier used to distinguish between multiple channels from the same device. The command is configured under the MACsec profile configuration mode.

Command Mode

MACsec Profile

Command Syntax

sci

Example

The following commands place the switch on MACsec profile configuration mode and add a SCI for the MACsec profile called **TEST**.

```
switch(config)# mac security
switch(config-mac-security)# profile TEST
switch(config-mac-security-profile-TEST)# sci
```

9.3.2.8.19 secure channel (MACsec)

The **secure channel** command enters MAC Security Profile Static SAK Secure Channel configuration mode. In this mode, you can add Association Numbers (AN) and Secure Channel Identifiers (SCI) for the specified channel. The available channels are Rx (receive) and Tx (transmit).

Command Mode

MAC Security Profile Static SAK Configuration Mode

Command Syntax

secure channel {**Rx**|**Tx**}

Parameters

- **Rx** Enter the configuration mode for the Rx channel.
- **Tx** Enter the configuration mode for the Tx channel.

Available Commands

- **an**
- **identifier**

Example

These commands enter MAC Security Profile Static SAK Secure Channel configuration mode for the Tx channel.

```
switch(config)# mac security
switch(config-mac-security)# profile test
switch(config-mac-security-profile-test)# key source sak static
switch(config-mac-security-profile-test-sak-static)# secure channel tx
switch(config-mac-security-profile-test-sak-static-sc-tx)#
```

9.3.2.8.20 show dot1x supplicant

The **show dot1x supplicant** command displays the 802.1X supplicant status.

Command Mode

EXEC

Command Syntax

show dot1x supplicant

Example

The following example displays information about 802.1X supplicant status.

```
switcb# show dot1x supplicant

Interface: Ethernet6/1
    Identity: arastra
    EAP method: fast
    Status: success
    Supplicant MAC: 44:4c:a8:34:bf:20
    Authenticator MAC: 00:1c:73:e0:d3:76
```

About the Output

- **Interface:** The port on which the supplicant is running.
- **Identity:** Configured supplicant identity.
- **EAP method:** Configured EAP method (Currently just EAP-FAST).
- **Status:** Supplicant Status. Can be one of the following:
  - Success Authentication has been successful.
  - Down Authentication sequence has not begun.
  - Failed Authentication has failed.
  - Connecting Authentication is in progress.
  - Unused Supplicant is uninitialized.
- **Supplicant MAC:** MAC address of the supplicant.
- **Authenticator MAC:** MAC address of the authenticator (peer). Existing Mac Security show commands can be used to look at Mac Security status.

9.3.2.8.21 show mac security counters detail

The **show mac security counters detail** command to displays the detail information about the MACsec security counters.

Command Mode

EXEC

Command Syntax

show mac security counters detail

Example

The following example displays detail information about MACsec security counters.

```
switch# show mac security counters detail
Ethernet4/1/1     Counter Name        Count
-----------------------------------------------------
                  outPktsEncrypted    112
                  outOctetsEncrypted  11984
                  outPktsUntagged     0
                  outPktsTooLong      0
                  outPktCtrl          224
                  inPktsDecrypted     2
                  inOctetsDecrypted   214
                  inPktsUnchecked     0
                  inPktsOK            2
                  inPktsNotValid      0
                  inPktsNotUsingSA    0
                  inPktsCtrl          223
                  inPktsNoTag         8
                  inPktsTagged        0
                  inPktsBadTag        0
                  inPktsNoSCI         0
                  inPktsLate          0

Ethernet4/3/1     Counter Name        Count
-----------------------------------------------------
                  outPktsEncrypted    2
                  outOctetsEncrypted  214
                  outPktsUntagged     0
                  outPktsTooLong      0
                  outPktCtrl          223
                  inPktsDecrypted     111
                  inOctetsDecrypted   11877
                  inPktsUnchecked     0
                  inPktsOK            111
                  inPktsNotValid      0
                  inPktsNotUsingSA    0
                  inPktsCtrl          224
                  inPktsNoTag         9
                  inPktsTagged        0
                  inPktsBadTag        0
                  inPktsNoSCI         0
                  inPktsLlate         0
```

9.3.2.8.22 show mac security counters

The **`show mac security counters`** command to displays information about the MACsec security counters.

Command Mode

EXEC

Command Syntax

`show mac security counters`

Example

The following example displays information about MACsec security counters.

```
switch# show mac security counters
Port       InPktsDecrypted  InOctetsDecrypted  OutPktsEncrypted OutOctetsEncrypted
Et4/1/1                  2                214               109             11663
```

```
Et4/3/1               109             11663               2               214
```

9.3.2.8.23 show mac security interface detail

The **show mac security interface detail** command displays the detail information about the MACsec on the interface.

Command Mode

EXEC

Command Syntax

show mac security interface detail

Example

The following example displays detail information about MACsec on the interface.

```
switch# show mac security interface detail
Interface: Ethernet4/1/1
    SCI: 28:99:3a:82:6f:82::605
    SSCI: 00000002
    Controlled port: True
    Key server priority: 16
    Session rekey period: 0
    Traffic: Protected
    Key in use: 9d5bc0d3076ea4a08b99b9d9:1
    Latest key: None
    Old key: 9d5bc0d3076ea4a08b99b9d9:1(RT)

Interface: Ethernet4/3/1
    SCI: 28:99:3a:82:6f:85::613
    SSCI: 00000001
    Controlled port: True
    Key server priority: 16
    Session rekey period: 0
    Traffic: Protected
    Key in use: 9d5bc0d3076ea4a08b99b9d9:1
    Latest key: None
    Old key: 9d5bc0d3076ea4a08b99b9d9:1(RT)
```

About the Output

- **Interface**: Name of the interface.
- **Secure Channel Identifier (SCI)**: Combination of MAC address and port number. Used to uniquely identify a Mac Security port.
- **Controlled Port**: Indicates if Mac Security is enabled on the port. A value of True indicates that encryption is enabled on the port.
- **Key In Use**: The SAK identifier currently in use. Combination of Key Servers message identifier (see below) and key number.
- **Key Server priority**: Configured key server priority.
- **Session Rekey Period**: Configured session rekey period.
- **Latest Key**: Latest SAK being negotiated by Mac Security Key Agreement Protocol (MKA)
- **Old Key**: The last SAK negotiated by Mac Security Key Agreement Protocol (MKA)



Note: Latest and Old key are MKA protocol specific terminology and are used to refer to the last two keys in use. For all practical purposes, Key In Use field is used to identify the current key.

9.3.2.8.24 show mac security interface

The **show mac security interface** command shows information about MACsec on the interface.

Command Mode

EXEC

Command Syntax

show mac security interface

Example

The following example displays information about MACsec on the interface.

```
switch# show mac security interface
Interface     SCI                    Controlled Port Key in Use
Ethernet4/1/1 28:99:3a:82:6f:82::605  True            9d5bc0d3076ea4a08b99b9d9:1
Ethernet4/3/1 28:99:3a:82:6f:85::613  True            9d5bc0d3076ea4a08b99b9d9:1 switch#
```

The following example displays the association numbers (ANs) of SAKs for both Rx and Tx on the interface ***Ethernet9/1***. Actual SAK values are never displayed in ***show*** command output.

```
switch# show mac security interface
Interface       SCI                       Controlled Port      Key in Use
Ethernet9/1     01:02:03:04:05:06::1235   True                 static SAK: Rx
 AN: 0,1 Tx AN: 0 switch#
```

The following example displays MACsec information for a unidirectional link. On the Rx side, the SCI is shown as ***00:00:00:00:00:00::0***, and only the Rx AN is shown.

```
switch# show mac security interface
Interface       SCI                       Controlled Port      Key in Use
Ethernet9/1     00:00:00:00:00:00::0000   True                 static SAK: Rx
 AN: 0 switch#
```

The following example displays MACsec information on the Tx side of a unidirectional link. In this case, the configured SCI is shown, along with the Tx AN.

```
switch(config)# show mac security interface
Interface       SCI                       Controlled Port      Key in Use
Ethernet9/1     01:02:03:04:05:06::1235   True                 static SAK: Tx
 AN: 0
```

9.3.2.8.25 show mac security mka counters

The **show mac security mka counters** command to display information about the MACsec MKA counters.

Command Mode

EXEC

Command Syntax

show mac security mka counters

Example

The following example displays information about MACsec MKA counters.

```
switch# show mac security mka counters
Interface       Rx Success      Rx Failure      Tx Success      Tx Failure
Ethernet4/1/1   287             0               288             0
Ethernet4/3/1   288             0               287             00
```

9.3.2.8.26 show mac security participants detail

The **show mac security participants detail** command displays detail information about the MACsec participants.

Command Mode

EXEC

Command Syntax

show mac security participants detail

Example

The following example displays information about MACsec participants details.

```
switch# show mac security participants detail
Interface: Ethernet4/1/1
    CKN: abcd
      Message ID: 9d5bc0d3076ea4a08b99b9d9
      Elected self: True
      Success: True
      Principal: True
      Default: False
      KeyServer SCI: 28:99:3a:82:6f:82::605
      SAK transmit: True
      LLPN exhaustion: 0
      Distributed key identifier: 9d5bc0d3076ea4a08b99b9d9:1
      Live peer list: ['c79ad8882c2dd3a8e838a691']
      Potential peer list: []

    CKN: dead
      Message ID: 4ef4cf009161bd551b5e7434
      Elected self: True
      Success: True
      Principal: False
      Default: True
      KeyServer SCI: 28:99:3a:82:6f:82::605
      SAK transmit: False
      LLPN exhaustion: 0
      Distributed key identifier: None
      Live peer list: ['3dfd4486b5f68a81014a37ec']
      Potential peer list: []

Interface: Ethernet4/3/1
    CKN: abcd
      Message ID: c79ad8882c2dd3a8e838a691
      Elected self: False
      Success: True
      Principal: True
      Default: False
      KeyServer SCI: 28:99:3a:82:6f:82::605
      SAK transmit: True
      LLPN exhaustion: 0
      Distributed key identifier: 9d5bc0d3076ea4a08b99b9d9:1
      Live peer list: ['9d5bc0d3076ea4a08b99b9d9']
      Potential peer list: []

    CKN: dead
      Message ID: 3dfd4486b5f68a81014a37ec
      Elected self: False
      Success: True
      Principal: False
      Default: True
```

```
      KeyServer SCI: 28:99:3a:82:6f:82::605
      SAK transmit: False
      LLPN exhaustion: 0
      Distributed key identifier: None
      Live peer list: ['4ef4cf009161bd551b5e7434']
      Potential peer list:
```

About the Output

- **Connectivity Association Key Name (CKN):** Configured name of the key in use.
- **Message ID:** A random 92 bit string used as an identifier for an MKA participant.
- **Elected Self:** True if this participant is the elected key server.
- **Success:** True if this participant is live and has at least one live peer.
- **Principal:** True if this participant is the principal participant elected to distribute SAKs.
- **Default:** True if this participant is a fallback/backup participant (spawned when a fallback key is configured in a Mac Security profile).
- **Key Server SCI:** The SCI of the key server.
- **SAK Transmit:** True if the participant is ready to use the negotiated key for transmit.
- **LLPN Exhaustion:** Increments if the number of data packets sent using the current key exceeds a certain threshold. Because we use a 64 bit packet number cipher suite, this should never increment.
- **Distributed Key Identifier:** Message ID + key number of the most recently generated SAK.

9.3.2.8.27 show mac security participants

The **`show mac security participants interface`** command displays information about the MACsec participants.

Command Mode

EXEC

Command Syntax

`show mac security interface`

Example

The following example displays information about MACsec participants.

```
switch# show mac security participants
Interface: Ethernet4/1/1
    CKN: abcd
      Message ID: 9d5bc0d3076ea4a08b99b9d9
      Elected self: True
      Success: True
      Principal: True
      Default: False

    CKN: dead
      Message ID: 4ef4cf009161bd551b5e7434
      Elected self: True
      Success: True
      Principal: False
      Default: True

Interface: Ethernet4/3/1
    CKN: abcd
      Message ID: c79ad8882c2dd3a8e838a691
      Elected self: False
      Success: True
      Principal: True
      Default: False
```

```
    CKN: dead
      Message ID: 3dfd4486b5f68a81014a37ec
      Elected self: False
      Success: True
      Principal: False
      Default: True
```

9.3.2.8.28 show mac security profile

The **show mac security profile** command displays information about the specified MACsec profile. If no profile is specified, information about all profiles is shown.

Command Mode

EXEC

Command Syntax

show mac security profile [***profile_name***]

Parameters

profile_name The MACsec profile to show information about.

Example

The following command shows information for the MACsec profile ***test***.

```
switch# show mac security profile
Profile: test
    Cipher: aes256-gcm-xpn
    Primary CKN:
    Primary CAK SHA-256 hash:
    Fallback CKN:
    Fallback CAK SHA-256 hash:
    Source: cli
    Priority: 100
    SCI Inclusion: disabled
    Key retirement policy: delayed
    Unprotected traffic policy: allow active-sak
    MKA lifetime: 6 seconds
    MKA key server priority: 16
    Session rekey period: 0
    Bypassed protocols:
    Max AN value of SAK: 3
    Configured on:
switch#
```

9.3.2.8.29 show mac security sak

The **show mac security sak** command displays information about MACsec static secure association key (SAK) status for the specified Ethernet interface. If no interface is specified, all interfaces are shown. The following information is shown for each Ethernet interface.

- The name of the Ethernet interface.
- The installed SAK IDs.
- The SAK profile name.
- The total number of SAKs generated.
- The number of SAKs generated due to a new live peer.
- The number of SAKs generated due to a rekey timer.
- The number of SAKs generated due to packet number exhaustion.

- The SAK installation time in seconds in each direction.
- The number of forced new Tx SAK installations.

Command Mode

EXEC

Command Syntax

show mac security sak [interface ethernet ***Ethernet_interface]***

Parameters

interface ethernet Show SAK status information about the specified Ethernet interface. If this option is omitted, information for all Ethernet interfaces is shown.

Ethernet_interface The Ethernet interface to show SAK status for.

Example

The following command displays the MACsec SAK status for the Ethernet interface Ethernet9/1.

```
switch(config-mac-security-profile-test)# show mac security sak
Interface: Ethernet9/1
Installed SAK ID: static SAK: Rx AN: 0,1 Tx AN: 0
Installed SAK from: static-SA
Total SAK generated: 0
SAK generated due to new live peer: 0
SAK generated due to rekey timer: 0
SAK generated due to packet number exhaustion: 0
SAK installation time( in seconds ):
Direction    0-1   1-2   2-3    3+
---------- ----- ----- ----- -----
Rx             1     0     0     0
Tx             1     0     0     0

Maximum Rx installation time: 0.0884998080001 seconds
Maximum Tx installation time: 0.0884941590002 seconds
Forced new Tx SAK installation count: 0
```

9.3.2.8.30 show mac security status

The **show mac security status** command displays the MACsec status information on a switch.

Command Mode

EXEC

Command Syntax

show mac security status

Example

The following command displays the MACsec status information.

```
switch# show mac security status
Active Profiles:                1
Data Delay Protection:         No
FIPS Mode:                     No
Secured Interfaces:             2
License:                       Enabled
```

9.3.2.8.31 supplicant profile

The **supplicant profile** command configures the supplicant profile containing all the credentials necessary for 802.1X authentication to succeed.

Command Mode

dot1x Configuration

Command Syntax

supplicant profile ***profile_name options***

Parameters

- ***profile_name*** Name of the supplicant profile.
- The following parameters can be included after entering the profile mode:
  - **eap-method** Extensible Authentication Protocol (EAP) method. Option include:
    - **fastEAP** Flexible Authentication via Secure Tunneling (FAST).
  - **identity** Extensible Authentication Protocol (EAP) user identity. Option include:
    - **WORD** User identity name.
  - **passphrase** Extensible Authentication Protocol (EAP) password. Options include:
    - ***0*** Specifies that an UNENCRYPTED key will follow.
    - ***7*** Specifies that an HIDDEN key will follow.
    - **LINE** The UNENCRYPTED (clear-text) shared key.

Examples

- The following commands place the switch in the supplicant profile mode.

```
Switch(config)# dot1x
Switch(config-dot1x)# supplicant profile test
Switch(config-dot1x-supp-profile-test)#
```

- The following commands configures the EAP FAST method for the supplicant profile called ***test profile*** for MAC security on the switch.

```
Switch(config)# dot1x
Switch(config-dot1x)# supplicant profile test
Switch(config-dot1x-supp-profile-test)#eap-method fast
```

- The following commands configures the Identity for the supplicant profile called test profile for MAC security on the switch.

```
Switch(config)# dot1x
Switch(config-dot1x)# supplicant profile test
Switch(config-dot1x-supp-profile-test)# identity New_User
```

- The following commands configures the passphrase for the supplicant profile called ***test profile*** for MAC security on the switch.

```
Switch(config)# dot1x
Switch(config-dot1x)# supplicant profile test
Switch(config-dot1x-supp-profile-test)# passphrase 7 070E334D5D1D0B04
```

9.3.2.8.32 traffic unprotected allow

The **traffic unprotected allow** command configures the switch to allow the unprotected traffic whenever there is no successful MKA session established with the peer.

The **no traffic unprotected allow** command disable the MACsec Fallback to Unprotected Traffic function by removing the traffic unprotected allow command from ***running-config***.

Command Mode

MACsec Profile

Command Syntax

traffic unprotected allow

no traffic unprotected allow

Example

The following commands configures the MACsec Fallback traffic unprotected allow feature on a switch for a MACsec profile called sample.

```
Switch(config)# mac security
Switch(config-mac-security)# profile sample
Switch(config-mac-security-profile-sample)# no traffic unprotected allow
```

9.3.3 Internet Protocol Security (IPsec)

This section describes Aristas IPsec implementation. Topics in this section include:

- IPsec Introduction
- IPsec Overview
- Configuring IPsec
- IPsec Commands

9.3.3.1 IPsec Introduction

Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents periodically during the session and negotiation of cryptographic keys to be used during the session. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection.

IPsec is used to protect data traffic between sites for example between Branch, HQ and Data center sites in an enterprise.

IPsec uses the following protocols to perform various functions:

- Authentication Headers (AH): provides the connectionless integrity and data origin authentication for IP datagrams and provides protection against replay attacks.
- Encapsulating Security Payloads (ESP): provides the confidentiality, data-origin authentication, connectionless integrity and an anti-replay service (a form of partial sequence integrity).
- Internet Key Exchange (IKE): is a key management protocol which provides security for virtual private networks' (VPNs) negotiations and network access to random hosts. It is also described as a method for exchanging keys for encryption and authentication over an unsecured medium, such as the Internet.

9.3.3.2 IPsec Overview

9.3.3.2.1 Security Associations

Security Associations (SA) provide the bundle of algorithms and data that provide the parameters necessary for AH and/or ESP operations. The Internet Security Association and Key Management Protocol (ISAKMP) provides a framework for authentication and key exchange, with actual authenticated keying material provided either by manual configuration with pre-shared keys, Internet Key Exchange (IKE and IKEv2) and other

mechanisms. In order to decide what protection is to be provided for an outgoing packet, IPsec uses the Security Parameter Index (SPI), an index to the security association database (SADB), along with the destination address in a packet header, which together uniquely identify a security association for that packet. A similar procedure is performed for an incoming packet, where IPsec gathers decryption and verification keys from the security association database.

Full bidirectional communication requires at least two SAs, one for each direction. SA is defined by the following parameters:

- Security Algorithms (AH) or Encapsulating Security Payloads (ESP) and keys.
- **Mode:** Tunnel or Transport.
- **Key Management Method:** Manual or IKE.
- **Lifetime:** Expressed in hours.

9.3.3.2.2 Mode of Operation

IPsec on Arista switches operates in tunnel mode. In tunnel mode, the entire IP packet is encrypted and/or authenticated. It is then encapsulated into a new IP packet with a new IP header.

Tunnel mode is used to create virtual private networks for network-to-network communications (for example, between routers to link sites). Tunnel mode is used for most network-to-network IPsec.

9.3.3.2.3 Key Management

Key management on Arista switches uses the Internet Key Exchange (IKE) method. Internet Key Exchange (IKE) supports automated generation and renegotiation of SAs (includes keys) between the devices at a configured interval so it is much more scalable and secure.

IPsec needs SAs to define the algorithms and keys to use for protecting traffic. IKE establishes the SA so IPsec can protect traffic.

There are two IKE versions, IKEv1 and IKEv2. IKEv2 builds on IKEv1 but both are still widely used today.

9.3.3.2.3.1 IKEv1

IKEv1 has two phases.

- IKEv1 Phase 1
- IKEv1 Phase 2

IKEv1 Phase 1

- Uses main or aggressive mode exchange
- Negotiates IKE SA
- Used for control plane
- Peer authentication

IKEv1 Phase 2

- Uses quick mode exchange
- Negotiates IPsec SAs

Note that there are two different SAs that are established. The IKE SA protects only the IKE key management session using the IKE policy defined. The policy should include the following parameters:

- - Encryption algorithm
  - Hash MAC (HMAC) algorithm
  - Peer authentication procedure
  - Diffie-Hellman group for initial key exchange
  - SA lifetime

IKE initially performs a Diffie-Hellman (DH) exchange at the start of the IKE session. A Diffie-Hellman (DH) exchange allows participants to produce a shared secret value. The strength of the technique is that it allows participants to create the secret value over an unsecured medium without passing the secret value through the wire. From that exchange, peers get shared keying material, which is then used for IKE encryption and integrity functions. The strength of that keying material can be used for faster performance, by choosing lower key sizes for Diffie-Hellman exchanges. The key length (strength) of Diffie-Hellman exchanges can be changed with the use of different DH groups.

When an IKE session's lifetime expires, a new Diffie-Hellman exchange is performed between peers and the IKE SA is re-established.

The IPsec protection policy resulting in IPsec SAs, defines the protection of network traffic. These IPsec SAs are usually negotiated over IKE sessions. The parameters that define the IPsec protection policy are:

- Encryption Algorithm
- Hash MAC (HMAC) Algorithm

Note that the key material for IPsec SA (also called Child SA) is derived from keying material from IKEv1 phase 1.

There are two different modes for phase 1:

- Main Mode
- 6 packet exchange
- Full identity protection and better anti-DoS protection
- Aggressive Mode
- 3 packet faster session establishment
- Identities are exchanged in clear
- Weak DoS protection

Authentication

- **Pre-Shared Keys (PSK):** As the name suggests, a shared secret is distributed out-of-band to the peers. The peers use this information and nonce parameters to create a hash that is used to authenticate messages.
- **PKI Certificates:** Here, certificates of the peers are exchanged and hashes are calculated over these certificates to authenticate each other.

9.3.3.2.3.2 IKEv2

IKEv2 differs from IKEv1 in the following ways:

- Faster setup because of reduced number of messages.
- More secure.
- ESP is reused for all IKEv2 messages.
- Suite-B support.
- There is no aggressive mode, so IKEv2 always provides identity protection.
- Additional authentication methods.
- Local and remote can use different authentication methods and use different pre-shared keys.
- Authentication is done unidirectionally in IKEv2.

9.3.3.2.4 Route-based VPN

A route-based VPN employs routed tunnel interfaces as the endpoints of the virtual network. All traffic passing through a tunnel interface is placed into the VPN. Rather than relying on an explicit policy to dictate which traffic enters the VPN, static and/or dynamic IP routes are formed to direct the desired traffic through the VPN tunnel interface.

Since route-based VPNs support dynamic routing information through VPN tunnels. EOS supports only route based VPN for dynamic routing support and for easier configuration and management.

In route-based VPN, features like NAT, ACL, QoS is applied to packets before they are encrypted by applying these features to tunnel interface and can be applied to encrypted packets to applying these features on the physical interface carrying the tunnel traffic.

Virtual Template Interface (VTI)

A new tunnel interface type vti is introduced to represent the VPN tunnel. This tunnel interface will participate in the routing and any packets forwarded to it will be encrypted and forwarded to the other end of the tunnel. Note, that this does not add a new header to the packet.

9.3.3.3 Configuring IPsec

Complete the following steps to configure IPsec tunnels over the switch.

This configuration will use the default IKE version 2 procedure.

1. Use **`ip security`** command to enter IP security mode.

```
switch(config)# ip security
```

2. To use IKE version 1, complete the following before completing the default IKE version the steps below.

```
switch(config)# ip security
switch(config-ipsec)# ike policy ike-peerRtr
switch(config-ipsec-ike)# version 1
```

3. Create an IKE Policy to be used to communicate with the peer to establish IKE. You have the option of configuring multiple IKE policies.

   The default IKE Policy values are:

   - **Encryption:** AES256 / AES128
   - **Integrity:** SHA256 / SHA128
   - **DH group:** Group 14
   - **IKE lifetime:** 8 hours

```
switch(config-ipsec)# ike policy ike-router
switch(config-ipsec-ike)# encryption aes256
switch(config-ipsec-ike)# integrity sha256
switch(config-ipsec-ike)# dh-group 24
switch(config-ipsec-ike)# version 2
```

4. If the router is behind a NAT, configure the local-id with the local public IP address. The public IP corresponds to the underlying interface over which the IKE communications are done with the peer.

```
switch(config-ipsec-ike)# local-id <public ip address>
```

5. Create an IPsec Security Association policy to be used in the data path for encryption and integrity. Use the option of enabling Perfect Forward Secrecy by configuring a DH group to the SA. In this example, ***AES256*** is used for encryption, ***SHA 256*** is used for integrity, and Perfect Forward Secrecy is enabled (the DH group is 14).

```
switch(config-ipsec)# sa policy sa-vrouter
switch(config-ipsec-sa)# esp encryption aes256
switch(config-ipsec-sa)# esp integrity sha256
switch(config-ipsec-sa)# pfs dh-group 14
switch(config-ipsec-sa)# sa lifetime 2
switch(config-ipsec-sa)# exit
```

6. Bind or associate the IKE and SA policies together using an IPsec profile. Provide a shared-key, which must be common on both peers. The default profile assigns default values for all parameters that are not explicitly configured in the other profiles. In this example, the IKE Policy ***ike-peerRtr*** and SA Policy ***sa-peerRtr*** are applied to profile ***peer-Rtr***. Dead Peer Detection is enabled and configured to delete the connection when the peer is down for more than ***50*** seconds. The peer ***peer-Rtr*** is set to be the responder.

```
switch(config-ipsec)# profile default
switch(config-ipsec-profile)# ike-policy ikedefault
switch(config-ipsec-profile)# sa-policy sadefault
switch(config-ipsec-profile)# shared-key arista
switch(config-ipsec-profile)# connection start
switch(config-ipsec)# profile vrouter
switch(config-ipsec-profile)# ike-policy ike-vrouter
switch(config-ipsec-profile)# sa-policy sa-vrouter
switch(config-ipsec-profile)# dpd 10 50 clear
switch(config-ipsec-profile)# connection add
```

7. Configure the WAN interface to be the underlying interface for the tunnel. You must specify an L3 address for the tunnel. If you do not, the switch cannot route packets using the tunnel.

```
switch(config)# interface Et1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# ip address 1.0.0.1/24
switch(config-if-Et1)# mtu 1500
```

8. Apply the IPsec profile to a new tunnel interface. You create the new tunnel interface as part of this step. You can configure the tunnel as a VTI IPsec tunnel. In this example, the new tunnel interface is ***Tunnel0***. The new tunnel interface is configured to use IPsec. The other end of the tunnel also needs to be configured as a GRE-over-IPsec tunnel.

```
switch(config)# interface tunnel0
switch(config-if-Tu0)# ip address 1.0.3.1/24
switch(config-if-Tu0)# mtu 1394
switch(config-if-Tu0)# tunnel source 1.0.0.1
switch(config-if-Tu0)# tunnel destination 1.0.0.2
switch(config-if-Tu0)# tunnel ipsec profile vrouter
```

Example Configuration

```
ip security
ike policy ikebranch1
integrity sha256
dh-group 15
!
sa policy sabranch1
sa lifetime 2
pfs dh-group 14
!
profile hq
mode tunnel
ike-policy ikebranch1
sa-policy sabranch1
connection add
shared-key keyAristaHq
dpd 10 50 clear
!
interface Tunnel1
mtu 1404
ip address 1.0.3.1/24
tunnel source 1.0.0.1
```

```
tunnel destination 1.0.0.2
tunnel ipsec profile hq
!
interface Ethernet1
no switchport
ip address 1.0.0.1/24
!
```

9.3.3.3.1 Displaying IPsec Information

- Use the **show ip security policy** command to display the IPsec policy information.

```
switch# show ip security policy
Policy Name   Authentication   Encryption   Integrity    Lifetime  Rekey  DH Group
ike-policy    Pre-shared       256-bit AES  256bit Hash  8 hours   False  3072 bit
```

- Use the **show ip security profile** command to display the IP security profile information.

```
switch# show ip security profile
Profile name       IKE Policy Name       SA
ipsec-profile      ike-policy            sa-policy
```

9.3.3.4 IPsec Commands

- ike policy
- interface tunnel (IPsec)
- ip security
- profile (IPsec)
- sa policy
- show ip security applied-profile
- show ip security connection
- show ip security policy
- show ip security profile
- show ip security security-association

9.3.3.4.1 ike policy

The **ike policy** command configures the Internet Security Association and Key Mgmt Protocol on the switch and related policies. The IKE policy is configured in IP security configuration mode.

The **no ike policy** command deletes the IKE policy configuration from the switch.

The **exit** command returns the switch to the global configuration mode.

Command Mode

IP Security Configuration

Command Syntax

ike policy ***policy-name***

no ike policy ***policy-name***

Parameters

- ***policy-name*** Specifies the IKE policy name.

The following parameters are allowed to configure when the switch is placed in IKE policy configuration mode:

- **authentication** specifies the authentication type.
- **dh-group** specifies Diffie-Hellman Group value.

- **encryption** specifies the encryption type.
- **ike-lifetime** sets the ikeLifetime for ISAKMP security association. Expressed in hours.
- **integrity** specifies the Integrity algorithm.
- **local-id** specifies the local IKE identification.
- **remote-id** remote peer IKE identification.
- **version** specifies the IKE version.

Example

This command configures the IKE policy test for IP security configuration.

```
switch(config)# ike policy test
switch(config-ipsec-ike)#
```

9.3.3.4.2 interface tunnel (IPsec)

The **`interface tunnel`** command places the switch in the interface tunnel configuration mode.

Interface tunnel configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed.

The **`no interface tunnel`** command deletes the interface tunnel configuration.

The **`exit`** command returns the switch to the global configuration mode.

Command Mode

Global Configuration

Command Syntax

`interface tunnel` ***value***

`no interface tunnel` ***value***

Parameter

value Tunnel interface number. The value ranges from ***0*** to ***255***.

Example

This command places the switch in interface tunnel configuration mode with a tunnel value ***10***.

```
switch(config)# interface tunnel 10
switch(config-if-Tu10)#
```

9.3.3.4.3 ip security

The **`ip security`** command places the switch in the IP security configuration mode.

IP security configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed.

The **`no ip security`** command deletes the IP security configuration.

The **`exit`** command returns the switch to the global configuration mode.

Command Mode

Global Configuration

Command Syntax

`ip security`

`no ip security`

Example

This command places the switch in IP security configuration mode.

```
switch(config)# ip security
switch(config-ipsec)# ike policy IKE1
switch(config-ipsec-IKE1)# exit
switch(config-ipsec)# sa policy SA1
switch(config-SA1)#
```

9.3.3.4.4 profile (IPsec)

The **profile** command configures the IP security profile on the switch. The profile is configured in IP security configuration mode.

The **no profile** command deletes the IP security profile configuration from the switch.

The **exit** command returns the switch to the global configuration mode.

Command Mode

IP Security Configuration

Command Syntax

profile ***profile-name***

no profile ***profile-name***

Parameter

- ***profile-name*** Specifies the IP security profile name.

The following parameters can be configured in SA policy configuration mode:

- **connection** IPsec Connection (Initiator/Responder/Dynamic).
- **dpd** Dead Peer Detection.
- **flow** sets the flow.
- **ike-policy** ISAKMP policy.
- **mode** IP security mode type.
- **sa-policy** security association name.
- **shared-key** specifies key value.

Example

This command configures the IP security profile test for IP security configuration.

```
switch(config)# profile test
switch(config-ipsec-profile)#
```

9.3.3.4.5 sa policy

The **sa policy** command specifies a Security Association (SA) policy to be used for IPsec configuration, and enters IP security SA policy configuration mode to configure the named policy.

The **no sa policy** command deletes the specified SA policy configuration from the switch.

The **exit** command returns the switch to the global configuration mode.

Note: Arista EOS 4.22.0F release supports two combinations of encapsulations only: "esp encryption aes128" with "esp integrity sha1" and "esp encryption aes256" with "esp integrity sha256".

Command Mode

IP Security Configuration

Command Syntax

sa policy ***policy_name***

no sa policy ***policy_name***

Parameter

- ***policy_name*** Specifies the SA policy name.

The following parameters are configured in IP security SA policy configuration mode:

- **anti-replay** IPsec duplicate IP datagram detection.
- **esp** Encapsulation Security Payload.
- **pfs** Perfect Forward Secrecy.
- **sa** Security Association.

Example

This command applies the SA policy called ***test*** for IP security and enters IP security SA policy configuration mode for the test policy.

```
switch(config)# sa policy test
switch(config-ipsec-sa)#
```

9.3.3.4.6 show ip security applied-profile

The **show ip security applied-profile** command displays the IP security profile names and the interfaces on which they are applied.

Command Mode

EXEC

Command Syntax

show ip security applied-profile

Example

This command displays the IP security ***profile-1*** and the interfaces on which it is applied.

```
switch# show ip sec applied-profile
Profile Name                 Interface
ipsec-profile-1              Tunnel1,
                             Tunnel2,
                             Tunnel3,
                             Tunnel4,
                             Tunnel5,
                             Tunnel6,
                             Tunnel7,
                             Tunnel8,
                             Tunnel9,
                             Tunnel10,
                             Tunnel11,
                             Tunnel12,
                             Tunnel13,
                             Tunnel14,
                             Tunnel15,
                             Tunnel16,
                             Tunnel17,
                             Tunnel18,
                             Tunnel19,
```

```
                                Tunnel20,
                                Tunnel21,
                                Tunnel22,
                                Tunnel23,
                                Tunnel24,
                                Tunnel25,
                                Tunnel26,
```

9.3.3.4.7 show ip security connection

The **`show ip security connection`** command displays the IP security connection status information.

Command Mode

EXEC

Command Syntax

`show ip security connection`

Example

These commands display the IP security connection status information.

```
switch# show ip sec conn tunnel 1
Tunnel   Source     Dest     Status       Uptime     Input    Output        Rekey Time
Tunnel1  11.1.1.1   11.2.1.1 Established  19 hours   0 bytes  0 bytes       4 hours
                                                    0 pkts   62937679 pkts

switch# show ip sec conn tunnel 1 detail
Tunnel1:
   source address 11.1.1.1, dest address 11.2.1.1
   state: Established
   uptime: 19 hours, 7 minutes, 23 seconds
   Inbound SPI 0xca5560f4:
      request id 193, mode tunnel replay-window 16384, seq 0x0
      stats errors:
         replay-window 0, replay 0, integrity_failed 0
      lifetime config:
         softlimit 4534352933249 bytes, hardlimit 6442450944000 bytes
         softlimit 2077499095 pkts, hardlimit 4000000000 pkts
         expire add soft 85619 secs, hard 86400 secs
      lifetime current:
         0 bytes, 0 pkts
         add time Mon May 13 17:33:54 2019, use time Mon May 13 17:33:54 2019
   Outbound SPI 0xc60da749:
      request id 193, mode tunnel replay-window 16384, seq 0x0
      stats errors:
         replay-window 0, replay 0, integrity_failed 0
      lifetime config:
         softlimit 3286021368749 bytes, hardlimit 6442450944000 bytes
         softlimit 2480571031 pkts, hardlimit 4000000000 pkts
         expire add soft 85418 secs, hard 86400 secs
      lifetime current:
         0 bytes, 62937679 pkts
         add time Mon May 13 17:33:54 2019, use time Mon May 13 18:06:42 2019
```

9.3.3.4.8 show ip security policy

The **`show ip security policy`** command displays the IP security policy information.

Command Mode

EXEC

Command Syntax

`show ip security policy`

Example

This command displays IP security policy configuration information.

```
switch# show ip security policy
Policy Name   Authentication  Encryption   Integrity    Lifetime  Rekey  DH Group

ike-policy    Pre-shared      256-bit AES  256bit Hash  8 hours   False  3072 bit
```

9.3.3.4.9 show ip security profile

The **show ip security profile** command displays the IP security profile information.

Command Mode

EXEC

Command Syntax

show ip security profile

Example

This command displays IP security profile configuration information.

```
switch# show ip security profile
Profile name            IKE Policy Name        SA
ipsec-profile           ike-policy             sa-policy
```

9.3.3.4.10 show ip security security-association

The **show ip security security-association** command displays the IP security SA information.

Command Mode

EXEC

Command Syntax

show ip security security-association

Example

This command displays the IP security SA information.

```
switch# show ip sec security-association
SA Name      ESP Encryption   ESP Integrity    Lifetime  PFS Group

sa-policy-1  256-bit AES      256bit Hash      24 hours  2k bit
```

9.3.4 Macro-Segmentation Service (CVX)

Arista MSS is designed as a service in CloudVision that provides the point of integration between individual vendor firewalls or a firewall manager and the Arista network fabric. MSS provides flexibility on where to place the service devices and workloads. It is specifically aimed at Physical-to-Physical (P-to-P) and Physical-to-Virtual (P-to-V) workloads.

Topics in this section include:

- Overview
- How MSS Works
- Configuration
- MSS Integration with Check Point
- MSS for Layer 3 Firewall Enhancements
- MSS Commands

9.3.4.1 Overview

The advent of contemporary networking features such as mobile applications and the Internet of Things (IoT) bring in additional security challenges that are unprotected by legacy infrastructure. These security breaches cannot be handled by installing a firewall at the Internet edge. Arista Macro-Segmentation Service (MSS) addresses the security breach issue, besides securing access, protecting critical data and end-user privacy.

Arista MSS is designed as a service in CloudVision that provides the point of integration between a vendor firewall or a firewall manager and the Arista network fabric. MSS provides flexibility on where to place the service devices and workloads. It is specifically aimed at Physical-to-Physical (P-to-P) and Physical-to-Virtual (P-to-V) workloads.

MSS components include:

- Arista leaf-spine switch fabric
- Arista CloudVision
- Vendor firewall attached to a spine or service leaf switches. Different vendor firewalls can be attached to different switches to enhance scalability.

The above component topology allows for consistency in application deployment, scale, manageability, and easier scalability of the network and service layers.

- Benefits
- Terminology
- Usage Scenarios

9.3.4.1.1 Benefits

MSS provides the following key benefits:

- Enhanced security between any physical and virtual workloads in the data center.
- The automatic and seamless service insertion ability of MSS eliminates manual steering of traffic for a workload or a tenant.
- Security policies are applied to the host and application throughout the network.
- MSS is flexible since there are no proprietary frame formats, tagging, or encapsulation.

9.3.4.1.2 Terminology

The following terms related to MSS are used to describe the MSS feature:

- **Intercept Switch/VTEP**: TOR switch and VXLAN tunnel end-point connected to host from which traffic is intercepted. In the topology diagram, Intercept-1 and Intercept-2 are intercept switches.
- **Service Switch/VTEP**: TOR switch and VXLAN tunnel end-point connected to a firewall. In the topology diagram, Service-1 is the service switch.
- **VXLAN**: Virtual eXtensible LAN - a standards-based method of encapsulating Layer 2 traffic across a Layer 3 fabric.
- **CVX**: Arista CloudVision eXchange (CVX) is a part of CloudVision and is a virtualized instance of the same Extensible Operating System (EOS) that runs on physical switches. It functions as a point of integration between customer firewalls or firewall policy managers and the Arista network in order to steer traffic to the firewall.

9.3.4.1.3 Usage Scenarios

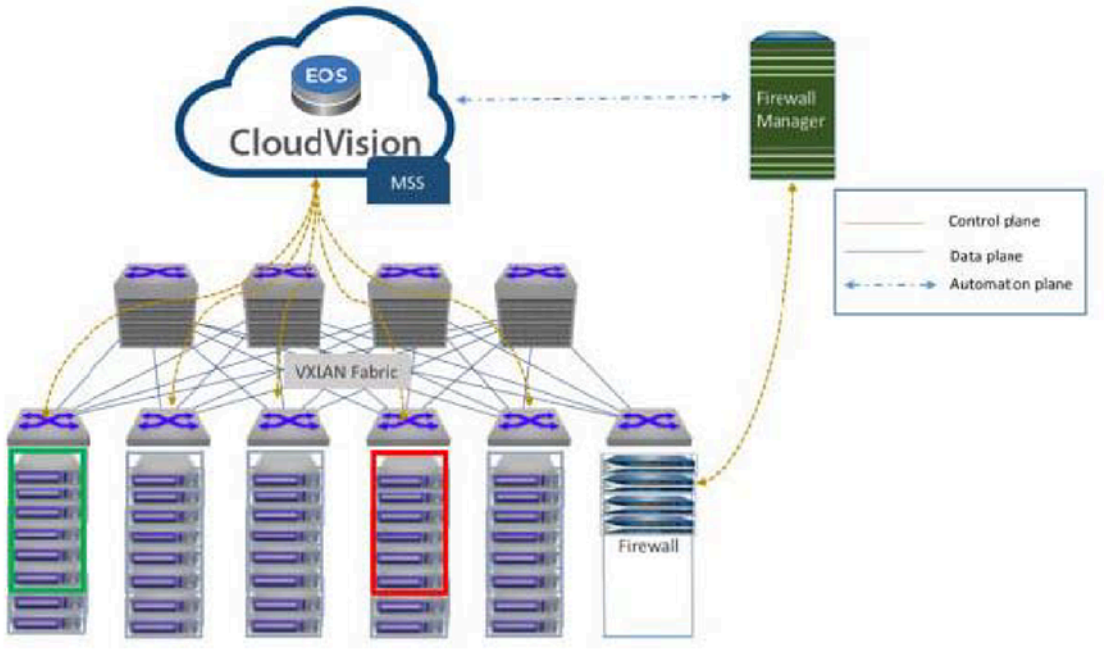
The following usage scenarios describe a few major security challenges in today’s data center that are successfully handled by MSS.

1. Securing server-to-server traffic.

This scenario provides information about the role of MSS in securing network traffic between physical-to-physical (P-to-P) and physical to virtual (P-to-V) servers. Prior to MSS, network infrastructure devices followed the firewall sandwich setup where firewalls were placed in line between the security zones. This setup would impact scalability and performance of the servers.

Using MSS, this restriction on firewall placement is reduced. Firewalls are now attached to a service leaf switch in the network fabric and they still protect hosts without concern about their physical location. The following topology demonstrates the usage scenario.

Figure 9-11: Securing server-server traffic



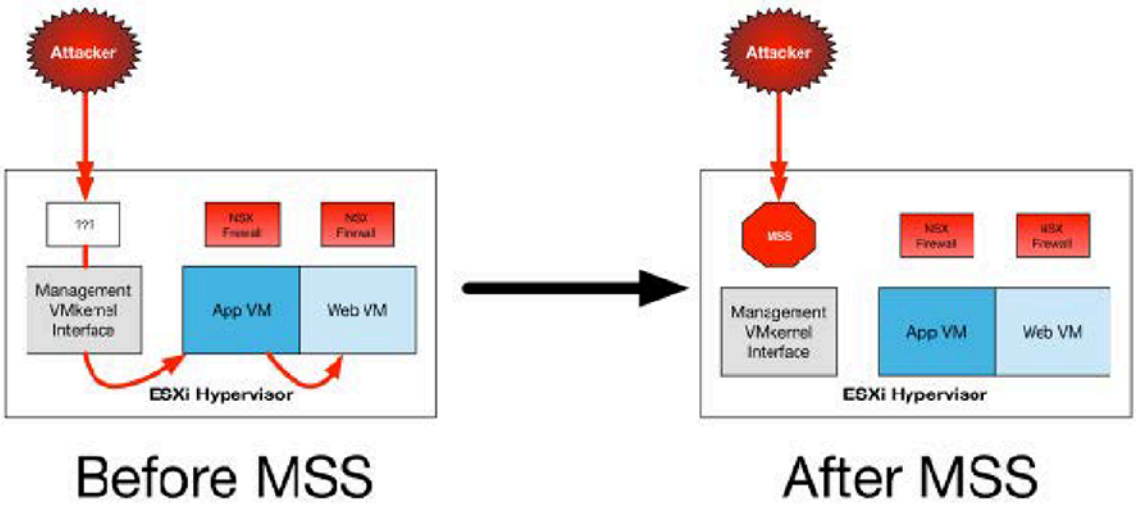
2. Monitoring and securing management traffic.

   This usage scenario demonstrates how MSS successfully monitors and secures management interfaces in the data center.

   The modern data center caters to managing the application, storage, virtualization, network, analytics and other layers. With virtualization, the hypervisor management also needs to be secured to prevent unwanted access to a hypervisor management interface. In the event of a rogue access, Aristas MSS protects management interfaces. The explicitly allowed hosts can gain access through a jump host or

administrator end-user computing instances. The following topology diagram illustrates the role of MSS in a data center.

Figure 9-12: Monitoring and Securing management traffic

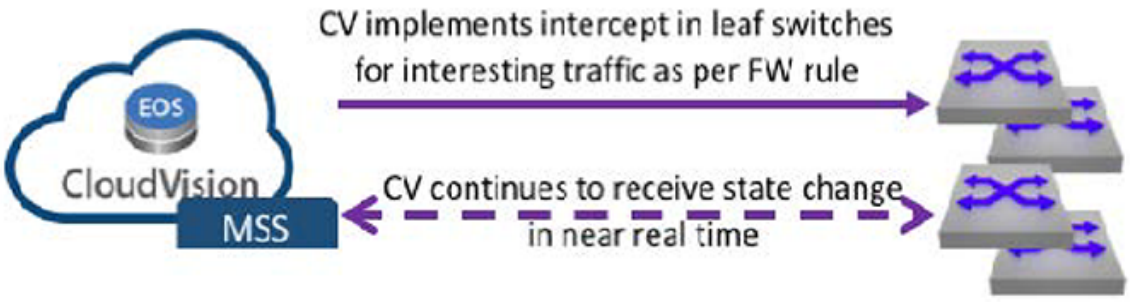


9.3.4.2 How MSS Works

The following steps provide information about how MSS works as a service in the data center.

1. MSS is enabled on the CloudVision eXchange (CVX) and the Arista switches are configured to stream their active state to CVX. This allows CVX to build a database of hosts and firewalls attached to the network and also to identify physical ports and IP addresses. CVX is also configured to communicate and synchronize policies from a vendor's firewall.
2. CVX sends a request to the firewall or firewall manager to provide information about the security policies which are tagged for MSS usage.
3. The MSS service on CVX determines the flow based forwarding rules to be pushed to the switches in the network.

Figure 9-13: CVX intercept



4. The leaf switch starts sending intercepted traffic to the service leaf when the intercept has been applied to the leaf switch.

Figure 9-14: Leaf switches intercept



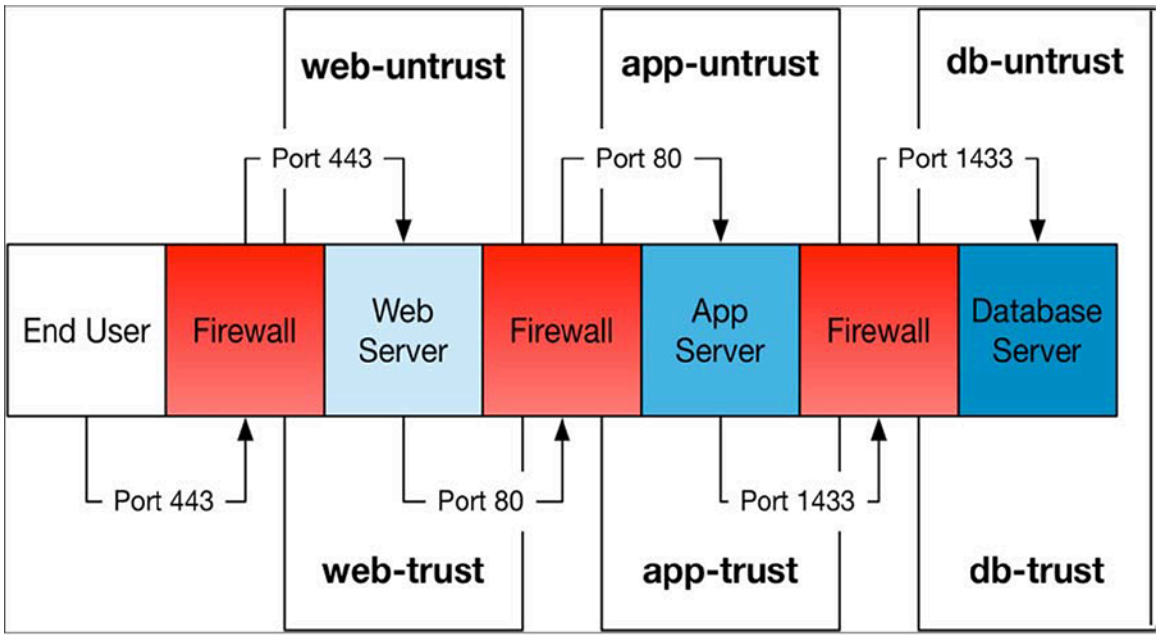
5. Traffic is forwarded completely unmodified to the firewall after it enters the service leaf where the firewall is attached. Based on the configuration policy, the firewall applies the required actions such as inspection, log, allow, or deny.
6. The service leaf switch sends the inspected traffic to its final destination or to the destination based on the firewall policy.

9.3.4.3 Configuration

The following sections provide detailed information about MSS configuration, system requirements, recommendations, and limitations.

The traffic flow below is an example of a typical MSS deployment with a 3-tiered application. The goal of this design is to limit access between hosts in the following zones: web-untrust, app-untrust, db-untrust, web-trust, app-trust, and db-trust.

Figure 9-15: Traffic flow in an MSS deployment

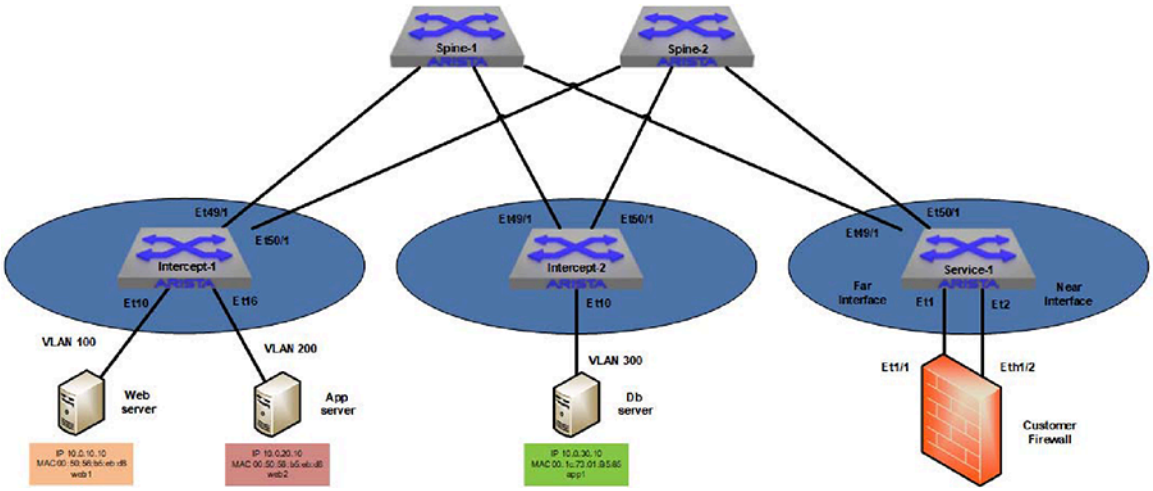


End users in the untrust zone access the web server through the TCP/443 port. Traffic flows through the active firewall to the web server interface in the web-untrust security zone. The web server interface in the web-trust security zone accesses the application server interface in the app-untrust security zone through port TCP/80 after traversing the firewall. From there, the application server interface in the app-trust security zone accesses the database through TCP/1433 in the db-untrust zone.

The following physical topology indicates the MSS setup.

The hosts are attached to a pair of intercept leaf switches. A firewall is connected to a service leaf switch using a pair of physical interfaces with a subinterface per zone or vWire.

Figure 9-16: Physical Topology of the MSS



- System Requirements
- Recommendations and Limitations
- Configuring MSS

9.3.4.3.1 System Requirements

The system requirements to effectively run MSS are listed below.

- Arista CloudVision eXchange (CVX).
- Arista 7280SR, 7280TR, 7280CR, 7020SR, 7020TR series switches; 7050X, 7050X2, 7060X, and 7060X2 series top of rack (TOR) switches.
- Connected to the hosts to intercept traffic from the firewall devices.
- The network must be a VXLAN-enabled fabric with CVX running the VXLAN Control Service (VCS) or EVPN.

9.3.4.3.2 Recommendations and Limitations

Firewall

The firewall policy name must not have any whitespace character in the name. As an example, ***PCI policy*** is an unacceptable policy name. An acceptable name would be ***PCI_policy***.

9.3.4.3.3 Configuring MSS

These sections describe steps to configure MSS.

- Deploying CVX
- Enabling the VXLAN Control Service on CVX
- Configuring the Access Switches and the Service Switch Ports
- Enabling DirectFlow on Access Switches and Service Switches
- Enabling VXLAN routing on the TOR switches
- Configuring MSS on CVX
- Configuring the Firewall

9.3.4.3.3.1 Deploying CVX

Deploy CloudVision and configure the Arista TOR switches to connect to it. A CVX cluster of three instances with host names of ***cvx01***, ***cvx02***, and ***cvx03*** are configured as an example.

Note: As a best practice, always deploy the CV in a HA cluster with a minimum of three instances.

9.3.4.3.3.2 Enabling the VXLAN Control Service on CVX

Enable the VXLAN Control Service (VCS) on every CVX instance after the three Arista CVX instances have been deployed and the TOR switches are configured to be managed by them.

VCS allows hardware VXLAN Tunnel End Points (VTEPs) to share state with each other in order to establish VXLAN tunnels without the need for a multicast control plane.

Example

CVX instance cvx01

```
cvx01(config-cvx)# service vxlan
cvx01(config-cvx-vxlan)# no shutdown
```

Similarly, VCS is enabled on the ***cvx02*** and ***cvx03*** devices.

9.3.4.3.3.3 Configuring the Access Switches and the Service Switch Ports

Configure the switch ports that are connected to the hosts, whose traffic should be steered to the firewalls and the service switch ports which are connected to the firewalls.

Access Switch Configuration

The switch ports connected to the hosts, whose traffic needs to be intercepted, need to be configured as 802.1q trunks with the VLAN that is mapped to the VNI requiring interception. Unique VLAN IDs are configured for each tier of the application.

Access Switch (Intercept-1)

```
intercept-1# configure
intercept-1(config)# interface et10
intercept-1(config-if-Et10)# description web server
intercept-1(config-if-Et10)# switchport mode trunk
intercept-1(config-if-Et10)# switchport trunk allowed vlan 100

intercept-1(config)# interface et16
intercept-1(config-if-Et16)# description app server
intercept-1(config-if-Et16)# switchport mode trunk
intercept-1(config-if-Et16)# switchport trunk allowed vlan 200
```

Access Switch (Intercept-2)

```
intercept-2# configure
intercept-2(config)# interface et10
intercept-2(config-if-Et1)# description db server
intercept-2(config-if-Et1)# switchport mode trunk
intercept-2(config-if-Et1)# switchport trunk allowed vlan 300
```

Note: For untagged traffic, configure a native VLAN on the port using the **switchport trunk native vlan** command.

Service Switch (Service-1)

```
service-1# configure
service-1(config)# interface port-channel 10
service-1(config-if-Po10)# description Far Interface
service-1(config-if-Po10)# switchport mode trunk
service-1(config-if-Po10)# switchport trunk allowed vlan none
service-1(config-if-Po10)# spanning-tree bpdufilter enable

service-1(config)# interface port-channel 20
service-1(config-if-Po20)# description Near Interface
service-1(config-if-Po20)# switchport mode trunk
service-1(config-if-Po20)# switchport trunk allowed vlan none
service-1(config-if-Po20)# spanning-tree bpdufilter enable
```

Note: Dynamically mapped VLANs are not shown in the switch port configuration. You can view them by running the **`show vlan`** command on the switch once a policy is applied.

9.3.4.3.3.4 Enabling DirectFlow on Access Switches and Service Switches

Arista MSS uses DirectFlow to intercept traffic while the VxLAN is used to carry tunnel traffic from the intercepted host to the firewall and back. DirectFlow should be enabled on every intercept switch as well as the service switches.

Switch Service-1

```
service-1# configure
service-1(config)# directflow
service-1(config-directflow)# no shutdown
```

Switch Intercept-1

```
intercept-1# configure
intercept-1(config)# directflow
intercept-1(config-directflow)# no shutdown
```

Switch Intercept-2

```
intercept-2# configure
intercept-2(config)# directflow
intercept-2(config-directflow)# no shutdown
```

9.3.4.3.3.5 Enabling VXLAN routing on the TOR switches

CVX uses Address Resolution Protocol (ARP) to determine where intercept hosts are physically located in the network. VXLAN routing should be configured on every TOR switch that will be intercepting traffic to ensure that CVX is aware of every host ARP entry.

The following configuration shows the routing configuration for each tier of the application, but not the entire VXLAN configuration. For more information on how to configure VXLAN and VXLAN routing, refer to the VXLAN section of the ***Arista EOS Configuration Guide***.

Switch Intercept-1

```
intercept-1# configure
intercept-1(config)# ip routing
intercept-1(cofig)# interface vlan100
intercept-1(config-if-Vl100)# ip address virtual 10.0.10.254/24
intercept-1(config)# interface vlan200
intercept-1(config-if-Vl200)# ip address virtual 10.0.20.254/24
```

```
intercept-1(config)# interface vlan300
intercept-1(config-if-Vl300)# ip address virtual 10.0.30.254/24
```

Switch Intercept-2

```
intercept-2# configure
intercept-2(config)# ip routing
intercept-2(cofig)# interface vlan100
intercept-2(config-if-Vl100)# ip address virtual 10.0.10.254/24
intercept-2(config)# interface vlan200
intercept-2(config-if-Vl200)# ip address virtual 10.0.20.254/24
intercept-2(config)# interface vlan300
intercept-2(config-if-Vl300)# ip address virtual 10.0.30.254/24
```

Switch Service-1

```
service-1# configure
service-1(config)# ip routing
service-1(cofig)# interface vlan100
service-1(config-if-Vl100)# ip address virtual 10.0.10.254/24
service-1(config)# interface vlan200
service-1(config-if-Vl200)# ip address virtual 10.0.20.254/24
service-1(config)# interface vlan300
service-1(config-if-Vl300)# ip address virtual 10.0.30.254/24
```

9.3.4.3.3.6 Configuring MSS on CVX

This step enables configuring Arista MSS on CVX. The topology diagram depicts three CVX instances in a cluster and the configuration is the same for every instance. The active and standby vendor firewalls are configured. If Panorama is used, only Panorama should be configured.

Example

In the example, the primary vendor firewall has a DNS name of ***fw-ha-node-1***. The standby firewall has a DNS name of ***fw-ha-node-2***. The username and password are set as ***admin***.

CVX instance *cvx01*

```
cvx01# configure
cvx01(config)# cvx
cvx01(config-cvx)# no shutdown
cvx01(config-cvx)# service mss
cvx01(config-cvx-mss)# no shutdown
cvx01(config-cvx-mss)# vni range 20000-30000
cvx01(config-cvx-mss)# dynamic device-set panfw1
cvx01(config-cvx-mss-panfw1)# tag Arista_MSS
cvx01(config-cvx-mss-panfw1)# type palo-alto firewall
cvx01(config-cvx-mss-panfw1)# state active
cvx01(config-cvx-mss-panfw1)# device fw-ha-node-1
cvx01(config-cvx-mss-panfw1-fw-ha-node-1)# username admin password 0 admin
```

CVX instance *cvx02*

```
cvx02# configure
cvx02(config)# cvx
cvx02(config-cvx)# no shutdown
cvx02(config-cvx)# service mss
cvx02(config-cvx-mss)# no shutdown
cvx02(config-cvx-mss)# vni range 20000-30000
cvx02(config-cvx-mss)# dynamic device-set panfw1
cvx02(config-cvx-mss-panfw1)# tag Arista_MSS
cvx02(config-cvx-mss-panfw1)# type palo-alto firewall
```

```
cvx02(config-cvx-mss-panfw1)# state active
cvx02(config-cvx-mss-panfw1)# device fw-ha-node-1
cvx02(config-cvx-mss-panfw1-fw-ha-node-1)# username admin password 0 admin
```

CVX instance *cvx03*

```
cvx03# configure
cvx03(config)# cvx
cvx03(config-cvx)# no shutdown
cvx03(config-cvx)# service mss
cvx03(config-cvx-mss)# no shutdown
cvx03(config-cvx-mss)# vni range 20000-30000
cvx03(config-cvx-mss)# dynamic device-set panfw1
cvx03(config-cvx-mss-panfw1)# tag Arista_MSS
cvx03(config-cvx-mss-panfw1)# type palo-alto firewall
cvx03(config-cvx-mss-panfw1)# state active
cvx03(config-cvx-mss-panfw1)# device fw-ha-node-1
cvx03(config-cvx-mss-panfw1-fw-ha-node-1)# username admin password 0 admin
```

9.3.4.3.3.7 Configuring the Firewall

Three policies are created in addition to the default implicit deny policy for inter-zone traffic. The implicit deny ensures that the inter-zone traffic is not allowed unless a policy explicitly allows for it.

The first policy ***untrust_to_web1*** is from the ***untrust*** zone to the ***web1*** zone, that allows HTTPS traffic from anywhere to the web server web.

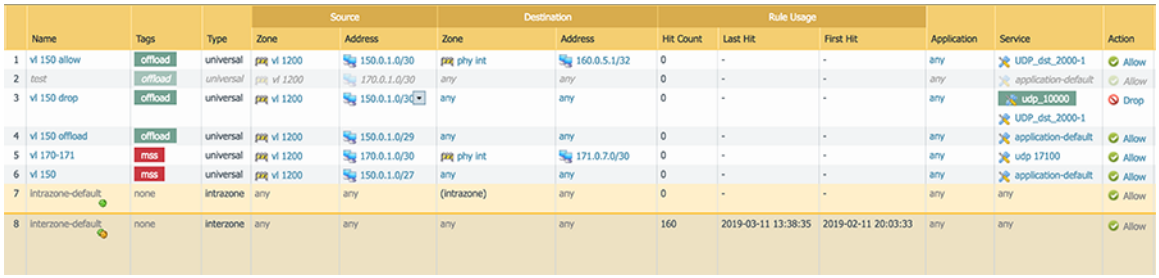
The third policy ***web2_to_app1*** is from the ***web2*** zone to the app1 zone that allows HTTP traffic between the web server web and the application server app.

The fifth policy ***app2_to_db1*** is from the ***app2*** zone to the ***db1*** zone that allows database traffic on port ***TCP/1433*** between the application server app and the database server db.

The second, fourth, and sixth policies prevent the firewall to drop a session for which does not see the initial connection to the protected resource. This could happen if the protected resource has not sent any traffic previous to this point.

Refer to the following images for more clarity about the above policies and interface configuration.

Figure 9-17: Firewall Policy configuration



	Name	Tags	Type	Source Zone	Source Address	Destination Zone	Destination Address	Rule Usage Hit Count	Rule Usage Last Hit	Rule Usage First Hit	Application	Service	Action
1	vl 150 allow	offload	universal	vl 1200	150.0.1.0/30	phy int	160.0.5.1/32	0	-	-	any	UDP_dst_2000-1	Allow
2	test	offload	universal	vl 1200	170.0.1.0/30	any	any	0	-	-	any	application-default	Allow
3	vl 150 drop	offload	universal	vl 1200	150.0.1.0/30	any	any	0	-	-	any	udp_10000 UDP_dst_2000-1	Drop
4	vl 150 offload	offload	universal	vl 1200	150.0.1.0/29	any	any	0	-	-	any	application-default	Allow
5	vl 170-171	mss	universal	vl 1200	170.0.1.0/30	phy int	171.0.7.0/30	0	-	-	any	udp 17100	Allow
6	vl 150	mss	universal	vl 1200	150.0.1.0/27	any	any	0	-	-	any	application-default	Allow
7	intrazone-default	none	intrazone	any	any	(intrazone)	any	0	-	-	any	any	Allow
8	interzone-default	none	interzone	any	any	any	any	160	2019-03-11 13:38:35	2019-02-11 20:03:33	any	any	Allow

Figure 9-18: Firewall Interface Configuration



Interface	Interface Type	Management Profile	Link State	IP Address	Virtual Router	Tag	VLAN / Virtual-Wire	Security Zone	Features	Comment
ethernet1/1	Aggregate (ae1)			none	none	Untagged	none	none		Po 100 - near interface
ethernet1/2	Aggregate (ae1)			none	none	Untagged	none	none		po 100 - near interface
ethernet1/3	Aggregate (ae2)			none	none	Untagged	none	none		po 101 - far interface
ethernet1/4	Aggregate (ae2)			none	none	Untagged	none	none		po 101 - far interface
ethernet1/5	Layer2			none	none	Untagged	none	none		
ethernet1/6	Virtual Wire			none	none	Untagged	none	none		
ethernet1/7	Tap			none	none		none	none		
ethernet1/8	Layer2			none	none	Untagged	none	none		
ethernet1/9	Layer2			none	none	Untagged	none	none		
ethernet1/10	Aggregate (ae7)			none	none	Untagged	none	none		
ethernet1/11	Aggregate (ae8)			none	none	Untagged	none	none		
ethernet1/12	Aggregate (ae7)			none	none	Untagged	none	none		
ethernet1/13	Aggregate (ae8)			none	none	Untagged	none	none		
ethernet1/14	Tap			none	none		none	none		
ethernet1/15	Virtual Wire			none	none	Untagged	none	none		
ethernet1/16	Virtual Wire			none	none	Untagged	none	none		
ethernet1/17	Layer3	allow ping		none	none	Untagged	none	none		
ethernet1/17.1200	Layer3	allow ping		199.2.0.2/21 199:2::100:2/64	default	1200	none	vl 1200		
ethernet1/17.1201	Layer3	allow ping		199.2.8.2/21	default	1201	none	vl 1201		

Create a rule that Arista MSS will use to intercept and redirect traffic and add a firewall policy with the default Arista_MSS tag as shown in the example above. MSS intercepts all traffic from endpoints identified in policies that match the tag values configured in CVX. The firewall will apply all rules (tagged or untagged) to all traffic.

Note: LLDP should always be enabled on the firewall interfaces attached to the service switches. To minimize reconvergence time on the network changes, reduce the LLDP transmit interval and hold time multiples on the firewall, while keeping the LLDP hold time above the LLDP timer configured on the connected Arista switches.

Alternatively, the **`device interface map`** command can be used on CVX to manually map a device to Arista switch interfaces. To map multiple devices, add a mapping entry for each device.

```
dynamic device-set fw1
device dc-firewall-1
map device-interface ethernet1/1 switch 00:1c:73:7e:21:bb interface Ethernet1
map device-interface ethernet1/2 switch 00:1c:73:7e:21:bb interface Ethernet9
```

The first policy ***untrust_to_web1*** is from the untrust zone to the web1 zone, that allows HTTPS traffic from anywhere to the web server web.

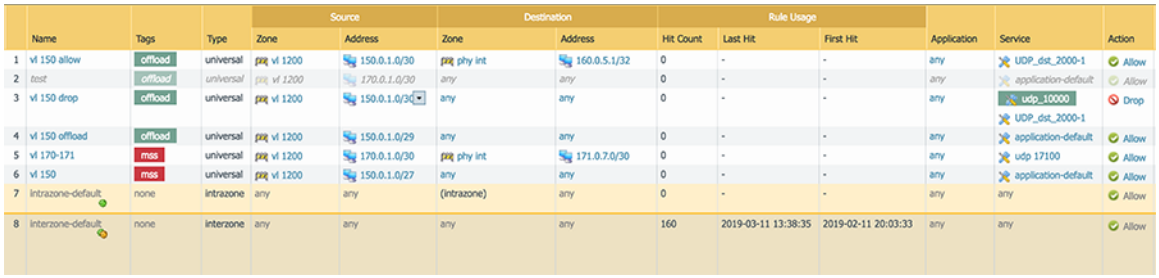
The third policy ***web2_to_app1*** is from the web2 zone to the app1 zone that allows HTTP traffic between the web server web and the application server app.

The fifth policy ***app2_to_db1*** is from the app2 zone to the db1 zone that allows database traffic on port TCP/1433 between the application server app and the database server db.

The second, fourth, and sixth policies prevent the firewall to drop a session for which does not see the initial connection to the protected resource. This could happen if the protected resource has not sent any traffic previous to this point.

Refer to the following images for more clarity about the above policies and interface configuration.

Figure 9-19: Firewall Policy Configuration



				Source		Destination		Rule Usage					
	Name	Tags	Type	Zone	Address	Zone	Address	Hit Count	Last Hit	First Hit	Application	Service	Action
1	vl 150 allow	offload	universal	vl 1200	150.0.1.0/30	phy int	160.0.5.1/32	0	-	-	any	UDP_dst_2000-1	Allow
2	test	offload	universal	vl 1200	170.0.1.0/30	any	any	0	-	-	any	application-default	Allow
3	vl 150 drop	offload	universal	vl 1200	150.0.1.0/30	any	any	0	-	-	any	udp_10000 UDP_dst_2000-1	Drop
4	vl 150 offload	offload	universal	vl 1200	150.0.1.0/29	any	any	0	-	-	any	application-default	Allow
5	vl 170-171	mss	universal	vl 1200	170.0.1.0/30	phy int	171.0.7.0/30	0	-	-	any	udp 17100	Allow
6	vl 150	mss	universal	vl 1200	150.0.1.0/27	any	any	0	-	-	any	application-default	Allow
7	intrazone-default	none	intrazone	any	any	(intrazone)	any	0	-	-	any	any	Allow
8	interzone-default	none	interzone	any	any	any	any	160	2019-03-11 13:38:35	2019-02-11 20:03:33	any	any	Allow

Figure 9-20: Firewall Interface Configuration



Interface	Interface Type	Management Profile	Link State	IP Address	Virtual Router	Tag	VLAN / Virtual-Wire	Security Zone	Features	Comment
ethernet1/1	Aggregate (ae1)			none	none	Untagged	none	none		Po 100 - near interface
ethernet1/2	Aggregate (ae1)			none	none	Untagged	none	none		po 100 - near interface
ethernet1/3	Aggregate (ae2)			none	none	Untagged	none	none		po 101 - far interface
ethernet1/4	Aggregate (ae2)			none	none	Untagged	none	none		po 101 - far interface
ethernet1/5	Layer2			none	none	Untagged	none	none		
ethernet1/6	Virtual Wire			none	none	Untagged	none	none		
ethernet1/7	Tap			none	none		none	none		
ethernet1/8	Layer2			none	none	Untagged	none	none		
ethernet1/9	Layer2			none	none	Untagged	none	none		
ethernet1/10	Aggregate (ae7)			none	none	Untagged	none	none		
ethernet1/11	Aggregate (ae8)			none	none	Untagged	none	none		
ethernet1/12	Aggregate (ae7)			none	none	Untagged	none	none		
ethernet1/13	Aggregate (ae8)			none	none	Untagged	none	none		
ethernet1/14	Tap			none	none		none	none		
ethernet1/15	Virtual Wire			none	none	Untagged	none	none		
ethernet1/16	Virtual Wire			none	none	Untagged	none	none		
ethernet1/17	Layer3	allow ping		none	none	Untagged	none	none		
ethernet1/17.1200	Layer3	allow ping		199.2.0.2/21 199:2::100:2/64	default	1200	none	vl 1200		
ethernet1/17.1201	Layer3	allow ping		199.2.8.2/21	default	1201	none	vl 1201		

Create a rule that Arista MSS will use to intercept and redirect traffic and add a firewall policy with the default ***Arista_MSS*** tag as shown in the example above. MSS intercepts all traffic from endpoints identified in policies that match the tag values configured in CVX. The firewall will apply all rules (tagged or untagged) to all traffic.

Note: LLDP should always be enabled on the firewall interfaces attached to the service switches. To minimize reconvergence time on the network changes, reduce the LLDP transmit interval and hold time multiples on the firewall, while keeping the LLDP hold time above the LLDP timer configured on the connected Arista switches.

Alternatively, the **`device interface map`** command can be used on CVX to manually map a device to Arista switch interfaces. To map multiple devices, add a mapping entry for each device.

```
dynamic device-set fw1
device dc-firewall-1
map device-interface ethernet1/1 switch 00:1c:73:7e:21:bb interface Ethernet1
map device-interface ethernet1/2 switch 00:1c:73:7e:21:bb interface Ethernet9
```

9.3.4.4 MSS Integration with Check Point

Macro Segmentation Service (MSS) is configurable for Check Point Software Technologies (Check Point) Firewalls. The configuration and deployment requires the use of Check Point Management Server (Gaia), a security management platform which allows central management of Check Point gateway security devices.

Requirements

The following requirements apply to the deployment.

- ***MSS version R80.30 version 1.5*** and above and a special URL access on the Management Server using a Gateway API provided by Check Point.
- ***Gateway version R80.30*** with API version 1.2 and above.
- ***Check Point Management Server release R31*** and above.

Configuration and Deployment

The following components of the solution require configuration:

- Check Point Gateway firewalls
- Check Point Management Server
- Arista leaf switches
- CVX

Check Point Firewalls (Gateways)

Interface Configuration

Configure IPv4 addresses on the routed L3 interfaces on the firewall interfaces connected to the Arista TORs.

IPv4 Static Routes Configuration

Configure IPv4 static routes to include routes to all subnets of the hosts which MSS will be intercepting either using a WebUI or CLI as shown below. The nexthop gateway addresses are the gateway of the subnet to which the firewall interfaces. The static route information is used by MSS to identify which firewall interface is connected to the subnet to which the intercepted traffic needs to be forwarded.

```
set static-route 192.0.2.0/24 nexthop gateway address 192.0.2.155 on
```

The following displays the configuration.

```
gateway1>show route static
Codes: C - Connect ed, S - Static, R - RIP, B - BGP (D - Default),
       O - OSPF IntraArea (IA - InterArea, E - External, N - NSSA),
       A - Aggregate, K - Kernel Remnant, H - Hidden, P - Suppressed,
       U - Unreachable, i - Inactive

S               0.0.0.0/0           via 172.2.18.12, Mgmt, cost 0, age 3134690
S               10.6.10.0/24        via 10.6.100.2, eth1, cost 0, age 3134690
S               10.6.20.0/24        via 10.6.200.2, eth2, cost 0, age 3134690
```

Check Point Management Server Configuration

The Check Point firewall devices intended to be used with Arista MSS must be registered and fully manageable via a Check Point Management Server.

- Identify or define a new security policy network layer to be considered by MSS where 'TestPolicy' is the security policy network layer that is referenced in the CVX configuration.
- Create firewall access rules (to be used by Arista MSS).
- In the access rule, the supported source and destination object types are Host, Network, and Security Zone.

- In the “Services & Applications,” the following services are supported: 1: ICMP, IGMP, IPv4, TCP, EGP, UDP, IPv6, RSVP, GRE, OSPFIGP, SCTP.
- Add tags in the policy comments/description field in this format: "tags(<tag1>, <tag2>, ...)", e.g. “tags(Arista_MSS1, Arista_MSS2)”
  - Arista MSS inspects management server access rules that have an embedded "tags( )" string in the comments field. Individual tags are extracted from within the enclosing parentheses and compared with the tags configured in the Arista MSS device-set on CVX.

1-to-1 HA Cluster Configuration

The following figure shows the 1-to-1 HA cluster. The HA interface pairs connected to the Arista switches should have Virtual IP addresses where the intercepted traffic will be forwarded. The active firewall sends out a GARP with its own MAC to indicate where traffic sent to the VIP should be forwarded.

Arista Leaf Switches Configuration

The following configures the switch interfaces connected to the firewall.

```
switchport trunk native vlan <interface vlan>
switchport mode trunk
spanning-tree portfast
spanning-tree bpdufilter enable

interface Vlan<interface vlan>
   ip address virtual <interface IPv4 address>/<mask>
```

CVX Configuration

The following displays the CVX configuration with Standalone Check Point firewall.

```
!! Standalone firewall
cvx
   no shutdown
   service mss
      no shutdown
      !
      dynamic device-set chkpt
         device <management-server-ip-or-dnsName>
            username admin password 7 PKigsm//o3IcnW5rqoZXWQ==
            protocol https 4434   (or the configured https port like 443)
            group <management-server-network-layer>
         !
         device member <checkpoint-device-name>
            map device-interface eth1 switch 00:1c:73:7e:28:11 interface
 Ethernet39
            map device-interface eth2 switch 00:1c:73:7e:28:11 interface
 Ethernet40
         type check-point management-server
         policy tag offload Arista_MSS_offload
         policy tag redirect Arista_MSS
         state active
```

The checkpoint-device-name used in the **`device member`** command is the name used in the Management Server to identify that firewall. A sample CVX configuration with Check Point firewalls in 1-to-1 High Availability cluster configuration will include more than one device member as follows:

```
!! HA Active/Passive firewall pair
cvx
   no shutdown
   service mss
      no shutdown
      !
```

```
      dynamic device-set chkpt
         device <management-server-ip-or-dnsName>
            username admin password 7 PKigsm//o3IcnW5rqoZXWQ==
            protocol https 4434   (or the configured https port like 443)
            group <management-server-network-layer>
         !
         device member <checkpoint-device1-name>
            map device-interface eth1 switch 00:1c:73:7e:28:11 interface
Ethernet39
            map device-interface eth2 switch 00:1c:73:7e:28:11 interface
Ethernet40
         device member <checkpoint-device2-name>
            map device-interface eth1 switch 00:1c:73:7e:28:11 interface
Ethernet41
            map device-interface eth2 switch 00:1c:73:7e:28:11 interface
Ethernet42
         type check-point management-server
         policy tag offload Arista_MSS_offload
         policy tag redirect Arista_MSS
         state active
```

9.3.4.5 MSS for Layer 3 Firewall Enhancements

The **verbatim** qualifier enhances the Macro Segmentation Service (MSS) with two policy actions: **redirect** and **offload**. For firewall policies tagged with the **redirect** tag, MSS extracts IP addresses from the policy and forwards all traffic destined to or generated from that set of IP addresses to the firewall. The additional **verbatim** tag, redirecting bidirectional traffic is restricted to the subset that matches the additional qualifiers of a firewall policy to a firewall (such as the source, destination IP addresses or subnets, protocol, L4 ports).

The **verbatim** tag can also be paired with the **offload** tag for a policy which installs necessary DirectFlow rules at the TORs to drop or allow the traffic matching the exact qualifiers in the policy definition. If the **verbatim** tag is not used with the offload tag, the behavior is to offload enforcement for all traffic matching the specific policy rule, while redirecting the remainder of the (non-matching) traffic to the firewall to ensure the security policy for the protected host remains in compliance. The addition of the **verbatim** tag removes this implicit redirection

Configuring for Verbatim Use

Firewall Configuration

The verbatim is a modifier of the original policy enforcement scheme and works with multiple firewalls such as those from Palo Alto Networks and Fortinet.

Policy Semantics

IP address extraction for **redirect** or **redirect** tag:

- If IP addresses are specified in source or destination field, Mss extracts those for redirection.
- If no IP addresses are specified, then Mss extracts the subnets corresponding to the source and destination zone for redirection.
- If no zones are specified, then Mss extracts all subnets in the **default** virtual-router for redirection.

Constraints on **offload** tag policies:

- Must have IP address specified in source or destination field if the corresponding zone is an **external** zone (zone towards default route).

Constraints on **redirect verbatim** tag policies:

- Must have IP address specified in source or destination field if the corresponding zone is an **external** zone (zone towards default route).

- Must have either zone or IP specified in both source and destination field. ‘Any’, ‘All’, or similar constructs are not supported for source or destination fields.

Policies with broadcast or multicast destination:

Only **offload** and **offloadverbatim** tags are supported for policies with IPv4 broadcast or IPv4 multicast destination.

CVX Configuration

The following configuration commands set 'tag-list' as the verbatim modifier on a per device basis for the **redirect** and **offload** tags.

```
cvx
   service mss
      dynamic device-set <device-set-name>
         device <device-name>
           [no | default] policy tag redirect <tag-list>
           [no | default] policy tag offload <tag-list>
           [no | default] policy tag modifier verbatim <tag-list>
```

TCAM Profile Configuration

The following depicts a recommended TCAM profile to be used with MSS.



Note: This is an example for some of the devices that are currently supported.

```
hardware tcam
   profile direct-flow-mssl3-vxlan
      feature acl port ip
         sequence 50
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops l4-src-
port src-ip tcp-control ttl
         action count drop
         packet ipv4 forwarding bridged
         packet ipv4 forwarding routed
         packet ipv4 forwarding routed multicast
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet ipv4 non-vxlan forwarding routed decap
         packet ipv4 vxlan eth ipv4 forwarding routed decap
         packet ipv4 vxlan eth ipv6 forwarding routed decap
         packet ipv4 vxlan forwarding bridged decap
      feature acl port ip egress mpls-tunnelled-match
         sequence 100
      feature acl port ipv6
         sequence 30
         key field dst-ipv6 ipv6-next-header ipv6-traffic-class l4-dst-port l4-
ops-3b l4-src-port
         src-ipv6-high src-ipv6-low tcp-control
         action count drop
         packet ipv6 forwarding bridged
         packet ipv6 forwarding routed
         packet ipv6 forwarding routed multicast
         packet ipv6 ipv6 forwarding routed decap
      feature acl port mac
         sequence 60
         key size limit 160
         key field dst-mac ether-type src-mac
         action count drop
         packet ipv4 forwarding bridged
```

```
         packet ipv4 forwarding routed
         packet ipv4 forwarding routed multicast
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet ipv4 non-vxlan forwarding routed decap
         packet ipv4 vxlan eth ipv4 forwarding routed decap
         packet ipv4 vxlan forwarding bridged decap
         packet ipv6 forwarding bridged
         packet ipv6 forwarding routed
         packet ipv6 forwarding routed decap
         packet ipv6 forwarding routed multicast
         packet ipv6 ipv6 forwarding routed decap
         packet mpls forwarding bridged decap
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding mpls
         packet mpls non-ip forwarding mpls
         packet non-ip forwarding bridged
      feature acl subintf ip
         sequence 45
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops-18b l4-
src-port src-ip tcp-control ttl
         action count drop
         packet ipv4 forwarding routed
      feature acl subintf ipv6
         sequence 20
         key field dst-ipv6 ipv6-next-header l4-dst-port l4-src-port src-ipv6-
high src-ipv6-low tcp-control
         action count drop
         packet ipv6 forwarding routed
      feature acl vlan ip
         sequence 40
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops-18b l4-
src-port src-ip tcp-control ttl
         action count drop
         packet ipv4 forwarding routed
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet ipv4 non-vxlan forwarding routed decap
         packet ipv4 vxlan eth ipv4 forwarding routed decap
         packet ipv4 vxlan eth ipv6 forwarding routed decap
      feature acl vlan ipv6
         sequence 15
         key field dst-ipv6 ipv6-next-header l4-dst-port l4-src-port src-ipv6-
high src-ipv6-low tcp-control
         action count drop
         packet ipv6 forwarding routed
         packet ipv6 ipv6 forwarding routed decap
      feature acl vlan ipv6 egress
         sequence 25
key field dst-ipv6 ipv6-next-header ipv6-traffic-class l4-dst-port l4-src-port
 src-ipv6-high src-ipv6-low
tcp-control
         action count drop
         packet ipv6 forwarding routed
      feature flow
         key size limit 160
         key field dst-ip ether-type in-port ip-protocol l4-dst-port l4-src-
port src-ip
         action drop redirect set-fwd-header
         packet ipv4 forwarding bridged
         packet ipv4 forwarding routed
      feature forwarding-destination mpls
```

```
         sequence 105
      feature mpls
         sequence 5
         key size limit 160
         action drop redirect set-ecn
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding mpls
         packet mpls non-ip forwarding mpls
      feature mpls pop ingress
         sequence 95
      feature pbr mpls
         sequence 70
         key size limit 160
         key field mpls-inner-ip-tos
         action count drop redirect
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding mpls
         packet mpls non-ip forwarding mpls
      feature tunnel vxlan
         sequence 55
         key size limit 160
         key field in-port vxlan-inner-etype vxlan-inner-ip-options vxlan-
inner-ip-ttl
         packet ipv4 vxlan eth ipv4 forwarding routed decap
         packet ipv4 vxlan eth ipv6 forwarding routed decap
         packet ipv4 vxlan forwarding bridged decap
      feature tunnel vxlan routing
         sequence 10
         packet ipv4 forwarding routed
         packet ipv4 non-vxlan forwarding routed decap
         packet ipv4 vxlan eth ipv4 forwarding routed decap
         packet ipv4 vxlan eth ipv6 forwarding routed decap
```

The following displays the profile. The platform does not support any arbitrarily created PMF profile. If the PMF profile cannot be programmed, the show command will print '**ERROR**' in the status column.

```
switch# show hardware tcam profile
                    Configuration            Status
FixedSystem         direct-flow-mssl3-vxlan  direct-flow-mssl3-vxlan
```

Limitations

- DirectFlow needs to be enabled at the TOR so that the policies enforced by MSS are correctly programmed.
- *Group* option is available only for some switches.
- Deployments with a mix of switches require special considerations. The following table summarizes supported configurations in different deployment models.

Table 5: Configuration Summary

Both compute and service TORs: DCS-7050X, DCS-7050X2, DCS-7050X3, DCS-7060X, DCS-7060X2	group, verbatim	• redirect • offload • redirect, verbatim • offload, verbatim
Both compute and service TORs: DCS-7020R, DCS-7280R, DCS-7280R2, DCS-7500R, DCS-7500R2	verbatim	• redirect, verbatim • offload, verbatim
Both compute and service TORs: DCS-7050X, DCS-7050X2, DCS-7050X3, DCS-7060X, DCS-7060X2, DCS-7020R, DCS-7280R, DCS-7280R2, DCS-7500R, DCS-7500R2	verbatim	• redirect, verbatim • offload, verbatim
DCS-7050X, DCS-7050X2, DCS-7050X3, DCS-7060X, DCS-7060X2 as compute TOR and other series as service TOR (with no intercepted hosts connected).	group, verbatim	• redirect • offload • redirect, verbatim • offload, verbatim

Backward Compatibility and Other Considerations

For existing deployments, (where any of DCS-7020R, DCS-7280R, DCS-7280R2, DCS-7500R, DCS-7500R2 switch platforms are used in the service rack), in order to upgrade hitlessly, upgrade the CVX cluster first and execute the following command prior to upgrading EOS on any switch:

```
cvx
   service mss
      policy enforcement rules group verbatim
```

The command **[no|default] policy enforcement rules {group verbatim | verbatim}** disables / enables policy enforcement.

Displaying CVX Status

The following displays the status of the mss policy.

```
switch#show service mss policy

<--snip-->
                Macro-Segmentation L3 Policy Table
------------------------------------------------------------------------------
Source             Device      Policy        Offload      Redirect
 Unconverged
                                              status       status       IPs
---------------  ---------   ----------    ----------   -----------
 ----------
PaloAltoFirewall   fwpan1     policy1       N/A          Active      N/A
PaloAltoFirewall   fwpan1     policy2       Active       N/A         N/A
PaloAltoFirewall   fwpan1     policy3       Active       Active      0 of 2
PaloAltoFirewall   fwpan1     policy4       N/A          Active      0 of 2
```

The following displays the status of the mss policy in more detail.

```
switch#show service mss policy detail
------------------------------------------------------------------------
Source: PaloAltoFirewall
------------------------------------------------------------------------
  Device: fwpan1
   Policy (L3): policy1
      Offload Status: N/A
      Redirect Status: Active
      Tags: MSS_redirect, MSS_verbatim
      Policy Modifier: Verbatim
      VRF: default
   Policy (L3): policy2
      Offload Status: Active
      Redirect Status: N/A
      Tags: MSS_offload, MSS_verbatim
      Policy Modifier: Verbatim
      VRF: default
   Policy (L3): policy3
      Offload Status: Active
      Redirect Status: Active
      Tags: MSS_offload
      VRF: default
      IP Addresses:
        Active: 10.10.10.1
        Active: 10.10.20.1
   Policy (L3): policy4
      Offload Status: N/A
      Redirect Status: Active
      Tags: MSS_redirect
      VRF: default
      IP Addresses:
        Active: 10.10.10.1
        Active: 10.10.10.2
```

Displaying Flow Information Details on TOR Switch

```
switch#show directflow detail
Flow default:spm:fwpan1:30000::10.10.20.2/32::10.10.20.3/32::::nh-1.100.0.2:
(Flow programmed)
  persistent: False
  priority: 30000
  priorityGroupType: default
  hard timeout: 0
  idle timeout: 0
  match:
    Ethernet type: IPv4
    source IPv4 address: 10.10.20.2/255.255.255.255
    destination IPv4 address: 10.10.20.3/255.255.255.255
    IPv4 protocol: TCP
    destination TCP/UDP port: 22
  actions:
    output nexthop: 1.10.100.2
  source: mssl3
  matched: 0 packets, 0 bytes

Flow default:spm:fwpan1:30000::10.10.20.3/32::10.10.20.2/32::::nh-1.100.0.2:
(Flow programmed)
  persistent: False
  priority: 30000
  priorityGroupType: default
  hard timeout: 0
```

```
  idle timeout: 0
  match:
    Ethernet type: IPv4
    source IPv4 address: 10.10.20.3/255.255.255.255
    destination IPv4 address: 10.10.20.2/255.255.255.255
    IPv4 protocol: TCP
    source TCP/UDP port: 22
  actions:
    output nexthop: 1.10.100.2
  source: mssl3
  matched: 0 packets, 0 bytes
<--snip-->
```

9.3.4.6 MSS Commands

Configuration Commands

- dynamic device-set
- exception device
- group
- name-resolution interval (CVX-OpenStack)
- service mss
- state
- tag
- type palo-alto

CVX Show Commands

- show service mss dynamic device-set
- show service mss policy
- show service mss status
- show service mss zone

9.3.4.6.1 dynamic device-set

The **`dynamic device-set`** command configures a device such as a firewall to communicate with the MSS in the MSS configuration mode.

The **`no dynamic device-set`** command removes a previously configured device from the MSS configuration and returns to the CVX mode.

Command Mode

MSS Configuration

Command Syntax

`dynamic device-set` *device-set_name*

`no dynamic device-set` *device-set_name*

Parameters

device-set_name a unique name for the device set.

Example

This example creates a set of firewalls with the name ***panfw1***.

```
cvx#configure
cvx(config)#cvx
cvx(config-cvx)#no shutdown
cvx(config-cvx)#service mss
```

```
cvx(config-cvx-mss)#no shutdown
cvx(config-cvx-mss)#vni range 30000-40000
cvx(config-cvx-mss)#dynamic device-set panfw1
cvx(config-cvx-mss-panfw1)#
```

Note: The **`vni range`** command configures a range of VXLAN Network Identifiers (VNI) that MSS uses to tunnel traffic to the firewall. If VNI range is not configured, the default VNIs in the range of ***1*** to ***16777214*** are used.

9.3.4.6.2 exception device

The **`exception device`** command bypasses or continues redirecting traffic to service device such as a firewall if the service device control-plane API is unreachable after initial policies have been processed.

The no exception device command.

Command Mode

MSS Configuration

Command Syntax

`exception device unreachable` [bypass | redirect]

`no exception device unreachable` [bypass | redirect]

`default exception device unreachable bypass`

Parameters

- **device**: service device in the device set.
- **`unreachable`**: service device control-plane API is unreachable.
- **bypass**: bypass the service device.
- **redirect**: continue redirecting traffic to the service device.

Example

This example redirects traffic to the service device.

```
cvx#configure
cvx(config)#cvx
cvx(config-cvx)#no shutdown
cvx(config-cvx)#service mss
cvx(config-cvx-mss)#no shutdown
cvx(config-cvx-mss)#vni range 30000-40000
cvx(config-cvx-mss)#dynamic device-set fw
cvx(config-cvx-mss-fw)#device firewall-dc7
cvx(config-cvx-mss-fw)#username admin password 7 PKigsmo3IcnW5rqoZXWQ
cvx(config-cvx-mss-fw)#state active
cvx(config-cvx-mss-fw)#type palo-alto firewall
cvx(config-cvx-mss-fw)#exception device unreachable redirect
```

9.3.4.6.3 group

The **`group`** command configures the Panorama device group name to be used with MSS.

The no group command removes the group from the MSS configuration when the Panorama firewall manager is used.

See the type palo-altocommand for more information about the firewall manager.

Command Mode

Device-set mode

Command Syntax

`group` *group_name*

`no group` *group_name*

Parameters

group_name the name of the group.

Example

This command configures the group name as ***mssDevices***.

```
cvx(config)#cvx
cvx(config-cvx)#service mss
cvx(config-cvx-mss)#dynamic device-set pano2
cvx(config-cvx-mss-pano2)#type palo-alto panorama
cvx(config-cvx-mss-pano2)#device myPanorama
cvx(config-cvx-mss-pano2-myPanorama)#group mssDevices
```

9.3.4.6.4 name-resolution interval (CVX-OpenStack)

The **`name-resolution interval`** command specifies the period between consecutive requests that the OpenStack controller sends to the Keystone service for VM and tenant name updates. Keystone is OpenStack's authentication and authorization service.

The default period is ***21600*** seconds (6 hours).

The name-resolution force (CVX-OpenStack) command performs an immediate update, as opposed to waiting for the periodic update.

Command Mode

CVX-OpenStack Configuration

Command Syntax

`name-resolution interval` period

Parameters

period: Keystone identity service polling interval (seconds).

Comment

`service openstack` places the switch in ***CVX-OpenStack*** configuration mode.

Example

These commands set the name resolution interval period at five hours.

```
switch(config)#cvx
switch(config-cvx)#service openstack
switch(config-cvx-openstack)#name-resolution interval 18000
switch(config-cvx-openstack)#
```

9.3.4.6.5 service mss

The **`service mss`** command enters the MSS configuration sub-mode.

The no service mss command exits the MSS configuration mode and returns to the CVX mode.

Command Mode

CVX Configuration

Command Syntax

```
service mss

no service mss

default service mss
```

Example

This example enables MSS on CVX and enters the MSS config mode.

Note: The **`no shutdown`** command enables MSS on the CloudVision eXchange (CVX).

```
cvx#configure
cvx(config)#cvx
cvx(config-cvx)#no shutdown
cvx(config-cvx)#service mss
cvx(config-cvx-mss)#no shutdown
```

9.3.4.6.6 show service mss dynamic device-set

The **`show service mss dynamic device-set`** command displays detailed information about a specific service device set. Information such as device group members, high availability, network, resource details are displayed.

Note: Interfaces from multiple switches can be placed in the same zone by the device.

Command Mode

EXEC

CVX Configuration

Command Syntax

`Show service mss dynamic device-set` *device_set_name* [device *device_name* [group-members | high-availability | neighbors | network | policies | resources]]

Parameters

- ***device_set_name*** defines the device set name.
- **device *device name*** defines the service device properties such as the DNS hostname or IP address of the service device.
- **group members** lists device-group members for an aggregation manager.
- **high-availability** displays service device high availability information.
- **neighbors** displays the service devices ethernet interface neighbor information.
- **network** displays the service devices network interface information.
- **policies** displays the list of policies read from service device that have the MSS tag.
- **resources** displays the service devices system resource information.

Related Commands

- show service mss status
- show service mss policy

Examples

- This command displays information about interfaces that are placed in a zone by the ***device1***.

```
switch#show service mss zone
Source: static
------------------------------------------------
```

```
Device: device1
```

- This command displays information about interfaces that are placed in a zone by the ***device1***.

```
switch#show service mss zone
Source: static
-----------------------------------------------
Device: device1
Zone: zone1
Switch: 00:00:00:00:00:01
Hostname: switch1.arista.com
Interfaces:
Ethernet1/1
Allowed VLAN: 1000-1010
Port-Channel2/1:
Allowed VLAN: 1000-2000
Switch: 00:00:00:00:00:02
Hostname: switch2.arista.com
Interfaces:
Ethernet10/1
Allowed VLAN: 1000-1010
Zone: zone2
Switch: 00:00:00:00:00:01
Hostname: switch1.arista.com
Interfaces:
Ethernet10/1
Allowed VLAN: 1000-1010
Ethernet 20/1
Allowed VLAN: 1000-2000
```

9.3.4.6.7 show service mss policy

The **`show service mss policy`** command displays generic information about the configuration and operational state of the macro-segmentation service (MSS) policies on a device.

Command Mode

EXEC

CVX Configuration

Command Syntax

`show service mss policy` [[device *device_name*][name *policy-name*][source (static | *plugin_name*)]]

Parameters

- **device *device name*** defines the service device name.
- **name *policy-name*** the filter policy name.
- **source** the source of the policy.
- **static** the policy configured using the command line interface.
- ***plugin_name*** the service device type.

Related Commands

- show service mss status
- show service mss zone

Example

This command displays information about the MSS policy ***policy1*** enabled on the device.

```
cvx#show service mss policy name policy1
Source  Device    Policy  Config   Status
```

```
------  --------  ------  -------  -------------  -------------
vendor  Firewall  pan100  policy1  Enabled        Initialized
```

The **Config** column indicates the configuration state of a policy. The different states are: **Enabled**, **dry run**, and **disabled** states.

The **Status** column indicates the operational state of a policy. The different status types are **initialized**, **pending**, **initializing**, **active**, **reinitializing**, **dry-run Complete**, and **deactivating**.

9.3.4.6.8 show service mss status

The **show service mss status** command displays the status of a macro-segmentation service (MSS) on the device.

Command Mode

EXEC

CVX Configuration

Command Syntax

```
show service mss status
```

Related Commands

- show service mss policy
- show service mss zone

Examples

- This command displays the MSS status on the device as Enabled.

```
switch#show service mss status
State: Enabled
Service VNIs: 1500-1600,1800,1900-2000
```

- This command displays the MSS status on the device as Disabled.

```
switch#show service mss status
State: Disabled
Service VNIs: 1-16777214
```

9.3.4.6.9 show service mss zone

The **show service mss zone** command displays information about the interfaces that are placed in a single zone by the service device. Along with the show service mss policy command, we can use this command to identify issues with the policy configuration.

Interfaces from multiple switches can be placed in the same zone by the device.

Command Mode

EXEC

CVX Configuration

Command Syntax

`show service mss zone` [[device *device_name*]|[name *zone_name*]|[source (static | *dynamic_source*)]]

Parameters

- **device *device name*** defines the service device properties.
- **name *policy-name*** the filter zone name.

- **source** the source of the zone.
- **static** the zone configured using the command line interface.
- ***dynamic_source*** the service device type.

Related Commands

- show service mss status
- show service mss policy

Example

This command displays information about interfaces that are placed in a zone by the ***device1***.

```
switch#show service mss zone
Source: static
--------------------------------------
Device: device1
Zone: zone1
Switch: 00:00:00:00:00:01
Hostname: switch1.arista.com
Interfaces:
Ethernet1/1
Allowed VLAN: 1000-1010
Port-Channel2/1:
Allowed VLAN: 1000-2000
Switch: 00:00:00:00:00:02
Hostname: switch2.arista.com
Interfaces:
Ethernet10/1
Allowed VLAN: 1000-1010
Zone: zone2
Switch: 00:00:00:00:00:01
Hostname: switch1.arista.com
Interfaces:
Ethernet10/1
Allowed VLAN: 1000-1010
Ethernet 20/1
Allowed VLAN: 1000-2000
```

9.3.4.6.10 state

The **state** command configures device set as active or disabled or suspended state.

The no state command disables the previously configured state of the device set.

Command Mode

MSS Configuration

Command Syntax

`state` [active | shutdown | suspend]

`no state`

Parameters

- **active**: the active state of the device set. Policy monitoring and network traffic redirection are enabled.
- **shutdown**: the disabled state of the device set. Policy monitoring and network traffic redirection is stopped.
- **suspend**: the suspended state of the device set. Policy monitoring is suspended but there is no change in the existing traffic redirection.

Example

This output example configures the device set state as active.

```
cvx#configure
cvx(config)#cvx
cvx(config-cvx)#no shutdown
cvx(config-cvx)#service mss
cvx(config-cvx-mss)#no shutdown
cvx(config-cvx-mss)#vni range 30000-40000
cvx(config-cvx-mss)#dynamic device-set panfw1
cvx(config-cvx-mss-panfw1)#tag Arista_MSS
cvx(config-cvx-mss-panfw1)#type palo-alto firewall
cvx(config-cvx-mss-panfw1)#state active
```

9.3.4.6.11 tag

The **`tag`** command specifies the tag or tags that MSS searches when it is reading the security policy from the firewall or firewall manager in the ***dynamic device-se***t configuration mode. You can specify more than one tag as well.

The no tag command removes the tag from the MSS configuration.



Note: The tag specified should always match with the firewall policy tags in the vendor firewall policy for the MSS to read the policy and set up the intercept.

Command Mode

MSS Configuration

Command Syntax

`tag` *tag_name*

`no tag`

`default tag` Arista_MSS

Parameters

tag_name: a unique name for the tag.

Examples

- This command specifies the tag with the name ***Arista_MSS***.

```
cvx#configure
cvx(config)#cvx
cvx(config-cvx)#no shutdown
cvx(config-cvx)#service mss
cvx(config-cvx-mss)#no shutdown
cvx(config-cvx-mss)#vni range 30000-40000
cvx(config-cvx-mss)#dynamic device-set panfw1
cvx(config-cvx-mss-panfw1)#tag Arista_MSS
```

- This command specifies multiple tags with names ***mss1***, ***mss2***, and ***mss3***.

```
cvx#configure
cvx(config)#cvx
cvx(config-cvx)#no shutdown
cvx(config-cvx)#service mss
cvx(config-cvx-mss)#no shutdown
cvx(config-cvx-mss)#vni range 30000-40000
cvx(config-cvx-mss)#dynamic device-set panfw1
cvx(config-cvx-mss-panfw1)#tag mss1 mss2 mss3
```

9.3.4.6.12 type palo-alto

The **type palo-alto** command configures the firewall type to be used in the MSS configuration.

The **no type palo-alto** command disables the firewall type from the MSS configuration.

Command Mode

MSS Configuration

Command Syntax

`type palo-alto` [firewall | panorama]

`no type palo-alto`

Parameters

- **firewall**: the Palo Alto Networks firewall.
- **panorama**: the Palo Alto Networks Panorama firewall manager.

Example

This command configures the Palo Alto Networks firewall type.

```
cvx#configure
cvx(config)#cvx
cvx(config-cvx)#service mss
cvx(config-cvx-mss)#dynamic device-set panfw1
cvx(config-cvx-mss-panfw1)#type palo-alto firewall
```

Chapter 10

Quality of Service and Traffic Management

This chapter describes Arista's Quality of Service (QoS) implementation and Traffic Management, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Quality of Service
- Traffic Management

10.1 Quality of Service

This chapter describes Arista's Quality of Service (QoS) implementation, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Quality of Service Conceptual Overview
- QoS Configuration: Platform-Independent Features
- QoS Configuration: Arad Platform Switches
- QoS Configuration: Jericho Platform Switches
- QoS Configuration: FM6000 Platform Switches
- QoS Configuration: Petra Platform Switches
- QoS Configuration: Trident and Tomahawk Platform Switches
- QoS Configuration: Trident II and Helix Platform Switches
- Support for Configuring Color Extended Communities
- ACL based QoS Configuration
- Configuring IPv6 Flow Label Matches for QoS
- Differentiated MMU Discard Counters
- Quality of Service Configuration Commands
- Chipset Mapping for QoS

10.1.1 Quality of Service Conceptual Overview

QoS processes apply to traffic that flows through Ethernet ports and control planes. These processes can modify data fields (CoS or DSCP) or assign data streams to traffic classes for prioritized handling. Transmission queues are configurable for individual Ethernet ports to shape traffic based on its traffic class. Many switches also support traffic policies that apply to data that is filtered by access control lists.

The following sections describe QoS features:

- Identifying the Switch Platform
- QoS Data Fields and Traffic Classes
- Transmit Queues and Port Shaping
- Explicit Congestion Notification (ECN)
- ACL Policing
- Quality of Service (QoS) Profiles

10.1.1.1 Identifying the Switch Platform

QoS configuration varies significantly by switch platform. A list of Arista switch model numbers and their corresponding switch platforms (chipsets) can be found in the Chipset Mapping for QoS.

On some switches, the platform can also be determined by entering **`platform ?`** in the CLI.

Example

This command shows that the example switch is running on the Trident platform.

```
switch(config)#platform ?
  trident  Trident chip
switch(config)#
```

10.1.1.2 QoS Data Fields and Traffic Classes

Quality of Service (QoS) defines a method of differentiating data streams to provide varying levels of service to the different streams.

Criteria determining a packet's priority level include packet field contents and the port where data packets are received. QoS settings are translated into traffic classes, which are then used by switches to manage all traffic flows. Traffic flow management varies with each switch platform.

10.1.1.2.1 QoS Data Fields

Quality of service decisions are based on the contents of the following packet fields:

- CoS (three bits): Class of service (CoS) is a 3-bit field in Ethernet frame headers using VLAN tagging. The field specifies a priority value between zero and seven. Class of service operates at Layer 2.
- DSCP (six bits): Differentiated Service Code Point (DSCP) is a 6-bit field in the Type Of Service (TOS) field of IP packet headers.

10.1.1.2.2 Port Settings – Trust Mode and Traffic Class

Ethernet and port channel interfaces support three QoS trust modes:

- **CoS Trust:** Ports use inbound packet CoS field contents to derive the traffic class.
- **DSCP Trust:** Ports use inbound packets DSCP field contents to derive the traffic class.
- **Untrusted:** Ports use their default values to derive the traffic class, ignoring packet contents.

The default mode setting is **CoS trust** for switched ports and **DSCP trust** for routed ports.

Ports are associated with default CoS, DSCP, and traffic class settings; defaults vary by platform.

These sections describe procedures for configuring port settings:

- CoS and DSCP Port Settings – Arad Platform Switches
- CoS and DSCP Port Settings – FM6000 Platform Switches
- CoS and DSCP Port Settings – Petra Platform Switches
- CoS and DSCP Port Settings – Trident and Tomahawk Platform Switches
- CoS and DSCP Port Settings – Trident II and Helix Platform Switches

10.1.1.2.3 Rewriting CoS and DSCP

CoS Rewrite

Switches can rewrite the CoS field for outbound tagged packets. The new CoS value is configurable, and is derived from a data stream's traffic class as specified by the traffic class-to-CoS rewrite map. CoS rewrite is disabled on all the traffic received on CoS trusted ports.

On Arad, Jericho, FM6000, Trident and Tomahawk, Trident II, and Helix platform switches, CoS rewrite can be enabled or disabled on DSCP trusted ports and untrusted ports.

- CoS rewrite is globally enabled by default for packets received on untrusted ports and DSCP trusted ports if at least one port is explicitly configured in **DSCP trust** or **untrusted** mode.

- CoS rewrite is globally disabled by default for packets received on untrusted ports and DSCP trusted ports if there are no ports explicitly configured in **DSCP trust** or **untrusted** mode.

On Petra platform switches, CoS rewrite is always enabled on DSCP trusted ports and untrusted ports.

DSCP Rewrite

Switches can rewrite the DSCP field for outbound IP packets. On FM6000, Trident and Tomahawk, Trident II, and Helix platform switches, DSCP rewrite is disabled by default on all ports and always disabled for traffic received on DSCP trusted ports. On Petra, Arad, and Jericho platform switches, DSCP rewrite is always disabled.

FM6000, Trident and Tomahawk, Trident II, and Helix platform switches provide a command that enables or disables DSCP rewrite for packets received on CoS trusted ports and untrusted ports. The new DSCP value is configurable, based on the data stream's traffic class, as specified by the traffic class-to-DSCP rewrite map.

These sections describe procedures for rewriting CoS and DSCP fields:

- CoS Rewrite – Arad Platform Switches
- CoS and DSCP Rewrite – FM6000 Platform Switches
- CoS Rewrite – Petra Platform Switches
- CoS and DSCP Rewrite – Trident and Tomahawk Platform Switches
- CoS and DSCP Rewrite – Trident II and Helix Platform Switches

10.1.1.2.4 Traffic Classes

Data stream distribution is based on their traffic classes. Data stream management varies by switch platform. Traffic classes are derived from these data stream, inbound port, and switch attributes:

- CoS field contents.
- DSCP field contents.
- Inbound port trust setting.
- CoS default setting (Arad, Jericho, FM6000, Trident and Tomahawk, Trident II, and Helix platform switches).
- DSCP default setting (Arad, Jericho, FM6000, Trident and Tomahawk, and Trident II platform switches).
- Traffic class default setting (Petra platform switches).

When a port is configured to derive a data stream's traffic class from the CoS or DSCP value associated with the stream, the traffic class is determined from a conversion map.

- A CoS-to-traffic class map derives a traffic class from a CoS value.
- A DSCP-to-traffic class map derives a traffic class from a DSCP value.

Map entries are configurable through CLI commands. Default maps determine the traffic class value when CLI map entry commands are not configured. Default maps vary by switch platform.

These sections describe traffic class configuration procedures:

- Traffic Class Derivations – Arad Platform Switches
- Traffic Class Derivations – Jericho Platform Switches
- Traffic Class Derivations – FM6000 Platform Switches
- Traffic Class Derivations – Petra Platform Switches
- Traffic Class Derivations – Trident and Tomahawk Platform Switches
- Traffic Class Derivations – Trident II and Helix Platform Switches

10.1.1.3 Transmit Queues and Port Shaping

Transmit queues are logical partitions of an Ethernet port's egress bandwidth. Data streams are assigned to queues based on their traffic class, then sent as scheduled by port and transmit settings. Support varies by switch platform. A queue's label determines its priority: queues with the suffix ***0*** have the lowest priority.

Parameters that determine transmission schedules include:

- **Traffic class-to-transmit queue** mapping determines the transmit queue for transmitting data streams based on traffic class. The set of available transmit maps vary by switch platforms:
  - **Arad, Jericho, FM6000, Trident II, and Helix platforms:** one map for all unicast and multicast traffic.
  - **Trident and Tomahawk platform:** one map for unicast traffic and one map for multicast traffic.
  - **Petra platform:** one map for unicast traffic. Queue shaping is not available for multicast traffic.
- **Port shaping** specifies a port's maximum egress bandwidth.
- **Queue shaping** specifies a transmit queue's maximum egress bandwidth, and implementation varies by platform.
  - **Trident and Tomahawk platform:** queue shaping is configurable separately for unicast and multicast queues.
  - **Trident II platform:** queue shaping is configurable for transmit queues. Port shaping and queue shaping are supported only in store-and-forward switching mode.
  - **Petra platform:** queue shaping is not available for multicast traffic.
  - **Helix platform:** queue shaping is configurable for transmit queues.
  - **FM6000 platform:** switches do not support simultaneous port shaping and queue shaping. Enabling port shaping on an FM6000 switch disables queue shaping, regardless of the previous configuration.
- **Guaranteed bandwidth** guarantees the allocation of a specified bandwidth for a transmit queue. Guaranteed bandwidth is supported only on Trident II platforms.
- **Queue priority** specifies the priority at which a transmit queue is serviced. The switch defines two queue priority types:
  - ***Strict priority queues*** are serviced in the order of their priority rank - subject to each queue's configured maximum bandwidth. Data is not handled for a queue until all queues with higher priority are emptied or their transmission limit is reached. These queues typically carry low latency real time traffic and require highest available priority.
  - ***Round robin queues*** are serviced simultaneously subject to assigned bandwidth percentage and configured maximum bandwidth. All round robin queues have lower priority than strict priority queues. Round robin queues can be starved by strict priority queues.
- **Queue scheduling** determines how packets from different transmit queues are serviced to be sent out on the port.
- **Queue bandwidth allocation** specifies the time slice (percentage) assigned to a round robin queue, relative to all other round robin queues.

These sections describe transmit queue and port shaping configuration procedures:

- Transmit Queues and Port Shaping – Arad Platform Switches
- Transmit Queues and Port Shaping – Jericho Platform Switches
- Transmit Queues and Port Shaping – FM6000 Platform Switches
- Transmit Queues and Port Shaping – Petra Platform Switches
- Transmit Queues and Port Shaping – Trident and Tomahawk Platform
- Transmit Queues and Port Shaping – Trident II and Helix Platform Switches

10.1.1.4 Explicit Congestion Notification (ECN)

Explicit Congestion Notification (ECN) is an IP and TCP extension that facilitates end-to-end network congestion notification without dropping packets. ECN recognizes early congestion and sets flags that signal affected hosts. Trident and Tomahawk, Trident II, and Helix platform switches extend ECN support to non-TCP packets.

ECN usage requires that it is supported and enabled by both endpoints. Although only unicast flows are modified by ECN markers, the multicast, broadcast, and unmarked unicast flows can affect network congestion and influence the indication of unicast packet congestion.

10.1.1.4.1 ECN Conceptual Overview

The ECN field in the IP header (bits 6 and 7 in the IPv4 TOS or IPv6 traffic class octet) advertises ECN capabilities:

- ***00***: Router does not support ECN.
- ***10***: Router supports ECN.
- ***01***: Router supports ECN.
- ***11***: Congestion encountered.

Networks typically signal congestion by dropping packets. After an ECN-capable host negotiates ECN, it signals impending congestion by marking the IP header of packets encountering the congestion instead of dropping the packets. The recipient echoes the congestion indication back to the sender, which reduces its transmission rate as if it had detected a dropped packet.

Switches support ECN for unicast queues through Weighted Random Early Detection (WRED), an Active Queue Management (AQM) algorithm that extends Random Early Detection (RED) to define multiple thresholds for an individual queue. WRED determines congestion by comparing average queue size with queue thresholds. Average queue size depends on the previous average and current queue size:

- average queue size = (old_avg * (1-2^(-weight))) + (current_queue_size * 2^(-weight))
- where weight is the exponential weight factor used for averaging the queue size.
- Packets are marked based on WRED as follows:
- If average queue size is below the minimum threshold, packets are queued as in normal operation without ECN.
- If average queue size is greater than the maximum threshold, packets are marked for congestion.
- If average queue size is between minimum and maximum queue threshold, packets are either queued or marked. The proportion of packets that are marked increases linearly from 0% at the minimum threshold to 100% at the maximum threshold.

Treatment of packets marked as not ECN capable varies by platform.

These sections describe ECN configuration procedures:

- ECN Configuration – Arad Platform Switches
- ECN Configuration – Trident and Tomahawk Platform Switches

10.1.1.5 ACL Policing

ACL policing monitors the ingress data rates for a particular class of traffic and performs the action configured when the traffic exceeds the user configured value. Therefore, it allows the user to control ingress bandwidth based on packet classification. The incoming traffic is metered and marked by the policing, and based on the metering results the actions are performed.

ACL policing uses a token bucket shaping algorithm for packet transmission. Packets are eligible for transmission when token count is positive, and when token count is negative the next packet will have to wait until the token count turns positive again. The tokens are renewed at 96ns time interval (Tc). The tokens are collected in the policer bucket up to a max burst size of 16KB, and any traffic beyond this shape rate and burst size is buffered in the shared memory. The packets are dropped if there is a memory overflow.



Note: The policer bucket is refilled at a sweeper period of 0.333 millisecond. This is applicable for all the platforms.

For example, let us assume that shaping is not enabled, and the link is at 10 Gbps, that is 1.25 bytes/nsec. In such case a each refill cycle will add tokens worth 120 bytes. For a shape rate of 500 Mbps, each refill cycle will add 6 bytes. And for 64 byte worth of tokens we need around 11 refill cycles = 1us. A 64 byte packet

coming immediately after a jumbo frame will have to wait longer compared to a jumbo frame coming after 64 byte packet.

Token size depends on the interface speed, following the last example:

- For 10 Gbps, each refill cycle will add tokens worth 120 bytes.
- For 1 Gbps, each refill cycle will add tokens worth 12 bytes.

At lower shaping rates (less than 10 Mbps), granularity and rounding errors may alter the actual shaping rate by 20% from the specified rate, and the rounding errors are much less at higher speeds. For example, At 100 Mbps you will see 98.9 Mbps configured in hardware. User can use the **`show qos interfaces`** command to verify the interface speed.

The policing uses three types of traffic metering and coloring mechanisms.

- **Single Rate Two Color Marker**
- **Single Rate Three Color Marker**
- **Two Rate Three Color Marker**

Single Rate Two Color Marker

It meters the packet stream and marks packets based on committed burst size (bc) and excess burst size (be).

Single Rate Three Color Marker

It meters the packet stream and marks packets based on single rate committed information rate (cir), and committed burst size (bc) and excess burst size (be). The packets are marked in green if it does not exceed the set burst size, and marked in yellow if it does exceed the burst size but not the excess burst size, and marked red otherwise. The packets are marked in two color modes.

- **Color-blind Mode**: In color-blind mode the incoming packet color is ignored.
- **Color-aware Mode**: In color-aware mode it is assumed that incoming packet is colored by preceding entity. And, in color-aware mode, a packet never get better than it was. If the input color of the packet is green, it can be marked as green, yellow, or red. But if the input color is yellow, then it can be marked only yellow or red.

Two Rate Three Color Marker

It meters the packet stream and marks its packets based on two rates, peak information rate (pir) and committed information rate (cir), and associated burst sizes (bc and be).The packet is marked red if rate exceeds ‘pir’, and yellow if it exceeds ‘cir’ but not 'pir' and marked green if rate is lower than 'cir'. The two rate mode is configured by setting four parameters pir, cir, bc, and be.

The ACL policing is supported on platforms specified in the table below.

ACL Policing Support Matrix

Platform Supported	ACL Policing	ACL Policing on LAG Interface
Trident	Yes	Yes
Trident II	Yes	Yes
Trident+	Yes	Yes
FM6000	Yes	Yes
Arad	Yes	Only Per-Port
Jericho	Yes	Yes
Helix	Yes	Yes
XP	Yes	Yes
Trident 3	Yes	Yes
Tomahawk	Yes	Yes
Tomahawk 2	Yes	Yes
Tofino	No	No

10.1.1.5.1 Configuring ACL Policing

The policer is applied to the class inside the policy map. Policy maps can contain one or more policy map classes, each with different match criteria and policers. The following is the default behavior on conditions and available policing actions:

- Police command creates a per-interface policer. If you attach per-interface policers to multiple ingress ports, each one polices the matched traffic on each ingress port separately. Per interface statistics gathered for conformed/allowed traffic and exceeded/dropped traffic.
- When there is no policer configured within a class, all traffic is transmitted without any policing. If there are any actions configured, the configured actions are applied.
  - conform-action (green): transmit (default).
  - exceed-action (yellow): drop (default).
  - violate-action (red): drop (default).
- The policer bucket is refilled at a sweeper period of 0.333 milliseconds, and the tokens in the policer bucket are renewed at 96ns time interval (Tc). This is applicable for all the platforms.

Steps to Configure ACL Policing

These commands set the CIR, burst size, and creates a class and applies the policing to the policy map:

1. Create a policy map.
2. Create a class-map.
3. Apply the policer to the policy map created.

Examples

- These commands configure the ACL policing for a policy map.

```
switch#configure terminal
switch(config)#policy-map [type qos] policy-name
switch(config-pmap)#class { class-name }
switch(config-pmap-c)#[no] police cir cir [{bps|kbps|mbps}] bc
committed-burst-size [{bytes|kbytes|mbytes}]
```

- These commands configure ACL policing in single-rate, two-color mode.

```
switch(config)#class-map type qos match-any class1
switch(config-cmap-class1)#match ip access-group acl1
switch(config-cmap-class1)#exit

switch(config)#policy-map type quality-of-service policy1
switch(config-pmap)#class class1
switch(config-pmap-c)#police cir 512000 bc 96000
switch(config-pmap-c)#exit
switch(config-pmap)#
```

Displaying ACL Policing Information

Examples

- This command shows the contents of all policy maps on the switch.

```
switch(config)#show policy-map
Service-policy p

 Class-map: c (match-any)
    Match: ip access-group name a
       police rate 1000 mbps burst-size 100 bytes
 Class-map: class-default (match-any)
Service-policy p
 Class-map: c (match-any)
    Match: ip access-group name a
       police rate 1000 mbps burst-size 100 bytes
 Class-map: class-default (match-any)
```

- This command shows the interface-specific police counters for ***interface Ethernet 1***.

```
switch(config)#show policy-map interface Ethernet 1 input counters
Service-policy input: policy1
Hardware programming status: Successful

Class-map: class1 (match-any) Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes Conformed 4351 packets, 1857386 bytes
Conformed 2536 packets, 3384260 bytes

Class-map: class-default (match-any) matched packets: 0
```

- This command shows the counters associated with the policy map called ***p1***.

```
switch(config)#show policy-map type qos p1 input counters
Service-policy input: p1 Class-map: c1 (match-any)
Match: ip access-group name a1
Police cir 512000 bps bc 96000 bytes Interface: Ethernet1
Conformed 4351 packets, 1857386 bytes
Exceeded 2536 packets, 3384260 bytes
Interface: Ethernet2
Conformed 2351 packets, 957386 bytes
Exceeded 1536 packets, 1384260 bytes
Class-map: class-default (match-any)
```

```
Matched packets : 3229
```

- This command shows the QoS policy map for ***interface Ethernet 1***.

```
switch(config)#show policy-map interface Ethernet 1 input type qos
Interface: Ethernet 1 Service-policy input: policy1
Hardware programming status: Successful Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 9000 bytes
Class-map: class2 (match-any)
Match: ip access-group name acl2 set dscp 2
Class-map: class3 (match-any) Match: ip access-group name acl3
Police cir 1280000 bps bc 9000 bytes
Class-map: class-default (match-any)
```

10.1.1.6 Quality of Service (QoS) Profiles

QoS profiles are sets of QoS configuration instructions defined and applied at the interface level. A QoS profile serves the traffic better by reducing disorder in the running configuration. QoS profiles can modify all interface-level QoS configurations, and are supported on fabric, Ethernet, and port-channel interfaces. Control-plane policies cannot be applied using QoS profiles. Because configuration can be applied through QoS profiles or directly at the interface level, multiple configurations can be applied to the same interface. In such cases, QoS configurations with non-default values, whether configured through the CLI at the interface level or through a QoS profile, are given priority. In the case of multiple non-default values being configured, the interface-level CLI configuration is given priority.

Policy maps incorporating traffic resolution commands can also be applied by a QoS profile. If two policy maps are applied to the same interface (one through a QoS profile and another directly to the CLI).

Policy maps cannot be used on fabric interfaces. If a QoS profile which includes a policy map is applied to a fabric interface, a warning message will be displayed and the policy map will not be applied to the interface, but any additional supported configurations in the QoS profile will be applied. On SVIs and subinterfaces, QoS profiles are not supported, so policy maps must be applied directly through the CLI for these interfaces.

Note: For tx-queue configuration, conflicts between QoS profiles and configuration entered via the CLI are resolved at the tx-queue level and not at the tx-queue attribute level. If any non-default configuration has been entered for the tx-queue through the CLI, all tx-queue configuration included in the QoS profile is ignored.

10.1.1.6.1 IPv6 Flow Label Matches for QoS

Certain packets may include a flow label in their IPv6 headers when the source requests special handling by routers, such as for a media stream or other “real-time” service, among others. A flow consists of packets which share a single flow label, which is preserved throughout their passage from source to destination.

QoS policy map rules can match IPv6 traffic based on their flow labels. This requires a special TCAM profile (qos-match-ipv6-flow-label). These rules require either an exact match, using the “eq” operator, or both a label and a mask.

10.1.2 QoS Configuration: Platform-Independent Features

10.1.2.1 Creating QoS Profiles

QoS profiles are created by using the **qos profile** command. This also places the switch in QoS profile configuration mode, where the QoS parameters applied to interfaces are configured. To delete a QoS profile from the running configuration, use the no form of the command.

Example

This command creates a QoS profile named ***Test-Profile*** and places the switch in QoS profile configuration mode for the profile.

```
switch(config)# qos profile Test-Profile
switch(config-qos-profile-Test-Profile)#
```

10.1.2.2 Configuring QoS Profiles

The parameters that a QoS profile applies to interfaces are configured in QoS profile configuration mode by issuing the same QoS configuration commands that are available in interface configuration mode. **QoS profile** configuration mode is a group change mode, and changes made in the mode are not saved until the mode is exited. To abandon all changes made while in the mode, use the **abort** command.

Example

These commands enter QoS profile configuration mode for a QoS profile named ***Test Profile***, configure the CoS value and transmit queue, and save the changes to the profile.

```
switch(config)# qos profile Test-Profile
switch(config-qos-profile-Test-Profile)# qos cos 3
switch(config-qos-profile-Test-Profile)# priority-flow-control on
switch(config-qos-profile-Test-Profile)# exit
switch(config)#
```

These commands enter QoS profile configuration mode for a QoS profile named ***Latency***, configure the maximum latency value for VOQ tail-drop threshold, and save the changes to the profile. The latency value can be specified to a maximum of ***50*** ms. Both milliseconds and microseconds may be used.

```
switch(config)# qos profile Latency
switch(config-qos-profile-Latency)# tx-queue 3
switch(config-qos-profile-Latency)# latency maximum <1-50000> microseconds
switch(config-qos-profile-Latency)# latency maximum <1-50> milliseconds
switch(config-qos-profile-Latency)# exit
switch(config)#
```

10.1.2.3 Attaching Policy-Map to a QoS Profile

The qos profile command places the switch in QoS profile configuration mode. The profile applies the QoS configurations to Ethernet and Port-Channel, and even to the Fabric interfaces, if it exists. A profile specifies the policy-map and other QoS supported configurations. The policy-map is then attached to the QoS profile using **service-policy** command.

Profiles are created in QoS-profile configuration mode, then applied to an interface in interface configuration mode.

Examples

- This command places the switch in QoS profile configuration mode, the policy-map is then attached to the profile using **service-policy** command in this mode.

  ```
  switch(config)# qos profile TP
  switch(config-qos-profile-TP)#
  ```

- This command applies the policy-map to the QoS profile.

  ```
  switch(config-qos-profile-TP)# service-policy type qos input PM-1
  ```

10.1.2.4 Applying a QoS profile on an Interface

The **service-profile** command applies a QoS profile to the configuration mode interface.

Example

This command applies the QoS profile TP to ***interface ethernet 13***.

```
switch(config)# interface ethernet 13
switch(config-if-Et13)# service-profile TP
```

10.1.2.5 Displaying the QoS Profile Information

The **show qos profile** command displays information about the QoS profiles configured and their parameters. To display the attribute of a specific profile, add the name of the profile. To display a list of configured QoS profiles and the interfaces on which they are configured, add the **summary** keyword.

Examples

- This command displays the configured profiles and their configuration.

```
switch# show qos profile
qos profile p
 qos cos 1
no priority-flow-control pause watchdog
priority-flow-control priority 1 no-drop
priority-flow-control priority 2 no-drop
qos profile p2
qos cos 3
priority-flow-control priority 0 no-drop
```

- This command displays the contents of a specific profile.

```
switch# show qos profile p2
qos profile p2
qos cos 3
priority-flow-control priority 0 no-drop
```

- This command displays the interfaces on which each profile is applied.

```
switch# show qos profile summary
Qos Profile: p
Configured on: Et13,7
Fabric
Po12
Qos Profile: p2
Configured on: Et56
```

10.1.3 QoS Configuration: Arad Platform Switches

Implementing QoS on an Arad platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- CoS and DSCP Port Settings – Arad Platform Switche
- Traffic Class Derivations – Arad Platform Switches
- CoS Rewrite – Arad Platform Switches
- Transmit Queues and Port Shaping – Arad Platform Switches
- ECN Configuration – Arad Platform Switches
- ACL Policing – Arad Platform Switches

Note: QoS traffic policy is supported on Trident and Tomahawk, Trident II, FM6000, Arad, and Jericho.

10.1.3.1 CoS and DSCP Port Settings – Arad Platform Switches

Port Settings – Trust Mode and Traffic Class describes port trust and default port CoS and DSCP values.

Configuring Port Trust Settings

The **qos trust** command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is **CoS**. The port-trust default for routed ports is **DSCP**.

- **`qos trust cos`** specifies **CoS** as the port's port-trust mode.
- **`qos trust dscp`** specifies **DSCP** as the port's port-trust mode.
- **`no qos trust`** specifies **untrusted** as the port's port-trust mode.

The show qos interfaces trust command displays the trust mode of specified interfaces.

Example

These commands configure and display the following trust modes:

- ***Ethernet 3/5/1:*** **dscp**
- ***Ethernet 3/5/2:*** **untrusted**
- ***Ethernet 3/5/3:*** **cos**
- ***Ethernet 3/5/4:*** **default** as a switched port
- ***Ethernet 3/6/1:*** **default** as a routed port

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# qos trust dscp
switch(config-if-Et3/5/1)# interface ethernet 3/5/2
switch(config-if-Et3/5/2)# no qos trust
switch(config-if-Et3/5/2)# interface ethernet 3/5/3
switch(config-if-Et3/5/3)# qos trust cos
switch(config-if-Et3/5/3)# interface ethernet 3/5/4
switch(config-if-Et3/5/4)# switchport
switch(config-if-Et3/5/4)# default qos trust
switch(config-if-Et3/5/4)# interface ethernet 3/6/1
switch(config-if-Et3/6/1)# no switchport
switch(config-if-Et3/6/1)# default qos trust
switch(config-if-Et3/6/1) #show qos interface ethernet 3/5/1 - 3/6/1 trust
Port                                    Trust Mode
                                Operational           Configured
-------------------------------------------------------------------
Ethernet3/5/1                   DSCP                  DSCP
Ethernet3/5/2                   UNTRUSTED             UNTRUSTED
Ethernet3/5/3                   COS                   COS
Ethernet3/5/4                   COS                   DEFAULT
Ethernet3/6/1                   DSCP                  DEFAULT

switch(config-if-Et3/6/1)#
```

Configuring Default Port Settings

Default CoS and DSCP values are assigned to each Ethernet and port channel interface. These commands specify the configuration mode interface commands specify the port's default CoS and DSCP values.

- **qos cos** configures a port's default CoS value.
- **qos dscp** configures a port's default DSCP value.

Example

These commands configure default CoS (***4***) and DSCP (***44***) values on ***Ethernet interface 3/6/2***.

```
switch(config)# interface ethernet 3/6/2
switch(config-if-Et3/6/2)# qos cos 4
switch(config-if-Et3/6/2)# qos dscp 44
switch(config-if-Et3/6/2)# show active
interface Ethernet3/6/2
   qos cos 4
   qos dscp 44
switch(config-if-Et3/6/2)# show qos interfaces ethernet 3/6/2
Ethernet3/6/2:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44

switch(config-if-Et3/6/2)#
```

10.1.3.2 Traffic Class Derivations – Arad Platform Switches

Traffic Classes describes traffic classes.

Traffic Class Derivation Source

The following table displays the source for deriving a data stream's traffic class.

Table 6: Traffic Class Derivation Source: Arad Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

CoS and DSCP Port Settings – Arad Platform Switches describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The **qos map cos** command assigns a traffic class to a list of CoS values. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the chip upon which it is received.

Example

This command assigns the traffic class of ***5*** to the classes of service ***1***, ***3***, ***5***, and ***7***.

```
switch(config)# qos map cos 1 3 5 7 to traffic-class 5
switch(config)# show qos maps
   Number of Traffic Classes supported: 8


   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  5  2  5  4  5  6  5
```

```
switch(config)#
```

The following table displays the default CoS to Traffic Class map on Arad platform switches.

Table 7: Default CoS to Traffic Class Map: Arad Platform Switches

Inbound CoS	Untagged	0	1	2	3	4	5	6	7
Traffic Class	Derived: use default CoS as inbound	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The **qos map dscp** command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

This command assigns the traffic class of ***0*** to DSCP values of ***12***, ***24***, ***41***, and ***44-47***.

```
switch(config)# qos map dscp 12 24 41 44 45 46 47 to traffic-class 0
switch(config)# show qos maps
   Number of Traffic Classes supported: 8

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  0  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  0  5  5  0  0  0  0  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

switch(config)#
```

The following table displays the default DSCP to traffic class map on Arad platform switches.

Table 8: Default DSCP to Traffic Class Map: Arad Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

10.1.3.3 CoS Rewrite – Arad Platform Switches

Rewriting CoS and DSCP describes the CoS rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The **qos map traffic-class to cos** command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

This command assigns the CoS of two to traffic classes ***1***, ***3***, and ***5***.

```
switch(config)# qos map traffic-class 1 3 5 to cos 2
switch(config)# show qos map
   Number of Traffic Classes supported: 8

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     --------------------------
     cos:  1  2  2  2  4  2  6  7

switch(config)#
```

The following table displays the default Traffic Class to CoS rewrite value map on Arad platform switches.

Table 9: Default Traffic Class to CoS Rewrite Value Map: Arad Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

DSCP rewrite is always disabled on Arad platform switches.

10.1.3.4 Transmit Queues and Port Shaping – Arad Platform Switches

Transmit Queues and Port Shaping describes transmit queues and port shaping.

Arad platform switches provide 16 physical queues for each egress port: eight unicast and eight multicast queues. Data is scheduled to the physical queues based on transmit queue assignments.

Multicast queue capacity that remains after multicast traffic is serviced is available for unicast traffic of a corresponding priority. Similarly, unicast queue capacity that remains after unicast traffic is serviced is available for overflow multicast traffic. Under conditions of unicast and multicast congestion, egress traffic is evenly split between unicast and multicast traffic.

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class–transmit queue map for unicast and multicast traffic on all Ethernet and port channel interfaces. The **show qos maps** command displays the traffic class–transmit queue map.

The following table displays the default traffic class to transmit queue map on Arad platform switches.

Table 10: Default Traffic Class to Transmit Queue Map: Arad Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue	0	1	2	3	4	5	6	7

Transmit queue parameters are configured in tx-queue configuration command mode, which is entered from ***interface-ethernet*** configuration mode.

Mapping Traffic Classes to a Transmit Queue

The **qos map traffic-class to tx-queue** command assigns traffic classes to a transmit queue. Multiple commands complete the traffic class-transmit queue map. Traffic class ***7*** and transmit queue ***7*** are always mapped to each other. This association is not editable.

Example

These commands assign traffic classes of ***1***, ***3***, and ***5*** to ***transmit queue 1***, traffic classes ***2***, ***4***, and ***6*** to transmit queue ***2***, and traffic class ***0*** to transmit queue ***0***, then display the resultant map.

```
switch(config)# qos map traffic-class 1 3 5 to tx-queue 1
switch(config)# qos map traffic-class 2 4 6 to tx-queue 2
switch(config)# qos map traffic-class 0 to tx-queue 0
switch(config)# show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8

   Tc - tx-queue map:
     tc:         0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:   0  1  2  1  2  1  2  7

switch(config)#
```

Entering Tx-Queue Configuration Mode

The **tx-queue (Arad/Jericho)** command places the switch in tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue 7 is not configurable. The **show qos interfaces** displays the transmit queue configuration for a specified port.

Example

This command enters Tx-queue configuration mode for ***transmit queue 4*** of ***interface Ethernet 3/3/3***.

```
switch(config)# interface ethernet 3/3/3
switch(config-if-Et3/3/3)# tx-queue 4
switch(config-if-Et3/3/3-txq-4)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies its maximum outbound traffic bandwidth. A transmit queue's shape rate specifies the queue's maximum outbound bandwidth. Shape rate commands specify data rates in kbps.

- To configure a port's shape rate, enter **shape rate (Interface – Arad/Jericho)** from the port's interface configuration mode.
- To configure a transmit queue's shape rate, enter **shape rate (Tx-queue – Arad/Jericho)** from the queue's tx-queue configuration mode.

Examples

- This command configures a port shape rate of ***5*** Gbps on ***interface Ethernet 3/5/1***.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# shape rate 5000000
switch(config-if-Et3/5/1)# show qos interfaces ethernet 3/5/1
Ethernet3/5/1:

   Port shaping rate: 5000012 / 5000000 kbps

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D

switch(config-if-Et3/5/1)#
```

- These commands configure a shape rate of ***1*** Gbps on transmit queues ***3*** and ***4*** of ***interface Ethernet 3/4/1***.

```
switch(config)# interface ethernet 3/4/1
switch(config-if-Et3/4/1)# tx-queue 4
switch(config-if-Et3/4/1-txq-4)# shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-4)# tx-queue 3
switch(config-if-Et3/4/1-txq-3)# shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-3)# show qos interface ethernet 3/4/1
Ethernet3/4/1:

   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -         - / -     (  -  )    SP / SP    D
   6      - / -         - / -     (  -  )    SP / SP    D
   5      - / -         - / -     (  -  )    SP / SP    D
   4      - / -       999 / 1000 ( Mbps )    SP / SP    D
   3      - / -       999 / 1000 ( Mbps )    SP / SP    D
   2      - / -         - / -     (  -  )    SP / SP    D
   1      - / -         - / -     (  -  )    SP / SP    D
   0      - / -         - / -     (  -  )    SP / SP    D

switch(config-if-Et3/4/1-txq-3)#
```

Configuring Queue Priority

The **priority (Arad/Jericho)** command configures a transmit queue's priority type:

- The **`priority strict`** command configures the queue as a strict priority queue.
- The **`no priority`** command configures the queue as a round robin queue.

A queue's configuration as **round robin** also applies to all lower priority queues regardless of other configuration statements.

The **bandwidth percent (Arad/Jericho)** command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Examples

- These commands configure queues ***0*** through ***3*** (***interface Ethernet 3/5/1***) as round robin, then allocate bandwidth for three queues at ***30%*** and one queue at ***10%***.

  The **`no priority`** statement for queue *3* also configures queues ***0***, ***1***, and ***2*** as round robin queues. Removing this statement reverts the other queues to **`strict priority`** type unless ***running-config*** contains a **`no priority`** statement for one of these queues.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# tx-queue 3
switch(config-if-Et3/5/1-txq-3)# no priority
switch(config-if-Et3/5/1-txq-3)# bandwidth percent 10
switch(config-if-Et3/5/1-txq-3)# tx-queue 2
switch(config-if-Et3/5/1-txq-2)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-2)# tx-queue 1
switch(config-if-Et3/5/1-txq-1)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-1)# tx-queue 0
switch(config-if-Et3/5/1-txq-0)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-0)# show qos interfaces ethernet 3/5/1
```

```
Ethernet3/5/1:

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     10 / 10       - / -    (  -  )    RR / RR    D
   2     30 / 30       - / -    (  -  )    RR / SP    D
   1     30 / 30       - / -    (  -  )    RR / SP    D
   0     30 / 30       - / -    (  -  )    RR / SP    D

switch(config-if-Et3/5/1-txq-0)#
```

- Changing the bandwidth percentage for queue ***3*** to ***30*** changes the operational bandwidth of each queue to its configured bandwidth divided by ***120%*** (10%+20%+30%+60%).

```
switch(config-if-Et3/5/1-txq-0)# tx-queue 3
switch(config-if-Et3/5/1-txq-3)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-3)# show qos interfaces ethernet 3/5/1
Ethernet3/5/1:

   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     24 / 30       - / -    (  -  )    RR / RR    D
   2     24 / 30       - / -    (  -  )    RR / SP    D
   1     24 / 30       - / -    (  -  )    RR / SP    D
   0     24 / 30       - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured

switch(config-if-Et3/5/1-txq-3)#
```

10.1.3.5 ECN Configuration – Arad Platform Switches

Explicit Congestion Notification (ECN) describes Explicit Congestion Notification (ECN).

ECN is independently configurable on all egress queues of each Ethernet interface. ECN settings for Port-Channels are applied on each of the channel's member Ethernet interfaces. Average queue length is tracked for transmit queues. When it reaches maximum threshold, all subsequent packets are marked.

Although the switch does not limit the number of queues that can be configured for ECN, hardware table limitations restrict the number of queues that can simultaneously implement ECN.

The **random-detect ecn (Arad/Jericho)** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes.

Example

These commands enable ECN marking of unicast packets from ***unicast transmit queue 4*** of ***interface Ethernet 3/5/1***, setting thresholds at ***128*** kbytes and ***1280*** kbytes.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# tx-queue 4
```

```
switch(config-if-Et3/5/1-txq-4)# random-detect ecn minimum-threshold 128 kbytes
 maximum-threshold 1280 kbyte
switch(config-if-Et3/5/1-txq-4)# show active
interface Ethernet3/5/1
   tx-queue 4
      random-detect ecn minimum-threshold 128 kbytes maximum-threshold 1280
 kbytes
switch(config-if-Et3/5/1-txq-4)#
```

10.1.3.6 ACL Policing – Arad Platform Switches

ACL Policing describes ACL policing.

Implementing ACL policing consists of configuring the following:

- policy-map settings.
- class-name.
- committed information rate (CIR) the data speed committed to any given circuit regardless of the number of users.
- burst size the maximum burst size in bytes the network commits to moving under normal conditions.

The default unit for the metering rate CIR is bits per second; the default unit for the burst size is bytes.

The policer is applied to the class inside the policy map. Policy maps can contain one or more policy map classes, each with different match criteria and policer.

Default behavior and available policing actions are as follows:

- Policy map can be applied on multiple interfaces. Interfaces on the same chip will share the policer. (Applicable for Arad only.)
- If there is no policer configured within a class, all traffic is transmitted without any policing.
- If there are any actions configured, the configured actions are applied:
  - Conform-action (green): transmit (default).
  - Violate-action (red): drop (default).

Example

These commands configure ACL policing in single-rate, two-color mode.

```
switch(config)# class-map type qos match-any class1
switch(config-cmap-class1)# match ip access-group acl1
switch(config-cmap-class1)# exit
switch(config)# policy-map type quality-of-service policy1
switch(config-policy1)# class class1
switch(config-policy1-class1)# police cir 512000 bc 96000
switch(config-policy1-class1)# exit
switch(config-policy1)# exit
switch(config)#
```

Displaying ACL Policing Information

Examples

- This command shows the contents of all policy maps on the switch.

```
switch(config)# show policy-map
Service-policy policy1

Class-map: class1 (match-any)
```

```
Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes

Class-map: class-default (match-any)
switch(config)#
```

- This command shows the interface-specific police counters for ***interface Ethernet 1***.

```
switch(config)# show policy-map interface Ethernet 1 input counters
Service-policy input: policy1
Hardware programming status: Successful

Class-map: class1 (match-any) Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes Conformed 4351 packets, 1857386 bytes
Conformed 2536 packets, 3384260 bytes

Class-map: class-default (match-any) matched packets: 0

switch(config)#
```

- This command shows the counters associated with the policy map called ***p1***.

```
switch(config)# show policy-map type qos p1 input counters
Service-policy input: p1 Class-map: c1 (match-any)
Match: ip access-group name a1
Police cir 512000 bps bc 96000 bytes Interface: Ethernet1
Conformed 4351 packets, 1857386 bytes
Exceeded 2536 packets, 3384260 bytes
Interface: Ethernet2
Conformed 2351 packets, 957386 bytes
Exceeded 1536 packets, 1384260 bytes
Class-map: class-default (match-any)
Matched packets : 3229

switch(config)#
```

- This command shows the QoS policy map for ***interface Ethernet 1***.

```
switch(config)# show policy-map interface Ethernet 1 input type qos
Interface: Ethernet 1 Service-policy input: policy1
Hardware programming status: Successful Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 9000 bytes
Class-map: class2 (match-any)
Match: ip access-group name acl2 set dscp 2
Class-map: class3 (match-any) Match: ip access-group name acl3
Police cir 1280000 bps bc 9000 bytes
Class-map: class-default (match-any)

switch(config)#
```

10.1.4 QoS Configuration: Jericho Platform Switches

Implementing QoS on an Jericho platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- CoS and DSCP Port Settings – Jericho Platform Switches
- Traffic Class Derivations – Jericho Platform Switches
- CoS Rewrite – Jericho Platform Switches
- Transmit Queues and Port Shaping – Jericho Platform Switches
- ACL Policing – Jericho Platform Switches

Note: QoS traffic policy is supported on Trident and Tomahawk, Trident II, FM6000, Arad. and Jericho.

10.1.4.1 CoS and DSCP Port Settings – Jericho Platform Switches

Port Settings – Trust Mode and Traffic Class describes port trust and default port CoS and DSCP values.

Configuring Port Trust Settings

The qos trust command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is **CoS**. The port-trust default for routed ports is **DSCP**.

- **`qos trust cos`** specifies **CoS** as the port's port-trust mode.
- **`qos trust dscp`** specifies **DSCP** as the port's port-trust mode.
- **`no qos trust`** specifies **untrusted** as the port's port-trust mode.

The show qos interfaces trust command displays the trust mode of specified interfaces.

Example

These commands configure and display the following trust modes:

- ***Ethernet 3/5/2:*** **untrusted**.
- ***Ethernet 3/5/3:*** **cos**.
- ***Ethernet 3/5/4:*** **default** as a switched port.
- ***Ethernet 3/6/1:*** **default** as a routed port.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# qos trust dscp
switch(config-if-Et3/5/1)# interface ethernet 3/5/2
switch(config-if-Et3/5/2)# no qos trust
switch(config-if-Et3/5/2)# interface ethernet 3/5/3
switch(config-if-Et3/5/3)# qos trust cos
switch(config-if-Et3/5/3)# interface ethernet 3/5/4
switch(config-if-Et3/5/4)# switchport
switch(config-if-Et3/5/4)# default qos trust
switch(config-if-Et3/5/4)# interface ethernet 3/6/1
switch(config-if-Et3/6/1)# no switchport
switch(config-if-Et3/6/1)# default qos trust
switch(config-if-Et3/6/1)# show qos interface ethernet 3/5/1 - 3/6/1 trust
Port                                Trust Mode
                                Operational           Configured
-------------------------------------------------------------------
Ethernet3/5/1                   DSCP                  DSCP
Ethernet3/5/2                   UNTRUSTED             UNTRUSTED
Ethernet3/5/3                   COS                   COS
Ethernet3/5/4                   COS                   DEFAULT
Ethernet3/6/1                   DSCP                  DEFAULT

switch(config-if-Et3/6/1)#
```

Configuring Default Port Settings

Default **CoS** and **DSCP** values are assigned to each Ethernet and port channel interface. These commands specify the configuration mode interface commands specify the port's default **CoS** and **DSCP** values.

- qos cos configures a port's default CoS value.
- qos dscp configures a port's default DSCP value.

Example

These commands configure default **CoS** (***4***) and **DSCP** (***44***) values on ***interface Ethernet 3/6/2***.

```
switch(config)# interface ethernet 3/6/2
switch(config-if-Et3/6/2)# qos cos 4
switch(config-if-Et3/6/2)# qos dscp 44
switch(config-if-Et3/6/2)# show active
interface Ethernet3/6/2
   qos cos 4
   qos dscp 44
switch(config-if-Et3/6/2)# show qos interfaces ethernet 3/6/2
Ethernet3/6/2:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44

switch(config-if-Et3/6/2)#
```

10.1.4.2 Traffic Class Derivations – Jericho Platform Switches

Traffic Classes describes traffic classes.

Traffic Class Derivation Source

The following table displays the source for deriving a data stream's traffic class on Jericho platform switches.

Table 11: Traffic Class Derivation Source: Jericho Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

CoS and DSCP Port Settings – Arad Platform Switches describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The qos map cos command assigns a traffic class to a list of CoS values. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the chip upon which it is received.

ExampleThis command assigns the traffic class of ***5*** to the classes of service ***1***, ***3***, ***5***, and ***7***.

```
switch(config)# qos map cos 1 3 5 7 to traffic-class 5
switch(config)# show qos maps
   Number of Traffic Classes supported: 8

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ---------------------------
     tc:   1  5  2  5  4  5  6  5

switch(config)#
```

The following table displays the default CoS to Traffic Class map on Jericho platform switches.

Table 12: Default CoS to Traffic Class Map: Jericho Platform Switches

Inbound CoS	Untagged	0	1	2	3	4	5	6	7
Traffic Class	Derived: use default CoS as inbound	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The qos map dscp command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

ExampleThis command assigns the traffic class of 0 to DSCP values of ***12***, ***24***, ***41***, and ***44-47***.

```
switch(config)# qos map dscp 12 24 41 44 45 46 47 to traffic-class 0
switch(config)# show qos maps
   Number of Traffic Classes supported: 8


   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  0  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  0  5  5  0  0  0  0  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7


switch(config)#
```

The following table displays the default DSCP to traffic class map on Jericho platform switches.

Table 13: Default DSCP to Traffic Class Map: Jericho Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

10.1.4.3 CoS Rewrite – Jericho Platform Switches

Rewriting CoS and DSCP describes the CoS rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The qos map traffic-class to cos command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

ExampleThis command assigns the CoS of two to traffic classes ***1***, ***3***, and ***5***.

```
switch(config)# qos map traffic-class 1 3 5 to cos 2
switch(config)# show qos map
   Number of Traffic Classes supported: 8

   Tc-cos map:
```

```
     tc:   0  1  2  3  4  5  6  7
     ---------------------------
     cos:  1  2  2  2  4  2  6  7

switch(config)#
```

The following table displays the default Traffic Class to CoS rewrite value map on Jericho platform switches.

Table 14: Default Traffic Class to CoS Rewrite Value Map: Jericho Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

DSCP rewrite is always disabled on Jericho platform switches.

10.1.4.4 Transmit Queues and Port Shaping – Jericho Platform Switches

Transmit Queues and Port Shaping describes transmit queues and port shaping.

Jericho platform switches provide 16 physical queues for each egress port: eight unicast and eight multicast queues. Data is scheduled to the physical queues based on transmit queue assignments.

Multicast queue capacity that remains after multicast traffic is serviced is available for unicast traffic of a corresponding priority. Similarly, unicast queue capacity that remains after unicast traffic is serviced is available for overflow multicast traffic. Under conditions of unicast and multicast congestion, egress traffic is evenly split between unicast and multicast traffic.

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class–transmit queue map for unicast and multicast traffic on all Ethernet and port channel interfaces. The show qos maps command displays the traffic class–transmit queue map.

The following table displays the default traffic class to transmit queue map on Jericho platform switches.

Table 15: Default Traffic Class to Transmit Queue Map: Jericho Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue	0	1	2	3	4	5	6	7

Transmit queue parameters are configured in tx-queue configuration command mode, which is entered from interface-ethernet configuration mode.

Mapping Traffic Classes to a Transmit Queue

The qos map traffic-class to tx-queue command assigns traffic classes to a transmit queue. Multiple commands complete the traffic class-transmit queue map. Traffic class 7 and transmit queue 7 are always mapped to each other. This association is not editable.

ExampleThese commands assign traffic classes of ***1***, ***3***, and ***5*** to ***transmit queue 1***, traffic classes ***2***, ***4***, and ***6*** to ***transmit queue 2***, and ***traffic class 0*** to ***transmit queue 0***, then display the resultant map.

```
switch(config)# qos map traffic-class 1 3 5 to tx-queue 1
switch(config)# qos map traffic-class 2 4 6 to tx-queue 2
switch(config)# qos map traffic-class 0 to tx-queue 0
switch(config)# show qos maps
```

```
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8


   Tc - tx-queue map:
     tc:         0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:   0  1  2  1  2  1  2  7

switch(config)#
```

Entering Tx-Queue Configuration Mode

The tx-queue (Arad/Jericho) command places the switch in tx-queue configuration mode to configure a transmit queue on the configuration mode interface. ***Tx-queue 7*** is not configurable. The show qos interfaces displays the transmit queue configuration for a specified port.

ExampleThis command enters Tx-queue configuration mode for ***transmit queue 4*** of ***interface ethernet 3/3/3***.

```
switch(config)# interface ethernet 3/3/3
switch(config-if-Et3/3/3)# tx-queue 4
switch(config-if-Et3/3/3-txq-4)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies its maximum outbound traffic bandwidth. A transmit queue's shape rate specifies the queue's maximum outbound bandwidth. Shape rate commands specify data rates in kbps.

- To configure a port's shape rate, enter shape rate (Interface – Arad/Jericho) from the port's interface configuration mode.
- To configure a transmit queue's shape rate, enter shape rate (Tx-queue – Arad/Jericho) from the queue's tx-queue configuration mode.

Examples

- This command configures a port shape rate of ***5*** Gbps on ***interface ethernet 3/5/1***.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# shape rate 5000000
switch(config-if-Et3/5/1)# show qos interfaces ethernet 3/5/1
Ethernet3/5/1:

   Port shaping rate: 5000012 / 5000000 kbps

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D

switch(config-if-Et3/5/1)#
```

- These commands configure a shape rate of ***1*** Gbps on transmit queues ***3*** and ***4*** on ***interface ethernet 3/4/1***.

```
switch(config)# interface ethernet 3/4/1
switch(config-if-Et3/4/1)# tx-queue 4
switch(config-if-Et3/4/1-txq-4)# shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-4)# tx-queue 3
switch(config-if-Et3/4/1-txq-3)# shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-3)# show qos interface ethernet 3/4/1
```

```
Ethernet3/4/1:

   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   -----------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -      999 / 1000 ( Mbps )    SP / SP    D
   3      - / -      999 / 1000 ( Mbps )    SP / SP    D
   2      - / -        - / -    (  -  )    SP / SP    D
   1      - / -        - / -    (  -  )    SP / SP    D
   0      - / -        - / -    (  -  )    SP / SP    D

switch(config-if-Et3/4/1-txq-3)#
```

Configuring Queue Priority

The priority (Arad/Jericho) command configures a transmit queue's priority type:

- The **`priority strict`** command configures the queue as a strict priority queue.
- The **`no priority`** command configures the queue as a round robin queue.

A queue's configuration as **round robin** also applies to all lower priority queues regardless of other configuration statements.

The bandwidth percent (Arad/Jericho) command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Examples

- These commands configure queues ***0*** through ***3*** (***interface ethernet 3/5/1***) as round robin, then allocate bandwidth for three queues at ***30%*** and one queue at ***10%***.

  The **`no priority`** statement for queue ***3*** also configures queues ***0***, ***1***, and ***2*** as round robin queues. Removing this statement reverts the other queues to **strict priority** type unless ***running-config*** contains a **`no priority`** statement for one of these queues.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# tx-queue 3
switch(config-if-Et3/5/1-txq-3)# no priority
switch(config-if-Et3/5/1-txq-3)# bandwidth percent 10
switch(config-if-Et3/5/1-txq-3)# tx-queue 2
switch(config-if-Et3/5/1-txq-2)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-2)# tx-queue 1
switch(config-if-Et3/5/1-txq-1)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-1)# tx-queue 0
switch(config-if-Et3/5/1-txq-0)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-0)# show qos interfaces ethernet 3/5/1
Ethernet3/5/1:

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   -----------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     10 / 10       - / -    (  -  )    RR / RR    D
```

```
   2     30 / 30       - / -    (  -  )    RR / SP    D
   1     30 / 30       - / -    (  -  )    RR / SP    D
   0     30 / 30       - / -    (  -  )    RR / SP    D

switch(config-if-Et3/5/1-txq-0)#
```

- Changing the bandwidth percentage for queue ***3*** to ***30*** changes the operational bandwidth of each queue to its configured bandwidth divided by ***120%*** (10%+20%+30%+60%).

```
switch(config-if-Et3/5/1-txq-0)# tx-queue 3
switch(config-if-Et3/5/1-txq-3)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-3)# show qos interfaces ethernet 3/5/1
Ethernet3/5/1:

   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     24 / 30       - / -    (  -  )    RR / RR    D
   2     24 / 30       - / -    (  -  )    RR / SP    D
   1     24 / 30       - / -    (  -  )    RR / SP    D
   0     24 / 30       - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured

switch(config-if-Et3/5/1-txq-3)#
```

10.1.4.5 ACL Policing – Jericho Platform Switches

ACL Policing describes ACL policing.

Implementing ACL policing consists of configuring the following:

- **policy-map settings**.
- **class-name**.
- **committed information rate (CIR)** the data speed committed to any given circuit regardless of the number of users.
- **burst size** the maximum burst size in bytes the network commits to moving under normal conditions.

The default unit for the metering rate CIR is bits per second; the default unit for the burst size is bytes.

The policer is applied to the class inside the policy map. Policy maps can contain one or more policy map classes, each with different match criteria and policer.

Default behavior and available policing actions are as follows:

- Policy map can be applied on multiple interfaces. Interfaces on the same chip will share the policer. (Applicable for Arad and Jericho only.)
- If there is no policer configured within a class, all traffic is transmitted without any policing.
- If there are any actions configured, the configured actions are applied:
  - Conform-action (green): transmit (default).
  - Violate-action (red): drop (default).

Example

These commands configure ACL policing in single-rate, two-color mode.

```
switch(config)# class-map type qos match-any class1
switch(config-cmap-class1)# match ip access-group acl1
switch(config-cmap-class1)# exit
switch(config)# policy-map type quality-of-service policy1
switch(config-policy1)# class class1
switch(config-policy1-class1)# police cir 512000 bc 96000
switch(config-policy1-class1)# exit
switch(config-policy1)# exit
switch(config)#
```

Displaying ACL Policing Information

Examples

- This command shows the contents of all policy maps on the switch.

```
switch(config)# show policy-map
Service-policy policy1

Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes

Class-map: class-default (match-any)
switch(config)#
```

- This command shows the interface-specific police counters for ***interface ethernet 1***.

```
switch(config)# show policy-map interface Ethernet 1 input counters
Service-policy input: policy1
Hardware programming status: Successful

Class-map: class1 (match-any) Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes Conformed 4351 packets, 1857386 bytes
Conformed 2536 packets, 3384260 bytes

Class-map: class-default (match-any) matched packets: 0

switch(config)#
```

- This command shows the counters associated with the policy map called ***p1***.

```
switch(config)# show policy-map type qos p1 input counters
Service-policy input: p1 Class-map: c1 (match-any)
Match: ip access-group name a1
Police cir 512000 bps bc 96000 bytes Interface: Ethernet1
Conformed 4351 packets, 1857386 bytes
Exceeded 2536 packets, 3384260 bytes
Interface: Ethernet2
Conformed 2351 packets, 957386 bytes
Exceeded 1536 packets, 1384260 bytes
Class-map: class-default (match-any)
Matched packets : 3229

switch(config)#
```

- This command shows the QoS policy map for ***interface ethernet 1***.

```
switch(config)# show policy-map interface Ethernet 1 input type qos
Interface: Ethernet 1 Service-policy input: policy1
Hardware programming status: Successful Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 9000 bytes
Class-map: class2 (match-any)
Match: ip access-group name acl2 set dscp 2
Class-map: class3 (match-any) Match: ip access-group name acl3
Police cir 1280000 bps bc 9000 bytes
Class-map: class-default (match-any)

switch(config)#
```

10.1.5 QoS Configuration: FM6000 Platform Switches

Implementing QoS on an FM6000 platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- CoS and DSCP Port Settings – FM6000 Platform Switches
- Traffic Class Derivations – FM6000 Platform Switches
- CoS and DSCP Rewrite – FM6000 Platform Switches
- Transmit Queues and Port Shaping – FM6000 Platform Switches

10.1.5.1 CoS and DSCP Port Settings – FM6000 Platform Switches

Port Settings – Trust Mode and Traffic Class describes port trust and default port CoS and DSCP values.

Configuring Port Trust Settings

The qos trust command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is **cos**. The port-trust default for routed ports is **dscp**.

- **`qos trust cos`** specifies **cos** as the port's port-trust mode.
- **`qos trust dscp`** specifies dscp as the port's port-trust mode.
- **`no qos trust`** specifies untrusted as the port's port-trust mode.

The show qos interfaces trust command displays the trust mode of specified interfaces.

Example

These commands configure and display the following trust modes:

- ***Ethernet 15:*** **dscp**.
- ***Ethernet 16:*** **untrusted**.
- ***Ethernet 17:*** **cos**.
- ***Ethernet 18:*** **default** as a switched port.
- ***Ethernet 19:*** **default** as a routed port.

```
switch(config)# interface ethernet 15
switch(config-if-Et15)# qos trust dscp
switch(config-if-Et15)# interface ethernet 16
switch(config-if-Et16)# no qos trust
switch(config-if-Et16)# interface ethernet 17
switch(config-if-Et17)# qos trust cos
switch(config-if-Et17)# interface ethernet 18
switch(config-if-Et18)# switchport
switch(config-if-Et18)# default qos trust
```

```
switch(config-if-Et19)# interface ethernet 19
switch(config-if-Et19)# no switchport
switch(config-if-Et19)# default qos trust
switch(config-if-Et19)# show qos interface ethernet 15 - 19 trust
Port                                Trust Mode
                                Operational           Configured
-------------------------------------------------------------------
Ethernet15                      DSCP                  DSCP
Ethernet16                      UNTRUSTED             UNTRUSTED
Ethernet17                      COS                   COS
Ethernet18                      COS                   DEFAULT
Ethernet19                      DSCP                  DEFAULT

switch(config-if-Et19)#
```

Configuring Default Port Settings

Default CoS and DSCP settings are assigned to individual port channel and Ethernet interfaces. These configuration mode interface commands specify the port's default CoS and DSCP values.

- qos cos configures a port's default CoS value.
- qos dscp configures a port's default DSCP value.

Example

These commands configure default CoS (***4***) and DSCP (***44***) settings on ***interface ethernet 19***.

```
switch(config)# interface ethernet 19
switch(config-if-Et19)# qos cos 4
switch(config-if-Et19)# qos dscp 44
switch(config-if-Et19)# show active
interface Ethernet19
   qos cos 4
   qos dscp 44
switch(config-if-Et19)# show qos interfaces ethernet 19
Ethernet19:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44

switch(config-if-Et19)#
```

10.1.5.2 Traffic Class Derivations – FM6000 Platform Switches

Traffic Classes describes traffic classes.

Traffic Class Derivation Source

The following table displays the source for deriving a data stream's traffic class.

Table 16: Traffic Class Derivation Source: FM6000 Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

CoS and DSCP Port Settings – FM6000 Platform Switches describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The qos map cos command assigns a traffic class to a list of CoS settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

This command assigns the traffic class of ***5*** to the classes of service ***1***, ***3***, ***5***, and ***7***.

```
switch(config)# qos map cos 1 3 5 7 to traffic-class 5
switch(config)# show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8


   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ---------------------------
     tc:   1  5  2  5  4  5  6  5


switch(config)#
```

The following table displays the default CoS to Traffic Class map on FM6000 platform switches.

Table 17: Default CoS to Traffic Class Map: FM6000 Platform Switches

Inbound CoS	Untagged	0	1	2	3	4	5	6	7
Traffic Class	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The qos map dscp command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

This command assigns the traffic class of three to the DSCP values of ***12***, ***13***, ***25***, and ***37***.

```
switch(config)# qos map dscp 12 13 25 37 to traffic-class 3
switch(config)# show qos map
   Number of Traffic Classes supported: 8
```

```
   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  3  3  0  0  2  2  2  2
      2 :     2  2  2  2  3  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  3  4  4
      4 :     5  5  5  5  5  5  5  5  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

switch(config)#
```

The following displays the default DSCP to Traffic Class map on FM6000 platform switches.

Table 18: Default DSCP to Traffic Class Map: FM6000 Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

10.1.5.3 CoS and DSCP Rewrite – FM6000 Platform Switches

Rewriting CoS and DSCP describes the CoS and DSCP rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The qos map traffic-class to cos command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

This command assigns the CoS rewrite value of two to traffic classes ***1***, ***3***, and ***5***.

```
switch(config)# qos map traffic-class 1 3 5 to cos 2
switch(config)# show qos map
   Number of Traffic Classes supported: 8

   Tc - tx-queue map:
     tc:         0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:   0  1  2  3  4  5  6  7

switch(config)#
```

The following table displays the default traffic class–CoS rewrite map on FM6000 platform switches.

Table 19: Default Traffic Class to CoS Rewrite Map: FM6000 Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

The DSCP rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-DSCP rewrite map. The qos map traffic-class to dscp command assigns a DSCP rewrite value to a list of traffic classes. Multiple commands create the complete traffic class-DSCP rewrite map.

Example

This command assigns the DSCP rewrite value of ***37*** to traffic classes ***2***, ***4***, and ***6***.

```
switch(config)# qos map traffic-class 2 4 6 to dscp 37
switch(config)# show qos map
   Number of Traffic Classes supported: 8

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ----------------------------
     dscp:  8  0 37 24 37 40 37 56

switch(config)#
```

The following table displays the default traffic class–DSCP rewrite map on on FM6000 platform switches.

Table 20: Default Traffic Class to DSCP Rewrite Map: FM6000 Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
DSCP Rewrite Value	8	0	16	24	32	40	48	56

10.1.5.4 Transmit Queues and Port Shaping – FM6000 Platform Switches

Transmit Queues and Port Shaping describes transmit queues and port shaping.

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class-transmit queue map for all Ethernet and port channel interfaces and is used for unicast and multicast traffic. The show qos maps command displays the traffic class to transmit queue map.

The following table displays the default traffic class to transmit queue map on FM6000 platform switches.

Table 21: Default Traffic Class to Transmit Queue Map: FM6000 Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue	0	1	2	3	4	5	6	7

Mapping Traffic Classes to a Transmit Queue

The qos map traffic-class to tx-queue command assigns traffic classes to a transmit queue. Multiple commands create the complete map.

Example

These commands assign traffic classes of ***1***, ***3***, and ***5*** to ***transmit queue 1***, traffic classes ***2***, ***4***, and ***6*** to ***transmit queue 2***, and ***traffic class 0*** to ***transmit queue 0***, then display the resultant map.

```
switch(config)# qos map traffic-class 1 3 5 to tx-queue 1
switch(config)# qos map traffic-class 2 4 6 to tx-queue 2
switch(config)# qos map traffic-class 0 to tx-queue 0
switch(config)# show qos maps
```

```
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8


   Tc - tx-queue map:
     tc:         0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:   0  1  2  1  2  1  2  7

switch(config)#
```

Entering TX-Queue Configuration Mode

Transmit queues are configurable on Ethernet ports and port channels. Queue parameters are configured in tx-queue configuration command mode, which is entered from interface ethernet configuration mode. The tx-queue (FM6000) command places the switch in tx-queue configuration mode. The show qos interfaces displays the transmit queue configuration for a specified port.

Example

This command enters tx-queue configuration mode for ***transmit queue 3*** of ***interface Ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# tx-queue 3
switch(config-if-Et5-txq-3)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies its maximum outbound traffic bandwidth. A transmit queue's shape rate specifies the queue's maximum outbound bandwidth. Shape rate commands specify data rates in kbps.

Note: Enabling port shaping on an FM6000 interface disables queue shaping internally. Disabling port shaping restores queue shaping as specified in ***running-config***.

- To configure a port's shape rate, enter shape rate (Interface – FM6000) from the port's ***interface*** configuration mode.
- To configure a transmit queue's shape rate, enter shape rate (Tx-queue – FM6000) from the queue's ***tx-queue*** configuration mode.

Example

These commands configure a shape rate of ***5*** Gbs on ***interface Ethernet 3***, then configure the shape rate for the following transmit queues:

- - transmit queues ***0***, ***1***, and ***2***: ***500*** Mbps.
  - transmit queues ***3***, ***4***, and ***5***: ***400*** Mbps.

```
switch(config)# interface ethernet 3
switch(config-if-Et3)# shape rate 5000000
switch(config-if-Et3)# tx-queue 0
switch(config-if-Et3-txq-0)# shape rate 500000
switch(config-if-Et3-txq-0)# tx-queue 1
switch(config-if-Et3-txq-1)# shape rate 500000
switch(config-if-Et3-txq-1)# tx-queue 3
switch(config-if-Et3-txq-3)# shape rate 400000
switch(config-if-Et3-txq-3)# tx-queue 4
switch(config-if-Et3-txq-4)# shape rate 400000
switch(config-if-Et3-txq-4)# tx-queue 5
switch(config-if-Et3-txq-5)# shape rate 400000
switch(config-if-Et3-txq-5)# exit
switch(config-if-Et3)# show qos interface ethernet 3
Ethernet3:
```

```
   Port shaping rate: 5000000Kbps

   Tx-Queue    Bandwidth     Shape Rate      Priority
               (percent)        (Kbps)
   ---------------------------------------------
          7          N/A     disabled         strict
          6          N/A     disabled         strict
          5          N/A       400000         strict
          4          N/A       400000         strict
          3          N/A       400000         strict
          2          N/A     disabled         strict
          1          N/A       500000         strict
          0          N/A       500000         strict

switch(config-if-Et3)#
```

Configuring Queue Priority

Queue priority rank is denoted by the queue number; transmit queues with higher numbers have higher priority. The priority (FM6000) command configures a transmit queue's priority type:

- **`priority strict`** configures the queue as a strict priority queue.
- **`no priority`** configures the queue as a round robin queue.

A queue's configuration as **round robin** also applies to all lower priority queues regardless of other configuration statements.

The bandwidth percent (FM6000) command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Examples

- These commands configure ***transmit queue 3*** (on ***interface Ethernet 19***) as a round robin queue, then allocates ***10%***, ***20%***, ***30%***, and ***40%*** bandwidth to queues ***0*** through ***3***.

  The **`no priority`** statement for ***queue 3*** also configures queues ***0***, ***1***, and ***2*** as round robin queues. Removing this statement reverts the other queues to **`strict priority`** type unless ***running-config*** contains a **`no priority`** statement for one of these queues.

```
switch(config)# interface ethernet 19
switch(config-if-Et19)# tx-queue 3
switch(config-if-Et19-txq-3)# no priority
switch(config-if-Et19-txq-3)# bandwidth percent 40
switch(config-if-Et19-txq-3)# tx-queue 2
switch(config-if-Et19-txq-2)# bandwidth percent 30
switch(config-if-Et19-txq-2)# tx-queue 1
switch(config-if-Et19-txq-1)# bandwidth percent 20
switch(config-if-Et19-txq-1)# tx-queue 0
switch(config-if-Et19-txq-0)# bandwidth percent 10
switch(config-if-Et19-txq-0)# show qos interface ethernet 19
Ethernet19:
   Port shaping rate: disabled

   Tx-Queue    Bandwidth     Shape Rate      Priority
               (percent)        (Kbps)
   ---------------------------------------------
          7          N/A     disabled         strict
          6          N/A     disabled         strict
          5          N/A     disabled         strict
          4          N/A     disabled         strict
```

```
           3          40     disabled   round-robin
           2          30     disabled   round-robin
           1          20     disabled   round-robin
           0          10     disabled   round-robin

switch(config-if-Et19-txq-0)#
```

- Changing the bandwidth percentage for queue ***3*** to ***60*** changes the operational bandwidth of each queue to its configured bandwidth divided by ***120%*** (10%+20%+30%+60%).

```
switch(config-if-Et19-txq-0) #tx-queue 3
switch(config-if-Et19-txq-3)# bandwidth percent 60
switch(config-if-Et19-txq-3)# show qos interface ethernet 19
Ethernet19:
   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          49     disabled   round-robin
          2          24     disabled   round-robin
          1          16     disabled   round-robin
          0           8     disabled   round-robin

switch(config-if-Et19-txq-3)#
```

10.1.6 QoS Configuration: Petra Platform Switches

Implementing QoS on a Petra platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- CoS and DSCP Port Settings – Petra Platform Switches
- Traffic Class Derivations – Petra Platform Switches
- CoS Rewrite – Petra Platform Switches
- Transmit Queues and Port Shaping – Petra Platform Switches

10.1.6.1 CoS and DSCP Port Settings – Petra Platform Switches

Port Settings – Trust Mode and Traffic Class describes port trust and default port CoS and DSCP values.

Configuring Port Trust Settings

The qos trust command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is **cos**. The port-trust default for routed ports is **dscp**.

- **`qos trust cos`** specifies **cos** as the port's port-trust mode.
- **`qos trust dscp`** specifies dscp as the port's port-trust mode.
- **`no qos trust`** specifies untrusted as the port's port-trust mode.

The show qos interfaces trust command displays the trust mode of specified interfaces.

Example

These commands configure and display the following trust modes:

- ***Ethernet 3/25:*** **dscp**.

- ***Ethernet 3/26:*** **untrusted**.
- ***Ethernet 3/27:*** **cos**.
- ***Ethernet 3/28:*** **default** as a switched port.
- ***Ethernet 3/29:*** **default** as a routed port.

```
switch(config)# interface ethernet 3/25
switch(config-if-Et3/25)# qos trust dscp
switch(config-if-Et3/25)# interface ethernet 3/26
switch(config-if-Et3/26)# no qos trust
switch(config-if-Et3/26)# interface ethernet 3/27
switch(config-if-Et3/27)# qos trust cos
switch(config-if-Et3/27)# interface ethernet 3/28
switch(config-if-Et3/28)# switchport
switch(config-if-Et3/28)# default qos trust
switch(config-if-Et3/28)# interface ethernet 3/29
switch(config-if-Et3/29)# no switchport
switch(config-if-Et3/29)# default qos trust
switch(config-if-Et3/29)# show qos interface ethernet 3/25 - 3/29 trust
Port                                    Trust Mode
                                Operational           Configured
-------------------------------------------------------------------
Ethernet3/25                    DSCP                  DSCP
Ethernet3/26                    UNTRUSTED             UNTRUSTED
Ethernet3/27                    COS                   COS
Ethernet3/28                    COS                   DEFAULT
Ethernet3/29                    DSCP                  DEFAULT

switch(config-if-Et3/29)#
```

Configuring Default Port Settings

Port channel and Ethernet interfaces are not assigned default CoS or DSCP settings.

10.1.6.2 Traffic Class Derivations – Petra Platform Switches

Traffic Classes describes traffic classes.

Traffic Class Derivation Source

The following table displays the source for deriving a data stream's default traffic class.

Table 22: Traffic Class Derivation Source: Petra Platform Switches

	Untrusted	**CoS Trusted**	**DSCP Trusted**
Untagged Non-IP	Default TC (chip)	Default TC (chip)	Default TC (chip)
Untagged IP	Default TC (chip)	Default TC (chip)	DSCP (packet)
Tagged Non-I	Default TC (chip)	CoS (packet)	Default TC (chip)
Tagged IP	Default TC (chip)	CoS (packet)	DSCP (packet)

Configuring Default Traffic Class

Petra platform switches assign a default traffic class to the set of Ethernet interfaces controlled by individual PetraA chips. Default traffic class values are configurable for each PetraA chip, not individual interfaces.

The platform petraA traffic-class command specifies the default traffic class used by all ports controlled by a specified chip. The show platform petraA traffic-class command displays traffic class assignments.

Examples

- This command configures the default traffic class to five for the ports ***32-39*** on ***linecard 3*** (7500 Series).

```
switch(config)# platform petraA petra3/4 traffic-class 5
switch(config)# show platform petraA module 3 traffic-class
Petra3/0 traffic-class: 1
Petra3/1 traffic-class: 1
Petra3/2 traffic-class: 1
Petra3/3 traffic-class: 1
Petra3/4 traffic-class: 5
Petra3/5 traffic-class: 1
switch(config)#
```

- This command configures the default traffic class to three for all ports on ***linecard 6*** (7500 Series).

```
switch(config)# platform petraA module 6 traffic-class 6
switch(config)# show platform petraA module 6 traffic-class
Petra6/0 traffic-class: 6
Petra6/1 traffic-class: 6
Petra6/2 traffic-class: 6
Petra6/3 traffic-class: 6
Petra6/4 traffic-class: 6
Petra6/5 traffic-class: 6
switch(config)#
```

Mapping CoS to Traffic Class

The qos map cos command assigns a traffic class to a list of CoS settings. Multiple commands create a complete CoS–traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

This command assigns ***traffic class 4*** to the classes of service ***1***, ***3***, ***5***, and ***7***.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 4
switch(config)#show qos maps
   Number of Traffic Classes supported: 8

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  4  2  4  4  4  6  4

switch(config)#
```

The following table displays the default CoS to traffic class map on Petra platform switches.

Table 23: Default CoS to Traffic Class Map: Petra Platform Switches

Inbound CoS	untagged	0	1	2	3	4	5	6	7
Traffic Class	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The qos map dscp command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

This command assigns ***traffic class 3*** to the DSCP values of ***12***, ***13***, ***25***, and ***37***.

```
switch(config)# qos map dscp 12 13 14 25 48 to traffic-class 3
switch(config)# show qos maps
   Number of Traffic Classes supported: 8

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  3  3  3  0  2  2  2  2
      2 :     2  2  2  2  3  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  5  5  5  5  5  5  5  3  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

switch(config)#
```

The following table displays the default DSCP to Traffic Class map on Petra platform switches.

Table 24: Default DSCP to Traffic Class Map: Petra Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

10.1.6.3 CoS Rewrite – Petra Platform Switches

Rewriting CoS and DSCP describes the CoS rewrite function.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The qos map traffic-class to cos command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class-CoS rewrite map.

Example

This command assigns the CoS of two to traffic classes ***1***, ***3***, and ***5***.

```
switch(config)# qos map traffic-class 1 3 5 to cos 2
switch(config)# show qos map
   Number of Traffic Classes supported: 8

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ---------------------------
     cos:  1  2  2  2  4  2  6  7

switch(config)#
```

The following table displays the default Traffic Class to CoS rewrite value map on Petra platform switches.

Table 25: Default Traffic Class to CoS Rewrite Value Map: Petra Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

DSCP rewrite is always disabled on Petra platform switches.

10.1.6.4 Transmit Queues and Port Shaping – Petra Platform Switches

Transmit Queues and Port Shaping describes transmit queues and port shaping.

Petra platform switches provide four physical queues for each egress port: Unicast High, Unicast Low, Multicast High, and Multicast Low. Data is scheduled for the high or low queue based on its priority as defined by its transmit queue assignment (unicast traffic) or traffic class (multicast traffic), as shown in the table below. A Petra transmit queue is a data structure that defines scheduling of unicast traffic among phyical egress queues.

Table 26: Traffic Distribution to Egress Port Queues

	High Priority Queue	**Low Priority Queue**
Unicast Traffic	Transmit Queues 5 – 7	Transmit Queues 0 – 4
Multicast Traffic	Traffic Classes 5 – 7	Traffic Classes 0 – 4

Multicast queue capacity that is available after multicast traffic is serviced is used for unicast traffic of a corresponding priority. Similarly, unicast queue capacity that is available after unicast traffic is serviced is used for overflow multicast traffic. Under conditions of unicast and multicast congestion, egress traffic is evenly split between unicast and multicast traffic.

Unicast Transmit Queues and Port Shaping describes unicast transmit queues and shaping. Multicast Egress Scheduling describes multicast priority and traffic classes.

10.1.6.4.1 Unicast Transmit Queues and Port Shaping

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class–transmit queue map for unicast traffic on all Ethernet interfaces. The show qos maps command displays the traffic class–transmit queue map. The following table displays the default traffic class to transmit queue map on Petra platform switches.

Table 27: Default Traffic Class to Transmit Queue Map: Petra Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue	0	1	2	3	4	5	6	7

Transmit queue parameters are configured in tx-queue configuration command mode.

Mapping Traffic Classes to a Transmit Queue

The qos map traffic-class to tx-queue command assigns traffic classes to a transmit queue. Multiple commands complete the traffic class-transmit queue map. Traffic class ***7*** and transmit queue ***7*** are always mapped to each other. This association is not editable.

Example

These commands assign traffic classes of ***1***, ***3***, and ***5*** to ***transmit queue 1***, traffic classes ***2***, ***4***, and ***6*** to ***transmit queue 2***, and ***traffic class 0*** to ***transmit queue 0***, then display the resultant map.

```
switch(config)# qos map traffic-class 1 3 5 to tx-queue 1
switch(config)# qos map traffic-class 2 4 6 to tx-queue 2
switch(config)# qos map traffic-class 0 to tx-queue 0
switch(config)# show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8

   Tc - tx-queue map:
     tc:         0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:   0  1  2  1  2  1  2  7

switch(config)#
```

Entering Tx-Queue Configuration Mode

The tx-queue (Petra) command places the switch in tx-queue configuration mode to configure a transmit queue on the configuration mode interface. ***Tx-queue 7*** not configurable. The show qos interfaces displays the transmit queue configuration for a specified port.

Example

This command enters tx-queue configuration mode for ***transmit queue 3*** of ***interface ethernet 3/28***.

```
switch(config)# interface ethernet 3/28
switch(config-if-Et3/28)# tx-queue 3
switch(config-if-Et3/28-txq-3)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies its maximum outbound traffic bandwidth. A transmit queue's shape rate specifies the queue's maximum outbound bandwidth. Shape rate commands specify data rates in kbps.

- To configure a port's shape rate, enter shape rate (Interface – Petra) from the port's ***interface*** configuration mode.
- To configure a transmit queue's shape rate, enter shape rate (Tx-queue – Petra) from the queue's ***tx-queue*** configuration mode.

Example

These commands configure a shape rate of ***5*** Gbs on ***interface Ethernet 3***, then configure the shape rate for the following transmit queues:

- transmit queues ***0***, ***1***, and ***2***: ***500*** Mbps
- transmit queues ***3***, ***4***, and ***5***: ***400*** Mbps

```
switch(config)# interface ethernet 3/28
switch(config-if-Et3/28)# shape rate 5000000
switch(config-if-Et3/28)# tx-queue 0
switch(config-if-Et3/28-txq-0)# shape rate 500000
switch(config-if-Et3/28-txq-0)# tx-queue 1
switch(config-if-Et3/28-txq-1)# shape rate 500000
switch(config-if-Et3/28-txq-1)# tx-queue 2
switch(config-if-Et3/28-txq-2)# shape rate 500000
switch(config-if-Et3/28-txq-5)# tx-queue 3
switch(config-if-Et3/28-txq-3)# shape rate 400000
```

```
switch(config-if-Et3/28-txq-3)# tx-queue 4
switch(config-if-Et3/28-txq-4)# shape rate 400000
switch(config-if-Et3/28-txq-4)# tx-queue 5
switch(config-if-Et3/28-txq-5)# shape rate 400000
switch(config-if-Et3/28-txq-5)# show qos interface ethernet 3/28
Ethernet3/28:

   Port shaping rate: 5000000Kbps

   Tx-Queue    Bandwidth     Shape Rate      Priority
               (percent)        (Kbps)
   ------------------------------------------------
          7          N/A     disabled         strict
          6          N/A     disabled         strict
          5          N/A       400000         strict
          4          N/A       400000         strict
          3          N/A       400000         strict
          2          N/A       500000         strict
          1          N/A       500000         strict
          0          N/A       500000         strict

switch(config-if-Et3/28-txq-5)#
```

Configuring Queue Priority

The priority (Petra) command configures a transmit queue's priority type:

- The **`priority strict`** command configures the queue as a strict priority queue.
- The **`no priority`** command configures the queue as a round robin queue.

A queue's configuration as **round robin** also applies to all lower priority queues regardless of other configuration statements.

The bandwidth percent (Petra) command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Examples

- These commands configure ***transmit queue 3*** (on ***interface Ethernet 3/28***) as a round robin queue, then allocates ***10%***, ***20%***, ***30%***, and ***40%*** bandwidth to queues ***0*** through ***3***.

  The **`no priority`** statement for ***queue 3*** also configures queues ***0***, ***1***, and ***2*** as round robin queues. Removing this statement reverts the other queues to **`strict priority`** type unless ***running-config*** contains a **`no priority`** statement for one of these queues.

```
switch(config-if-Et3/28)# tx-queue 3
switch(config-if-Et3/28-txq-3)# no priority
switch(config-if-Et3/28-txq-3)# bandwidth percent 40
switch(config-if-Et3/28-txq-3)# tx-queue 2
switch(config-if-Et3/28-txq-2)# bandwidth percent 30
switch(config-if-Et3/28-txq-2)# tx-queue 1
switch(config-if-Et3/28-txq-1)# bandwidth percent 20
switch(config-if-Et3/28-txq-1)# tx-queue 0
switch(config-if-Et3/28-txq-0)# bandwidth percent 10
switch(config-if-Et3/28-txq-0)# show qos interface ethernet 3/28
Ethernet3/28:

   Port shaping rate: 5000000Kbps

   Tx-Queue    Bandwidth     Shape Rate      Priority
               (percent)        (Kbps)
```

```
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A       400000        strict
          4         N/A       400000        strict
          3          40       400000   round-robin
          2          30       500000   round-robin
          1          20       500000   round-robin
          0          10       500000   round-robin

switch(config-if-Et3/28-txq-0)#
```

- Changing the bandwidth percentage for queue ***3*** to ***60*** changes the operational bandwidth of each queue to its configured bandwidth divided by ***120%*** (10%+20%+30%+60%).

```
switch(config-if-Et3/28-txq-0)# tx-queue 3
switch(config-if-Et3/28-txq-3)# bandwidth percent 60
switch(config-if-Et3/28-txq-3)# show qos interface ethernet 3/28
Ethernet3/28:

   Port shaping rate: 5000000Kbps

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A       400000        strict
          4         N/A       400000        strict
          3          49       400000   round-robin
          2          24       500000   round-robin
          1          16       500000   round-robin
          0           8       500000   round-robin
switch(config-if-Et3/28-txq-3)#
```

10.1.6.4.2 Multicast Egress Scheduling

Multiclass traffic is not affected by traffic class assignment or port shaping statements. Multicast traffic is assigned to port egress queues based on traffic class and uses strict priority to schedule egress between the high and low queues.

10.1.7 QoS Configuration: Trident and Tomahawk Platform Switches

Implementing QoS on a Trident and Tomahawk platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- CoS and DSCP Port Settings – Trident and Tomahawk Platform Switches
- Traffic Class Derivations – Trident and Tomahawk Platform Switches
- CoS and DSCP Rewrite – Trident and Tomahawk Platform Switches
- Transmit Queues and Port Shaping – Trident and Tomahawk Platform Switches
- ECN Configuration – Trident and Tomahawk Platform Switches

10.1.7.1 CoS and DSCP Port Settings – Trident and Tomahawk Platform Switches

Configuring Port Trust Settings

The qos trust command configures the QoS port trust mode for the configuration mode interface. Trust-enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is **CoS**. The port-trust default for routed ports is **DSCP**.

- **`qos trust cos`** specifies **CoS** as the port's trust mode.
- **`qos trust dscp`** specifies **DSCP** as the port's trust mode.
- **`no qos trust`** specifies **untrusted** as the port's trust mode.

The show qos interfaces trust command displays the trust mode of specified interfaces.

Example

- These commands configure and display the following trust modes:
  - ***Ethernet 15***: **dscp**
  - ***Ethernet 16***: **untrusted**
  - ***Ethernet 17***: **cos**
  - ***Ethernet 18***: **default** as a switched port
  - ***interface ethernet 19***: **default** as a routed port

```
switch(config)# interface ethernet 15
switch(config-if-Et15)# qos trust dscp
switch(config-if-Et15)# interface ethernet 16
switch(config-if-Et16)# no qos trust
switch(config-if-Et16)# interface ethernet 17
switch(config-if-Et17)# qos trust cos
switch(config-if-Et17)# interface ethernet 18
switch(config-if-Et18)# switchport
switch(config-if-Et18)# default qos trust
switch(config-if-Et18)# interface ethernet 19
switch(config-if-Et19)# no switchport
switch(config-if-Et19)# default qos trust
switch(config-if-Et19)# show qos interface ethernet 15 - 19 trust
Port                                Trust Mode
                                Operational           Configured
------------------------------------------------------------------
Ethernet15                      DSCP                  DSCP
Ethernet16                      UNTRUSTED             UNTRUSTED
Ethernet17                      COS                   COS
Ethernet18                      COS                   DEFAULT
Ethernet19                      DSCP                  DEFAULT

switch(config-if-Et19)#
```

Configuring Default Port Settings

Default CoS and DSCP settings are assigned to individual port channel and Ethernet interfaces. These configuration mode interface commands specify the port's default CoS and DSCP values.

- qos cos configures a port's default CoS value.
- qos dscp configures a port's default DSCP value.

Example

These commands configure default CoS (***4***) and DSCP (***44***) values on ***interface ethernet 7***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# qos cos 4
switch(config-if-Et7)# qos dscp 44
switch(config-if-Et7)# show active
interface Ethernet7
   qos cos 4
   qos dscp 44
switch(config-if-Et7)# show qos interfaces ethernet 7
Ethernet7:
   Trust Mode: COS
```

```
   Default COS: 4
   Default DSCP: 44

switch(config-if-Et7)#
```

10.1.7.2 Traffic Class Derivations – Trident and Tomahawk Platform Switches

Traffic Classes describes traffic classes.

Traffic Class Derivation Source

The following table displays the source for deriving a data stream's traffic class.

Table 28: Traffic Class Derivation Source: Trident and Tomahawk Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

CoS and DSCP Port Settings – Trident and Tomahawk Platform Switches describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The qos map cos command assigns a traffic class to a list of CoS settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

This command assigns the ***traffic class 5*** to the classes of service ***1***, ***3***, ***5***, and ***7***.

```
switch(config)# qos map cos 1 3 5 7 to traffic-class 5
switch(config)# show qos maps
   Number of Traffic Classes supported: 8

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ---------------------------
     tc:   1  5  2  5  4  5  6  5

switch(config)#
```

The following table displays the default CoS–traffic class map on Trident and Tomahawk platform switches.

Table 29: Default CoS to Traffic Class Map: Trident II Platform Switches

Inboun CoS	0	1	2	3	4	5	6	7
Traffic Class	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The qos map dscp command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

This command assigns the ***traffic class 0*** to DSCP values of ***12***, ***24***, ***41***, and ***44-47***.

```
switch(config)# qos map dscp 12 24 41 44 45 46 47 to traffic-class 0
switch(config)# show qos maps
   Number of Traffic Classes supported: 8

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  0  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  0  5  5  0  0  0  0  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

switch(config)#
```

The following table displays the default DSCP–traffic class map on Trident and Tomahawk platform switches.

Table 30: Default DSCP to Traffic Class Map: Trident and Tomahawk Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

10.1.7.3 CoS and DSCP Rewrite – Trident and Tomahawk Platform Switches

Rewriting CoS and DSCP describes the CoS and DSCP rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The qos map traffic-class to cos command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

This command assigns the ***CoS 2*** to traffic classes ***1***, ***3***, and ***5***.

```
switch(config)# qos map traffic-class 1 3 5 to cos 2
switch(config)# show qos map
   Number of Traffic Classes supported: 8

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ---------------------------
     cos:  1  2  2  2  4  2  6  7

switch(config)#
```

The following table displays the default Traffic Class to CoS rewrite value map on Trident and Tomahawk platform switches.

Table 31: Default Traffic Class to CoS Rewrite Value Map: Trident and Tomahawk Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

The DSCP rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-DSCP rewrite map. The qos map traffic-class to dscp command assigns a DSCP rewrite value to a list of traffic classes. Multiple commands create the complete traffic class-DSCP rewrite map.

Example

This command assigns the DSCP value of ***29*** to traffic classes ***2***, ***4***, and ***6***.

```
switch(config)#qos map traffic-class 2 4 6 to dscp 29
switch(config)#show qos map
   Number of Traffic Classes supported: 8

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ----------------------------
     dscp:  8  0 29 24 29 40 29 56

switch(config)#
```

The following displays the default traffic class to DSCP rewrite map on Trident and Tomahawk platform switches.

Table 32: Traffic Class to DSCP Rewrite Value Map: Trident and Tomahawk Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
DSCP	8	0	16	24	32	40	48	56

10.1.7.4 Transmit Queues and Port Shaping – Trident and Tomahawk Platform Switches

Transmit Queues and Port Shaping describes transmit queues and port shaping.

Trident and Tomahawk platform switches define ***12*** transmit queues: eight unicast (UC0 – UC7) and four multicast (MC0 – MC03). The traffic class–transmit queue maps are configured globally and apply to all Ethernet interfaces. The show qos maps command displays the traffic class–transmit queue maps.

The following table displays the default traffic class–transmit queue maps.

Table 33: Default Traffic Class to Transmit Queue Map: Trident and Tomahawk Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Unicast Transmit Queue	0	1	2	3	4	5	6	7
Multicast Transmit Queue	0	0	1	1	2	2	3	3

Mapping Traffic Classes to a Transmit Queue

These commands assign traffic classes to a transmit queue:

- qos map traffic-class to uc-tx-queue associates a unicast queue to a traffic class set.
- qos map traffic-class to mc-tx-queue associates a multicast queue to a traffic class set.

Multiple commands create the complete maps.

Example

These commands assign the following on ***interface ethernet 7***:

- traffic classes ***1***, ***3***, and ***5*** to ***unicast queue 1***
- traffic classes ***2***, ***4***, and ***6*** to ***unicast queue 5***
- traffic classes ***1***, ***2***, and ***3*** to ***multicast queue 1***
- traffic classes ***4***, ***5***, and ***6*** to ***multicast queue 3***
- ***traffic class 0*** to ***unicast queue 0*** and ***multicast queue 0***

```
switch(config)# default interface ethernet 7
switch(config)# qos map traffic-class 1 3 5 to uc-tx-queue 1
switch(config)# qos map traffic-class 2 4 6 to uc-tx-queue 5
switch(config)# qos map traffic-class 1 2 3 to mc-tx-queue 1
switch(config)# qos map traffic-class 4 5 6 to mc-tx-queue 3
switch(config)# qos map traffic-class 0 to uc-tx-queue 0
switch(config)# qos map traffic-class 0 to mc-tx-queue 0
switch(config)# show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 12

   Tc - uc-tx-queue map:
     tc:           0  1  2  3  4  5  6  7
     ---------------------------------
     uc-tx-queue:  0  1  5  1  5  1  5  7

   Tc - mc-tx-queue map:
     tc:           0  1  2  3  4  5  6  7
     ---------------------------------
     mc-tx-queue:  0  1  1  1  3  3  3  3

switch(config)#
```

Entering a Transmit Queue Configuration Mode

Transmit queues are configurable on individual Ethernet ports. Parameters for individual transmit queues are configured in one of two transmit queue configuration modes. Transmit queue modes are accessed from an interface-ethernet configuration mode.

- uc-tx-queue places the switch in ***uc-tx-queue*** mode to configure a unicast transmit queue.
- mc-tx-queue places the switch in ***mc-tx-queue*** mode to configure a multicast transmit queue.

The show qos interfaces displays the transmit queue configuration for a specified port.

Examples

- This command enters the mode that configures ***unicast transmit queue 3*** of ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# uc-tx-queue 3
switch(config-if-Et5-uc-txq-3)#
```

- This command enters the mode to configure ***multicast transmit queue 3*** of ***interface ethernet 5***.

```
switch(config-if-Et5)# mc-tx-queue 2
switch(config-if-Et5-mc-txq-2)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies the port's maximum outbound traffic bandwidth. A shape rate can also be configured for all transmit queues on each port. All shape rate commands use kbps to specify data rates.

- To configure a port's shape rate, enter shape rate (Interface – Trident and Tomahawk) from the port's ***interface*** configuration mode.
- To configure a transmit queue's shape rate, enter shape rate (Tx-queue – Trident and Tomahawk) from the queue's ***tx-queue*** configuration mode.

Example

These commands configure a shape rate of ***5*** Gbs on ***interface Ethernet 7***, then configure the shape rate for the following transmit queues:

- unicast transmit queues ***0*** and ***1***: ***500*** Mbps.
- unicast transmit queues ***3*** and ***4***: ***400*** Mbps.
- multicast transmit queues ***0*** and *2*: ***300*** Mbps.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# shape rate 5000000
switch(config-if-Et7)# uc-tx-queue 0
switch(config-if-Et7-uc-txq-0)# shape rate 500000
switch(config-if-Et7-uc-txq-0)# uc-tx-queue 1
switch(config-if-Et7-uc-txq-1)# shape rate 500000
switch(config-if-Et7-uc-txq-1)# uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)# shape rate 400000
switch(config-if-Et7-uc-txq-3)# uc-tx-queue 5
switch(config-if-Et7-uc-txq-5)# shape rate 400000
switch(config-if-Et7-uc-txq-5)# mc-tx-queue 0
switch(config-if-Et7-mc-txq-0)# shape rate 300000
switch(config-if-Et7-mc-txq-0)# mc-tx-queue 2
switch(config-if-Et7-mc-txq-2)# shape rate 300000
switch(config-if-Et7-mc-txq-2)# exit
switch(config-if-Et7)# show qos interface ethernet 7
Ethernet7:

   Port shaping rate: 5000000Kbps

   Tx-Queue    Bandwidth     Shape Rate      Priority    Priority Group
               (percent)        (Kbps)
   ------------------------------------------------------------------
        UC7          N/A     disabled         strict                 1
        UC6          N/A     disabled         strict                 1
        MC3          N/A     disabled         strict                 1
        UC5          N/A       400000         strict                 0
        UC4          N/A     disabled         strict                 0
        MC2          N/A       300000         strict                 0
        UC3          N/A       400000         strict                 0
        UC2          N/A     disabled         strict                 0
        MC1          N/A     disabled         strict                 0
        UC1          N/A       500000         strict                 0
        UC0          N/A       500000         strict                 0
        MC0          N/A       300000         strict                 0

switch(config-if-Et7)#
```

Configuring Queue Priority

Trident and Tomahawk platform switch queues are categorized into two priority groups. Priority group 1 queues have priority over priority 0 queues. The following lists display the priority group queues in order from higher priority to lower priority.

- Priority Group 1: UC7, UC6, MC3
- Priority Group 0: UC5, UC4, MC2, UC3, UC2, MC1, UC1, UC0, MC0

The priority (Trident and Tomahawk) command configures a transmit queue's priority type:

- The **`priority strict`** command configures the queue as a strict priority queue.
- The **`no priority`** command configures the queue as a round robin queue.

A queue's configuration as **round robin** also applies to all lower priority queues regardless of other configuration statements.

The bandwidth percent (Trident and Tomahawk) command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Priority Group 1 queues (UC7, UC6, MC3) are not configurable as round robin queues. The **`bandwidth percent`** command is not available for these queues.

Examples

- These commands configure ***unicast transmit queue 3*** as a round robin queue, then allocates ***5%***, ***15%***, ***25%***, ***35%***, ***8%***, and ***12%*** bandwidth to unicast transmit queues ***0*** through ***3*** and multicast transmit queues ***0*** and ***1***, respectively.

  The **`no priority`** statement for ***queue 3*** also configures priority for all lower priority queues. Removing the statement reverts the other queues to **`strict priority`** type unless ***running-config*** contains a **`no priority`** statement for one of these queues.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)# no priority
switch(config-if-Et7-uc-txq-3)# bandwidth percent 5
switch(config-if-Et7-uc-txq-3)# uc-tx-queue 2
switch(config-if-Et7-uc-txq-2)# bandwidth percent 15
switch(config-if-Et7-uc-txq-2)# uc-tx-queue 1
switch(config-if-Et7-uc-txq-1)# bandwidth percent 25
switch(config-if-Et7-uc-txq-1)# uc-tx-queue 0
switch(config-if-Et7-uc-txq-0)# bandwidth percent 35
switch(config-if-Et7-uc-txq-0)# mc-tx-queue 1
switch(config-if-Et7-mc-txq-1)# bandwidth percent 12
switch(config-if-Et7-mc-txq-1)# mc-tx-queue 0
switch(config-if-Et7-mc-txq-0)# bandwidth percent 8
switch(config-if-Et7-mc-txq-0)# show qos interface ethernet 7
Ethernet7:

   Port shaping rate: disabled

   Tx-Queue    Bandwidth     Shape Rate      Priority    Priority Group
               (percent)        (Kbps)
   ------------------------------------------------------------------
        UC7          N/A     disabled        strict                 1
        UC6          N/A     disabled        strict                 1
        MC3          N/A     disabled        strict                 1
        UC5          N/A     disabled        strict                 0
        UC4          N/A     disabled        strict                 0
        MC2          N/A     disabled        strict                 0
        UC3            5     disabled   round-robin                 0
```

```
        UC2           15     disabled   round-robin                0
        MC1           12     disabled   round-robin                0
        UC1           25     disabled   round-robin                0
        UC0           35     disabled   round-robin                0
        MC0            8     disabled   round-robin                0

switch(config-if-Et7-mc-txq-0)#
```

- Changing the bandwidth percentage for unicast queue ***3*** to ***30*** changes the operational bandwidth of each queue to its configured bandwidth divided by ***125%*** (8%+12%+30%+15%+25%+35%).

```
switch(config-if-Et7-uc-txq-0)# uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)# bandwidth percent 30
switch(config-if-Et7-uc-txq-3)# show qos interface ethernet 7
Ethernet7:

   Port shaping rate: disabled

   Tx-Queue    Bandwidth     Shape Rate      Priority   Priority Group
               (percent)        (Kbps)
   ------------------------------------------------------------------
        UC7          N/A     disabled         strict                1
        UC6          N/A     disabled         strict                1
        MC3          N/A     disabled         strict                1
        UC5          N/A     disabled         strict                0
        UC4          N/A     disabled         strict                0
        MC2          N/A     disabled         strict                0
        UC3           24     disabled    round-robin                0
        UC2           12     disabled    round-robin                0
        MC1            9     disabled    round-robin                0
        UC1           20     disabled    round-robin                0
        UC0           28     disabled    round-robin                0
        MC0            6     disabled    round-robin                0

switch(config-if-Et7-uc-txq-3)#
```

10.1.7.5 ECN Configuration – Trident and Tomahawk Platform Switches

ECN is independently configurable on all egress queues of each Ethernet interface. ECN settings for Port-Channels are applied on each of the channel's member Ethernet interfaces. ECN is also globally configurable to mark packets from the shared pool used for dynamically allocating memory to the queues. Multicast packets contribute to the globally shared pool and can contribute to global level congestion that result in ECN marking of unicast packets queued after the multicast packets.

Average queue length is tracked for transmit queues and the global pool independently. When either entity reaches its maximum threshold, all subsequent packets are marked.

Although the switch does not limit the number of queues that can be configured for ECN, hardware table limitations restrict the number of queues (including the global shared pool) that can simultaneously implement ECN.

The qos random-detect ecn global-buffer (Trident and Tomahawk) command enables ECN marking for globally shared packet memory and specifies minimum and maximum queue threshold sizes.

Examples

- This command enables ECN marking of unicast packets from the global data pool and sets the minimum and maximum thresholds at ***20*** and ***500*** segments.

```
switch(config)# qos random-detect ecn global-buffer minimum-threshold 20
 segments
maximum-threshold 500 segments
```

```
switch(config)#
```

- This command disables ECN marking of unicast packets from the global data pool.

```
switch(config)# no qos random-detect ecn global-buffer
switch(config)#
```

The random-detect ecn (Trident and Tomahawk) command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes.

- These commands enable ECN marking of unicast packets from ***transmit queue 4*** of ***interface Ethernet 15***, setting thresholds at ***10*** and ***100*** segments.

```
switch(config)# interface ethernet 15
switch(config-if-Et15)# uc-tx-queue 4
switch(config-if-Et15-uc-txq-4)# random-detect ecn minimum-threshold 10
 segments
maximum-threshold 100 segments
switch(config-if-Et15-uc-txq-4)# show active
interface Ethernet15
   uc-tx-queue 4
      random-detect ecn minimum-threshold 10 segments maximum-threshold 100
segments
switch(config-if-Et15-uc-txq-4)# exit
switch(config-if-Et15)#
```

- This command disables ECN marking of unicast packets from ***transmit queue 4*** of ***interface Ethernet 15***.

```
switch(config-if-Et15-uc-txq-4)# no random-detect ecn
switch(config-if-Et15-uc-txq-4)# show active
interface Ethernet15
switch(config-if-Et15-uc-txq-4)# exit
switch(config-if-Et15)#
```

10.1.8 QoS Configuration: Trident II and Helix Platform Switches

Implementing QoS on a Trident platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- CoS and DSCP Port Settings – Trident II and Helix Platform Switches
- Traffic Class Derivations – Trident II and Helix Platform Switches
- CoS and DSCP Rewrite – Trident II and Helix Platform Switches
- Transmit Queues and Port Shaping – Trident II and Helix Platform Switches
- Ingress Policing on LAG
- Fabric QoS -- – Trident II Platform Switches

10.1.8.1 CoS and DSCP Port Settings – Trident II and Helix Platform Switches

Configuring Port Trust Settings

The qos trust command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is ***cos***. The port-trust default for routed ports is ***dscp***.

- **`qos trust cos`** specifies ***cos*** as the port's port-trust mode.
- **`qos trust dscp`** specifies ***dscp*** as the port's port-trust mode.
- **`no qos trust`** specifies ***untrusted*** as the port's port-trust mode.

The show qos interfaces trust command displays the trust mode of specified interfaces.

Example

These commands configure and display the following trust modes:

- ***Ethernet 7/1***: **dscp**.
- ***Ethernet 7/2***: **untrusted**.
- ***Ethernet 7/3***: **cos**.
- ***Ethernet 7/4***: **default** as a switched port.
- ***Ethernet 8/1***: **default** as a routed port.
-

```
switch(config)# interface ethernet 7/1
switch(config-if-Et7/1)# qos trust dscp
switch(config-if-Et7/1)# interface ethernet 7/2
switch(config-if-Et7/2)# no qos trust
switch(config-if-Et7/2)# interface ethernet 7/3
switch(config-if-Et7/3)# qos trust cos
switch(config-if-Et7/3)# interface ethernet 7/4
switch(config-if-Et7/4)# switchport
switch(config-if-Et7/4)# default qos trust
switch(config-if-Et7/4)# interface ethernet 8/1
switch(config-if-Et8/1)# no switchport
switch(config-if-Et8/1)# default qos trust
switch(config-if-Et8/1)# show qos interface ethernet 7/1 - 8/1 trust

Port                                    Trust Mode
                                Operational           Configured
-------------------------------------------------------------------
Ethernet7/1                     DSCP                  DSCP
Ethernet7/2                     UNTRUSTED             UNTRUSTED
Ethernet7/3                     COS                   COS
Ethernet7/4                     COS                   DEFAULT
Ethernet8/1                     DSCP                  DEFAULT

switch(config-if-Et8/1)#
```

Configuring Default Port Settings

Default CoS and DSCP settings are assigned to individual port channel and Ethernet interfaces. These configuration mode interface commands specify the port's default CoS and DSCP values.

- qos cos configures a port's default CoS value.
- qos dscp configures a port's default DSCP value.

Example

These commands configure default ***CoS 4*** and ***DSCP 44*** values on ***interface ethernet 7/3***.

```
switch(config)# interface ethernet 7/3
switch(config-if-Et7/3)# qos cos 4
switch(config-if-Et7/3)# qos dscp 44
switch(config-if-Et7/3)# show active

interface Ethernet7/3
   qos cos 4
   qos dscp 44

switch(config-if-Et7/3)# show qos interfaces ethernet 7/3

Ethernet7/3:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44
```

```
switch(config-if-Et7/3)#
```

10.1.8.2 Traffic Class Derivations – Trident II and Helix Platform Switches

Traffic Classes describes traffic classes.

Note: Qos traffic policy is supported on Trident II platform switches.

Traffic Class Derivation Source

The following table displays the source for deriving a data stream's traffic class.

Table 34: Traffic Class Derivation Source: Trident II Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

CoS and DSCP Port Settings – Trident II and Helix Platform Switches describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The qos map cos command assigns a traffic class to a list of CoS settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

This command assigns the ***traffic class 5*** to the classes of service ***1***, ***3***, ***5***, and ***7***.

```
switch(config)# qos map cos 1 3 5 7 to traffic-class 5
switch(config)# show qos maps
   Number of Traffic Classes supported: 8

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ---------------------------
     tc:   1  5  2  5  4  5  6  5

switch(config)#
```

The following table displays the default CoS–traffic class map on Trident II platform switches.

Table 35: Default CoS to Traffic Class Map: Trident II Platform Switches

Inbound CoS	0	1	2	3	4	5	6	7
Traffic Class	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The qos map dscp command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

This command assigns the *traffic class 0* to DSCP values of ***12***, ***24***, ***41***, and ***44-47***.

```
switch(config)# qos map dscp 12 24 41 44 45 46 47 to traffic-class 0
switch(config)# show qos maps
   Number of Traffic Classes supported: 8

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  0  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  0  5  5  0  0  0  0  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

switch(config)#
```

The following table displays the default DSCP–traffic class map on Trident II platform switches.

Table 36: Default DSCP to Traffic Class Map: Trident II Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

10.1.8.3 CoS and DSCP Rewrite – Trident II and Helix Platform Switches

Rewriting CoS and DSCP describes the CoS and DSCP rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The qos map traffic-class to cos command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

This command assigns the CoS of two to traffic classes ***1***, ***3***, and ***5***.

```
switch(config)# qos map traffic-class 1 3 5 to cos 2
switch(config)#show qos map

   Number of Traffic Classes supported: 8

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ---------------------------
     cos:  1  2  2  2  4  2  6  7

switch(config)#
```

The following table displays the default Traffic Class to CoS rewrite value map on Trident II platform switches.

Table 37: Default Traffic Class to CoS Rewrite Value Map: Trident II Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

The DSCP rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-DSCP rewrite map. The qos map traffic-class to dscp command assigns a DSCP rewrite value to a list of traffic classes. Multiple commands create the complete traffic class-DSCP rewrite map.

Example

This command assigns the DSCP value of ***29*** to traffic classes ***2***, ***4***, and ***6***.

```
switch(config)# qos map traffic-class 2 4 6 to dscp 29
switch(config)# show qos map

   Number of Traffic Classes supported: 8

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ---------------------------
     dscp:  8  0 29 24 29 40 29 56

switch(config)#
```

The following table displays the default traffic class to DSCP rewrite map on Trident II platform switches.

Table 38: Traffic Class to DSCP Rewrite Value Map: Trident II Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
DSCP	8	0	16	24	32	40	48	56

10.1.8.4 Transmit Queues and Port Shaping – Trident II and Helix Platform Switches

Transmit Queues and Port Shaping describes transmit queues and port shaping.

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class-transmit queue map for all Ethernet interfaces and is used for unicast and multicast traffic. The traffic class to transmit queue maps are configured globally and apply to all Ethernet and port channel interfaces. The show qos maps command displays the traffic class to transmit queue map.

Trident II platform switches have eight unicast (UC0 – UC7) and eight multicast (MC0 – MC7) queues. Each UCx-MCx queue set is combined into a single queue group (L1.x), which is exposed to the CLI through this command.

The following table displays the default traffic class to transmit queue maps.

Table 39: Default Traffic Class to Transmit Queue Map: Trident II Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue Group	0	1	2	3	4	5	6	7

Mapping Traffic Classes to a Transmit Queue

The qos map traffic-class to tx-queue command assigns traffic classes to a transmit queue. Multiple commands create the complete map.

Example

These commands assign traffic classes of ***1***, ***3***, and ***5*** to ***transmit queue 1***, traffic classes ***2***, ***4***, and ***6*** to ***transmit queue 2***, and ***traffic class 0*** to ***transmit queue 0***, then display the resultant map.

```
switch(config)# qos map traffic-class 1 3 5 to tx-queue 1
switch(config)# qos map traffic-class 2 4 6 to tx-queue 2
switch(config)# qos map traffic-class 0 to tx-queue 0
switch(config)# show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8

   Tc - tx-queue map:
     tc:         0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:   0  1  2  1  2  1  2  7

switch(config)#
```

Entering a Transmit Queue Configuration Mode

Transmit queues are configurable on Ethernet ports and port channels. Queue parameters are configured in tx-queue configuration command mode, which is entered from the appropriate interface configuration mode. The tx-queue (Trident II) command places the switch in tx-queue configuration mode. The show qos interfaces displays the transmit queue configuration for a specified port.

Example

This command enters tx-queue configuration mode for ***transmit queue 3*** of ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# tx-queue 3
switch(config-if-Et5-txq-3)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies the port's maximum outbound traffic bandwidth. A shape rate can also be configured for all transmit queues on each port. All shape rate commands use kbps to specify data rates.

- To configure a port's shape rate, enter shape rate (Interface – Trident II) from the port's ***interface*** configuration mode.
- To configure a transmit queue's shape rate, enter shape rate (Tx-queue – Trident II) from the queue's ***tx-queue*** configuration mode.

Example

These commands configure a shape rate of ***5*** Gbs on ***interface Ethernet 3***, then configure the shape rate for the following transmit queues:

- - transmit queues ***0***, ***1***, and ***2***: ***500*** Mbps
  - transmit queues ***3***, ***4***, and ***5***: ***400*** Mbps

```
switch(config)# interface ethernet 17/3
switch(config-if-Et17/3)# shape rate 5000000
switch(config-if-Et17/3)# tx-queue 0
switch(config-if-Et17/3-txq-0)# shape rate 500000
switch(config-if-Et17/3-txq-0)# tx-queue 1
```

```
switch(config-if-Et17/3-txq-1)# shape rate 500000
switch(config-if-Et17/3-txq-1)# tx-queue 3
switch(config-if-Et17/3-txq-3)# shape rate 400000
switch(config-if-Et17/3-txq-3)# tx-queue 4
switch(config-if-Et17/3-txq-4)# shape rate 400000
switch(config-if-Et17/3-txq-4)# tx-queue 5
switch(config-if-Et17/3-txq-5)# shape rate 400000
switch(config-if-Et17/3-txq-5)# exit
switch(config-if-Et17/3)# show qos interface ethernet 17/3
Ethernet17/3:

  Tx       Bandwidth                  Shape Rate         Priority
 Queue     Guaranteed (units)          (units)
   ------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
   5        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   4        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   3        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   2        - / -    (  -  )       - / -    (  -  )    SP / SP
   1        - / -    (  -  )     500 / 500  ( Mbps )   SP / SP
   0        - / -    (  -  )     500 / 500  ( Mbps )   SP / SP

switch(config-if-Et17/3)#
```

Configuring Queue Priority

Queue priority rank is denoted by the queue number; transmit queues with higher numbers have higher priority. Trident II supports strict priority queues; round robin queues are not supported.

The bandwidth guaranteed (Trident II) command configures specifies the minimum bandwidth for outbound traffic on the transmit queue.

Example

These commands configure a minimum egress bandwidth of ***1*** Mbps for ***transmit queue 4*** on ***interface ethernet 17/3***.

```
switch(config-if-Et17/3)# tx-queue 4
switch(config-if-Et17/3-txq-4)# show qos interface ethernet 17/3

  Tx       Bandwidth                  Shape Rate         Priority
 Queue     Guaranteed (units)          (units)
   ------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
   5        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   4        1 / 1    ( Mbps )    400 / 400  ( Mbps )   SP / SP
   3        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   2        - / -    (  -  )       - / -    (  -  )    SP / SP
   1        - / -    (  -  )     500 / 500  ( Mbps )   SP / SP
   0        - / -    (  -  )     500 / 500  ( Mbps )   SP / SP

switch(config-if-Et17/3-txq-4)#
```

10.1.8.5 Ingress Policing on LAG

Ingress policing on a port-channel polices the matched traffic from all member interfaces combined, i.e. it provides aggregate policing and statistics (DCS-7050X, DCS-7010T, DCS-7250X, and DCS-7300X series). When a per-interface policer is attached to a port-channel, one set of TCAM entries is created for all member

interfaces. The associated interface bitmap is updated, and aggregate policing is performed on all member interfaces.

Examples

- These commands configure a service-policy (with policer action) on LAG by creating the service-policy and applying the service-policy on a port-channel.

```
switch(config)# policy-map policy-1
switch(config-pmap-qos-policy-1)# class class-1
switch(config-pmap-c-qos-policy-1-class-1)# police cir 512000 bps bc 96000
switch(config-pmap-c-qos-policy-1-class-1)# exit
switch(config-pmap-qos-policy-1)# exit
switch(config)# interface Et 4 / 5 / 4
switch(config-if-Et4/5/4)# channel-group 2 mode active
switch(config-if-Et4/5/4)# exit
switch(config)# interface po2
switch(config-if-Po2)# service-policy type qos input policy-1
switch(config-if-Po2)# exit
switch(config)#
```

- These commands configure ACL policing in single-rate, two-color mode.

```
switch(config)# class-map type qos match-any class1
switch(config-cmap-qos-class1)# match ip access-group acl1
switch(config-cmap-qos-class1)# exit
switch(config)# policy-map type quality-of-service policy1
switch(config-pmap-qos-policy1)# class class1
switch(config-pmap-c-qos-policy1-class1)# police cir 512000 bc 96000
switch(config-pmap-c-qos-policy1-class1)# exit
switch(config-pmap-qos-policy1)# exit
switch(config)# show policy-map
Service-policy policy1

  Class-map: class1 (match-any)
    Match: ip access-group name acl1
       police rate 512000 bps burst-size 96000 bytes

  Class-map: class-default (match-any)

switch(config)#
```

10.1.8.6 Fabric QoS -- – Trident II Platform Switches

EOS is optimized to support QoS configuration on the Fabric interfaces on 7250x and 7300 series switches. Configuring QoS on the Fabric interfaces in addition to front panel ports allows user to have end-to-end control and helps to manage traffic better over these switches. By default, tx queues are configured as strict priority on 7250X and 7300X series.

The following QoS configuration options are supported on Fabric interfaces on 7250x and 7300 series switches.

- **Guaranteed Bandwidth:** In order to prevent queue starvation on fabric ports EOS supports minimum bandwidth configuration on per queue basis across all fabric ports.
- **Explicit Congestion Notification (ECN):** EOS supports enabling ECN on a per queue basis across all fabric ports.
- **Priority Flow Control (PFC):** Queue back-pressure propagates across the backplane such that flow control messages can be generated to the upstream devices. This is done by enabling PFC for the desired backplane traffic-classes.

- **Weight Round Robin (WRR):** EOS supports configuring Weighted Round Robin (WRR) on a per queue basis across all fabric ports.

10.1.8.6.1 Configuring Fabric QoS on 7250X and 7300X Series

Fabric QoS is configured using a QoS profiles which is then applied on fabric interfaces on 7250x and 7300x series. Following are the steps to configure Fabric QoS.

1. Use **`qos profile`** command to create a QoS profile.
2. Use **`tx-queue (Trident II)`** command to configure a transmit queue on the configuration mode interface.
3. Use **`bandwidth guaranteed (Trident II)`** command to specifie the minimum bandwidth for outbound traffic on the transmit queue.
4. Use **`random-detect ecn (Trident)`** command to enable the ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes.
5. Use **`priority-flow-control priority`** command to configure the packet resolution setting on the configuration mode interface.
6. Use **`interface fabric`** command to configure Fabric interface.
7. Use **`service-profile`** command to apply the QoS profile to the Fabric interface.

Examples

- These commands create a QoS profile named fabricProfile with tx queue, bandwidth, ECN, PFC and DLB values defined in it and then the profile is attached to Fabric interface of the switch.



Note: To support PFC on a particular priority, DLB is disabled for that priority.

```
switch(config)# qos profile fabricProfile
switch(config-qos-profile-fabricProfile)# tx-queue 0
switch(config-qos-profile-fabricProfile-txq-0)# bandwidth guaranteed
 10000 kbps
switch(config-qos-profile-fabricProfile-txq-0)# random-detect ecn
minimum-threshold 10 mbytes maximum-threshold 10 mbytes
switch(config-qos-profile-fabricProfile)# priority-flow-control
 priority 1
no-drop
switch(config-qos-profile-fabricProfile)# priority-flow-control
 priority 6
no-drop dlb
```

- Applying the QoS profile on **`interface fabric`** of the switch.

```
switch# configure terminal
switch(config)# interface fabric
switch(config-if-fabric)# service-profile fabricProfile
```

10.1.8.6.2 Displaying Fabric QoS Information

These show commands display the configured Fabric QoS information on the switch.

- **`show qos profile`** **[*profile Name*]:** Displays the list of QoS profiles configured on the switch.
- **`show qos interfaces fabric:`** Displays the profile applied on the fabric interface on the switch.

Examples

- This command displays the ***fabricProfile*** profile information.

```
switch# show qos profile fabricProfile
  qos profile fabricProfile
     priority-flow-control priority 1 no-drop
```

```
     priority-flow-control priority 6 no-drop dlb
     tx-queue 0
         bandwidth guaranteed 10000 kbps
         random-detect ecn minimum-threshold 10 mbytes maximum-threshold 10
 mbytes
```

- This command displays the profile applied on the fabric interface.

```
switch# show qos interfaces fabric
   qos profile fabricProfile
```

10.1.9 Support for Configuring Color Extended Communities

EOS Release 4.23.1F introduces the ability to configure the color extended community in route-map set clauses and in an extcommunity-list for inbound and outbound policy application.

Use the color extended community for per-destination steering into Segment-Routing Traffic-Engineered (SR-TE) policies. If the next-hop and color of a BGP route match a particular policy (composed of an endpoint and color), any traffic bound to the destination can be steered according to the policy instead of forwarded via an IGP path or tunnel.

This section describes support for configuring color extended communities, including configuration instructions and command descriptions. Topics covered by this section include:

- Platform Compatibility
- Configuration
- Limitations

10.1.9.1 Platform Compatibility

Configuring color extended communities is supported on all platforms.

10.1.9.2 Configuration

Use the following command for color extended community expressions:

color COLOR-VALUE [color-only (exact-match | (endpoint-match (any | null)))]

Use this command in route-maps and extcommunity-lists to apply inbound and outbound policy.

Configuring Color Extended Community in route-map mode	
Command(config-route-map mode)	**`[no | default] set extcommunity`** **COLOR-EXPRESSION** [additive | delete]
Action	Adds a color extended community to be applied to routes affected by the route-map. Multiple set clauses can be applied to a single route-map to configure multiple colors for routes. **Additive** adds the extended communities to those received, while **delete** deletes any matching extended color communities. Negating the command removes the entry from the route-map.
Default	None

Example	`switch(config)# route-map foo`
	`switch(config-route-map foo)# set extcommunity color 1`
	`switch(config-route-map foo)# set extcommunity color 2 color-only exact-match`
	`switch(config-route-map foo)# set extcommunity color 3 color-only endpoint-match null`
	`switch(config-route-map foo)# set extcommunity color 4 color-only endpoint-match any`

Configuring Color Extended Community in extcommunity-list	
Command(config mode)	`[(no|default)] ip extcommunity-list` **WORD** `(permit|deny){`**COLOR-EXPRESSION**`}`
Action	Adds a color extended community to an extcommunity-list. Multiple color extended communities can be added to the list. Negating the command removes the corresponding color extended community from the list.
Default	None
Example	`switch(config)# ip extcommunity-list foo permit color 1 color 2` `switch(config)# ip extcommunity-list bar permit color 3 color-only endpoint-match null color 4 color-only endpoint-match any`

10.1.9.3 Limitations

The color extended community is only supported in multi-agent mode. Enable Multi-agent mode via the following command:

`service routing protocols mode multi-agent`

10.1.10 ACL based QoS Configuration

10.1.10.1 ACL Based QoS (DCS-7160)

The IPv4 ACL based QoS is enabled on switches through policy-map configuration. The ACL based QoS can be configured on front panel ports, port-channel interfaces on **DCS-7160** series switches.

10.1.10.1.1 ACL based QoS on SVIs

The ACL based QoS policy applied on SVIs modify the QoS parameters for SVI traffic (L3 VLAN) based on ACL classification. The ACL based QoS on Switched Virtual Interface (SVI) ports is supported on DCS-7500E, DCS-7280E, DCS-7010, DCS-7050, DCS-7050X, DCS-7250X, DCS-7300X, DCS-7020TR.

10.1.10.1.2 ACL Sharing on QoS

The ACLs applied on QoS shares the hardware resources (TCAM) when potentially large QoS policy-maps are applied to multiple SVIs. For ACL based QoS on SVIs in sharing mode we share TCAM for class-maps without policer action and replicate entries for policer class-maps. The ACL Sharing on QoS is supported only on selected platforms.

The QoS actions is applicable only to the routed traffic flowing through the members of the corresponding VLAN.

The steps to configure ACL based QoS is as follows:

1. Create a access list using **`ip access-list`** command.
2. Create a class map and attach it to the access list using **`class-map`** command.
3. Create a policy and attach the class map to the policy created, using the **`policy-map`** command.
4. Apply the policy to the interface using the **`service-policy input`** command.

Examples

- These command configure the access list ***acl1***.

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# permit ip 10.1.1.1/24 any
switch(config-acl-acl1)# exit
```

- These commands configure the class map ***class1***.

```
switch(config)# class-map match-any class1
switch(config-cmap-qos-class1)# match ip access-group acl1
switch(config-cmap-qos-class1)# exit
```

- These commands configure the policy map ***policy1***.

```
switch(config)# policy-map policy1
switch(config-pmap-qos-policy1)# class class1
switch(config-pmap-c-qos-policy1-class1)# set dscp 20
switch(config-pmap-c-qos-policy1-class1)# set traffic-class 2
switch(config-pmap-c-qos-policy1-class1)# exit
switch(config-pmap-c-qos-policy1)# exit
```

- These commands apply the ***policy1*** to the ***interface ethernet 1/1***.

```
switch(config)# interface Et1/1
switch(config-if-Et1/1)# service-policy input policy1
switch(config-if-Et1/1)# exit
```

- These commands configure a ACL based QoS on the SVI interface ***VLAN10***.

```
switch(config)# interface vlan 10
switch(config-if-Vl10)# service-policy [type qos] input policy1
```

- This command enables the resource (hardware) sharing when a ACL based QoS is attached to VLAN interface. The no form of the command disables it.

```
switch(config)# hardware access-list qos resource sharing vlan in
```

- This command allows inbound broadcast IP packets with source IP address as one of the permitted hosts and denies the rest of the directed broadcast traffic.

```
switch(config)# ip directed-broadcast
 switch(config-ip-directed-broadcast)# field-set ipv4 prefix ALLOWED1
 switch(config-ipdb-field-set-ipv4-prefix-ALLOWED)# 10.1.1.1/32
 20.1.1.1/32 30.1.1.1/32
 switch(config-ipdb-field-set-ipv4-prefix-ALLOWED)# commit

 switch(config-ip-directed-broadcast)# field-set ipv4 prefix ALLOWED2
 switch(config-ipdb-field-set-ipv4-prefix-ALLOWED)# 10.2.1.1/32
 20.2.1.1/32 30.2.1.1/32
 switch(config-ipdb-field-set-ipv4-prefix-ALLOWED)# commit
```

Show Commands

The following show commands display the status of policy-maps programmed on the interface, for more information on these commands refer Quality of Service Commands.

- **`show policy-map [policy-name]`**: Displays the policy-map programming status.
- **`show policy-map interface interface id`**: Displays the policy-map that is currently programmed on the interface.
- **`show policy-map [policy-name] counters`**: Displays the policy-map traffic hits.
- **`show platform xp qos tcam [hits]`**: Displays the TCAM entries programmed for each policy-map as well as the traffic hits. The hits option is used to see the TCAM entries with nonzero traffic hits.
- **`show run | grep sharing`**: Displays if whether the ACL based QoS is enabled or disabled.
- **`show platform trident tcam shared vlan interface-class-id`**: Displays what SVIs are currently sharing the QoS policy-map.
- **`show platform trident tcam directed-broadcast`**: Displays the permitted hosts via field-set.
- **`show platform trident tcam qos detail`**: Displays the list of all the SVIs that are sharing the TCAM entries.

10.1.10.1.3 Limitations

- Maximum number of TCAM entries that can be programmed in hardware for all QoS policy-maps on the box is ***1024***.
- Layer 4 port ranges are not supported for ACL based QoS. The ranges will be expanded into multiple TCAM rules and programmed in the hardware.
- Configured policer rate should be above 1mbps and recommended burst value is ***2000*** bytes.
- Policer action cannot be associated with policy-maps applied to Port-Channels.

The following are the limitations specific to DCS-7500E, DCS-7280E and DCS-7020TR:

UThe user cannot apply more than 31 QoS service policies per chip on L3 interfaces.

- When different QoS service policies are applied to the SVI and its member interfaces, the behavior is indeterministic.
- When QoS service policies are applied on SVIs with partial failures due to limited hardware resources, any event that triggers a forwarding agent restart will lead to indeterministic behavior.

- When QoS service policies are applied on 2 SVIs, any event that triggers the VLAN membership change of a member interface may result in a policy-map programming failure. To change the VLAN membership, remove the interface from the first VLAN and then add it to the other.
- Outgoing COS rewrite is not supported.
- QoS policy-map counters are not supported.

The following are the limitations specific to DCS-7010, DCS-7050, DCS-7050X, DCS-7250X, DCS-7300X:

- TCAM resources will not be shared for the same policy-map applied to multiple SVIs.
- Policy-map applied to a SVI will result in TCAM allocation on all chips irrespective of whether the SVI members are present or not.

When QoS service policies are applied to both SVI and its member interfaces and packets hit both policies, the behavior is indeterministic.

10.1.11 Configuring IPv6 Flow Label Matches for QoS

In addition to criteria like COS and DSCP, QoS decisions can be based on the IPv6 flow labels of packets. To use IPv6 flow labels as a criterion for QoS decisions, use the **permit** command to create ACL rules to select packets based on their IPv6 flow labels using an exact match and an optional mask, and then apply the access list to a class map in a QoS policy map, which can then be applied to individual interfaces. This requires a TCAM profile which must be installed explicitly.

Install the QoS Flow Label TCAM Profile

Open the QoS flow label TCAM profile Web page at https://www.arista.com/en/support/toi/tcam-profile?pn=qos-match-ipv6-flow-label. Copy the entire contents of the file which starts with the following commands:

```
hardware tcam
system profile qos-match-ipv6-flow-label
```

There are approximately 130 lines.

Paste this into the CLI of the switch. Each command in the file will be executed in turn when you paste the file contents. When the paste completes, if the cursor is at the end of the last command, type **enter** to execute it.

Confirm the QoS Flow Label TCAM Profile Installation

Configure the TCAM with the **hardware tcam** command, and use the **system profile** command to confirm that the “qos-match-ipv6-flow-label” profile has been installed. If the profile is installed, the **?** operator will show “qos-match-ipv6-flow-label” as an available profile.

Example

The following commands confirm whether the “qos-match-ipv6-flow-label” profile is installed and available to be applied.

```
switch(config)#hardware tcam
switch(config-tcam)#system profile qos-match?
WORD  qos-match-ipv6-flow-label
switch(config-tcam)#exit
switch(config)#
```

Apply the QoS Flow Label TCAM Profile

If the profile "qos-match-ipv6-flow-label" is listed, use the **system profile** command to apply it. Confirm that the profile has been successfully applied with the **show hardware tcam profile** command.

Example

The following commands apply the profile “qos-match-ipv6-flow-label” to the TCAM configuration, then confirm that the profile has been applied. The warning is normal, as the restart of forwarding agents is part of the procedure.

```
switch(config)#hardware tcam
switch(config-tcam)#system profile qos-match-ipv6-flow-label
!
WARNING!
Changing TCAM profile will cause forwarding agent(s) to exit and restart.
All traffic through the forwarding chip managed by the restarting
forwarding agent will be dropped.

Proceed [y/n]y
switch(config-tcam)#exit
switch(config)#show hardware tcam profile
                    Configuration            Status
FixedSystem         qos-match-ipv6-flow-label qos-match-ipv6-flow-label
switch(config)#
```

Create IPv6 ACL Rules

Configure an access list with the **`ipv6 access-list`** command. Then create rules with the **`permit ipv6`** command.

Example

The following commands create an IPv6 ACL rule which matches the flow label ***23*** in the access list ***L1***.

```
switch(config)#ipv6 access-list L1
switch(config-ipv6-acl-L1)#permit ipv6 any any flow-label eq 23
switch(config-ipv6-acl-L1)#exit
switch(config)#
```

Add Access List to QoS Class Map

Configure a QoS class map with the **`class-map`** command. Then add an access list to the class map with the **`match`** command.

Example

The following commands create a QoS class map called ***C1***, and then add the access list ***L1***.

```
switch(config)#class-map type qos match-any C1
switch(config-cmap-qos-C1)#match ipv6 access-group L1
switch(config-cmap-qos-C1)#exit
switch(config)#
```

Add Class Map to Policy Map

Configure a QoS policy map, creating it if necessary, with the **`policy-map`** command. Then add a class map to the policy map and define the traffic class.

Example

The following commands configure the QoS policy map ***P1*** and add the class map ***C1***, and then assign the class to traffic class ***4***.

```
switch(config)#policy-map type quality-of-service P1
switch(config-pmap-quality-of-service-P1)#class C1
switch(config-pmap-c-quality-of-service-P1-C1)#set traffic-class 4
switch(config-pmap-c-quality-of-service-P1-C1)#exit
```

```
switch(config-pmap-quality-of-service-P1)#exit
switch(config)#
```

Configure An Interface with QoS Policy

Configure an interface with the **interface** command. Apply the QoS policy to the interface with the **service-policy** command.

Example

The following commands configure the Ethernet interface ***1/1***, and apply the QoS policy ***P1*** to the interface.

```
switch(config)#interface Ethernet 1/1
switch(config-if-Et1/1)#service-policy type qos input P1
switch(config-if-Et1/1)#exit
switch(config)#
```

Example

The following commands confirm that the policy map ***P1*** has been programmed successfully.

```
switch(config)#show policy-map P1
Service-policy input: P1
  Hardware programming status: Successful

  Class-map: C1 (match-any)
    Match: ipv6 access-group name L1
       set traffic-class 4

  Class-map: class-default (match-any)

switch(config)#show ipv6 access-lists L1
IPV6 Access List L1
        10 permit ipv6 any any flow-label eq 23
switch(config)#
```

10.1.12 Differentiated MMU Discard Counters

To count discarded packets through tagging, assign drop-precendes for a certain class of packets on platforms that support such tagging.

Configuring Differentiated MMU Discard Counters

The following steps configure the Differentiated MMU Discard Counters.

1. Configure an IP access-lists to match traffic

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# permit 41 any any !!41 = 0x29 = IPv6
```

2. Add the access-list to a class-map

```
switch(config)# class-map type qos match-any class1
switch(config-cmap-qos-class1)# match ip access-group acl1
```

3. Add the class-map to a policy-map

```
switch(config)# policy-map type quality-of-service policy1
switch(config-pmap-quality-of-service-policy1)# class class1
```

4. Add drop-precedence action to the policy-map

```
switch(config-pmap-c-quality-of-service-policy1-class1)# set drop-precedence
 2
```

5. Create qos profile with the policy map assigned

```
switch(config)# qos profile qos1
switch(config-qos-profile-qos1)# service-policy input policy1
```

6. Apply the policy-map to the interface

```
switch(config)# int et3/1
switch(config-if-Et3/1)# service-profile qos1
```

Repeat step 6 on all interfaces that receive traffic to be counted. Packets are tagged on ingress. If the same packet gets dropped on egress as an MMU discard, the corresponding counter gets incremented.

Displaying Differentiated MMU Discard Counters

The **show interface counters queue drop-precedence** command displays the drop-precedence counters. Based on the above configuration, drop-precedence 2 counts IPv6 packets.

```
switch# show interface counters queue drop-precedence
intf         0           1           2
Et1/1      100           0         200
Et1/2      200           0         300
…
```

10.1.13 Quality of Service Commands

QoS Data Field and Traffic Class Configuration Commands

- color
- hardware access-list qos resource sharing vlan in
- ip extcommunity-list
- platform petraA traffic-class
- qos cos
- qos dscp
- qos map cos
- qos map dscp
- qos map dscp to traffic-class
- qos map dscp to traffic-class (MPLS tunnel termination VRF)
- qos map traffic-class to cos
- qos map traffic-class to dscp
- qos map traffic-class to mc-tx-queue
- qos map traffic-class to tx-queue
- qos map traffic-class to uc-tx-queue
- qos profile
- qos rewrite cos
- qos rewrite dscp
- qos trust
- service-policy type qos input
- service-profile
- set extcommunity

QoS and ECN Display Commands

- show interface counters queue drop-precedence
- show platform petraA traffic-class
- show platform trident tcam shared vlan interface-class-id
- show platform trident tcam qos detail
- show platform xp qos tcam hit
- show policy-map
- show policy-map interface
- show qos interfaces
- show qos interfaces latency maximum
- show qos interfaces trust
- show qos interfaces random-detect ecn
- show qos maps
- show qos map dscp to traffic-class
- show qos profile
- show run|grep sharing
- show qos profile summary
- show qos random-detect ecn

ECN Configuration Commands

- qos random-detect ecn allow non-ect chip-based (Tomahawk and Trident)
- qos random-detect ecn global-buffer (Helix)
- qos random-detect ecn global-buffer (Trident and Tomahawk)
- random-detect ecn (Arad/Jericho)
- random-detect ecn (Helix)
- random-detect ecn (Trident and Tomahawk)

Transmit Queue and Port Shaping Commands – Arad and Jericho Platforms

- bandwidth percent (Arad/Jericho)
- priority (Arad/Jericho)
- shape rate (Interface – Arad/Jericho)
- shape rate (Tx-queue – Arad/Jericho)
- tx-queue (Arad/Jericho)

Transmit Queue and Port Shaping Commands – FM6000 Platform

- bandwidth percent (FM6000)
- priority (FM6000)
- shape rate (Interface – FM6000)
- shape rate (Tx-queue – FM6000)
- tx-queue (FM6000)

Transmit Queue and Port Shaping Commands – Helix Platform

- bandwidth guaranteed (Helix)
- shape rate (Interface – Helix)
- shape rate (Tx-queue – Helix)
- tx-queue (Helix)

Transmit Queue and Port Shaping Commands – Petra Platform

- bandwidth percent (Petra)
- priority (Petra)
- shape rate (Interface – Petra)
- shape rate (Tx-queue – Petra)
- tx-queue (Petra)

Transmit Queue and Port Shaping Commands – Trident and Tomahawk Platform

- bandwidth percent (Trident and Tomahawk)
- mc-tx-queue
- priority (Trident and Tomahawk)
- shape rate (Interface – Trident and Tomahawk)
- shape rate (Tx-queue – Trident and Tomahawk)
- uc-tx-queue

Transmit Queue and Port Shaping Commands – Trident II Platform

-
- bandwidth guaranteed (Trident II)
- shape rate (Tx-queue – Trident II)
- shape rate (Interface – Trident II)
- tx-queue (Trident II)
- interface fabric (Trident II)

10.1.13.1 bandwidth guaranteed (Helix)

The **`bandwidth guaranteed`** command specifies the minimum bandwidth for outbound traffic on the transmit queue. By default, no bandwidth is guaranteed to any transmit queue.

The **`no bandwidth guaranteed`** and **`default bandwidth guaranteed`** commands remove the minimum bandwidth guarantee on the transmit queue by deleting the corresponding **`bandwidth guaranteed`** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

`bandwidth guaranteed rate` **DATA_MIN**

`no bandwidth guaranteed`

`default bandwidth guaranteed`

Parameters

DATA_MIN Minimum bandwidth. Value range varies with data unit:

- ***8*** to ***40000000*** ***8*** to ***40000000*** kbytes per second.
- ***8*** to ***40000000*** **kbps** ***8*** to ***40000000*** kbytes per second.
- ***1*** to ***60000000*** **pps** ***1*** to ***60000000*** packets per second.

Related Command

tx-queue (Helix) places the switch in tx-queue configuration mode.

Example

These commands configure a minimum egress bandwidth of ***1*** Mbps for ***transmit queue 4*** of ***interface ethernet 17/3***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# tx-queue 4
switch(config-if-Et17-txq-4)# bandwidth guaranteed 1000 kbps
switch(config-if-Et17-txq-4)# show qos interfaces ethernet 17

Ethernet17/3:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

  Tx       Bandwidth                  Shape Rate         Priority
 Queue     Guaranteed (units)          (units)
   ---------------------------------------------------------------
   7         - / -    (  -  )       - / -    (  -  )    SP / SP
   6         - / -    (  -  )       - / -    (  -  )    SP / SP
   5         - / -    (  -  )       - / -    (  -  )    SP / SP
   4         1 / 1    ( Mbps )      - / -    (  -  )    SP / SP
   3         - / -    (  -  )       - / -    (  -  )    SP / SP
   2         - / -    (  -  )       - / -    (  -  )    SP / SP
   1         - / -    (  -  )       - / -    (  -  )    SP / SP
   0         - / -    (  -  )       - / -    (  -  )    SP / SP

Note: Values are displayed as Operational/Configured

switch(config-if-Et17-txq-4)#
```

10.1.13.2 bandwidth guaranteed (Trident II)

The **`bandwidth guaranteed`** command specifies the minimum bandwidth for outbound traffic on the transmit queue. By default, no bandwidth is guaranteed to any transmit queue.

The **`no bandwidth guaranteed`** and **`default bandwidth guaranteed`** commands remove the minimum bandwidth guarantee on the transmit queue by deleting the corresponding **`bandwidth guaranteed`** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

`bandwidth guaranteed rate` **DATA_MIN**

`no bandwidth guaranteed`

`default bandwidth guaranteed`

Parameters

DATA_MIN minimum bandwidth. Value range varies with data unit:

- ***8*** to ***40000000*** kbytes per second.
- ***8*** to ***40000000*** **kbps** kbytes per second.
- ***1*** to ***60000000*****pps** packets per second.

Related Command

tx-queue (Trident II) places the switch in tx-queue configuration mode.

Example

These commands configure a minimum egress bandwidth of ***1*** Mbps for ***transmit queue 4*** of ***interface ethernet 17/3***.

```
switch(config)# interface ethernet 17/3
switch(config-if-Et17/3)# tx-queue 4
switch(config-if-Et17/3-txq-4)# bandwidth guaranteed 1000 kbps
switch(config-if-Et17/3-txq-4)# show qos interfaces ethernet 17/3

Ethernet17/3:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

  Tx       Bandwidth                  Shape Rate         Priority
 Queue     Guaranteed (units)          (units)
   ---------------------------------------------------------------
   7         - / -    (  -  )       - / -    (  -  )    SP / SP
   6         - / -    (  -  )       - / -    (  -  )    SP / SP
   5         - / -    (  -  )       - / -    (  -  )    SP / SP
   4         1 / 1    ( Mbps )      - / -    (  -  )    SP / SP
   3         - / -    (  -  )       - / -    (  -  )    SP / SP
   2         - / -    (  -  )       - / -    (  -  )    SP / SP
   1         - / -    (  -  )       - / -    (  -  )    SP / SP
   0         - / -    (  -  )       - / -    (  -  )    SP / SP

Note: Values are displayed as Operational/Configured

switch(config-if-Et17/3-txq-4)#
```

10.1.13.3 bandwidth percent (Arad/Jericho)

The **`bandwidth percent`** command configures the bandwidth share of the transmit queue when configured for round robin priority. Bandwidth is allocated to all queues based on the cumulative configured bandwidth of all the port's round robin queues.

The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

The **`no bandwidth percent`** and **`default bandwidth percent`** commands restore the default bandwidth share of the transmit queue by removing the corresponding **`bandwidth percent`** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

`bandwidth percent` ***proportion***

`no bandwidth percent`

`default bandwidth percent`

Parameters

proportion Bandwidth percentage assigned to queues. Values range from ***1*** to ***100***.

Related Command

tx-queue (Arad/Jericho) places the switch in tx-queue configuration mode.

Examples

- These commands configure queues ***0*** through ***3*** (***interface ethernet 3/5/1***) as round robin, then allocate bandwidth for three queues at ***30%*** and one queue at ***10%***.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# tx-queue 3
switch(config-if-Et3/5/1-txq-3)# no priority
switch(config-if-Et3/5/1-txq-3)# bandwidth percent 10
switch(config-if-Et3/5/1-txq-3)# tx-queue 2
switch(config-if-Et3/5/1-txq-2)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-2)# tx-queue 1
switch(config-if-Et3/5/1-txq-1)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-1)# tx-queue 0
switch(config-if-Et3/5/1-txq-0)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-0)# show qos interfaces ethernet 3/5/1

Ethernet3/5/1:

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     10 / 10       - / -    (  -  )    RR / RR    D
   2     30 / 30       - / -    (  -  )    RR / SP    D
   1     30 / 30       - / -    (  -  )    RR / SP    D
   0     30 / 30       - / -    (  -  )    RR / SP    D

switch(config-if-Et3/5/1-txq-0)#
```

- These commands re-configure the bandwidth share of the fourth queue at ***30%***.

```
switch(config-if-Et3/5/1-txq-0)# tx-queue 3
switch(config-if-Et3/5/1-txq-3)# bandwidth percent 30
switch(config-if-Et3/5/1-txq-3)# show qos interfaces ethernet 3/5/1

Ethernet3/5/1:

   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     24 / 30       - / -    (  -  )    RR / RR    D
   2     24 / 30       - / -    (  -  )    RR / SP    D
   1     24 / 30       - / -    (  -  )    RR / SP    D
   0     24 / 30       - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured

switch(config-if-Et3/5/1-txq-3)#
```

- These commands configure the bandwidth share of the fourth queue at ***2%***.

```
switch(config-if-Et3/5/1-txq-3)# bandwidth percent 2
switch(config-if-Et3/5/1-txq-3)# show qos interfaces ethernet 3/5/1

Ethernet3/5/1:

   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)         (units)
   ------------------------------------------------------
   7       - / -        - / -    (  -  )    SP / SP    D
   6       - / -        - / -    (  -  )    SP / SP    D
   5       - / -        - / -    (  -  )    SP / SP    D
   4       - / -        - / -    (  -  )    SP / SP    D
   3       2 / 2        - / -    (  -  )    RR / RR    D
   2      30 / 30       - / -    (  -  )    RR / SP    D
   1      30 / 30       - / -    (  -  )    RR / SP    D
   0      30 / 30       - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured

switch(config-if-Et3/5/1-txq-3)#
```

10.1.13.4 bandwidth percent (FM6000)

The **`bandwidth percent`** command configures the bandwidth share of the transmit queue when configured for round robin priority. Bandwidth is allocated to all queues based on the cumulative configured bandwidth of all the port's round robin queues.

The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

The **`no bandwidth percent`** and **`default bandwidth percent`** commands restore the default bandwidth share of the transmit queue by removing the corresponding **`bandwidth percent`** command ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

`bandwidth percent` ***proportion***

`no bandwidth percent`

`default bandwidth percent`

Parameters

proportion Configured bandwidth percentage. Value ranges from ***1*** to ***100***. Default value is ***0***.

Related Command

tx-queue (FM6000) places the switch in tx-queue configuration mode.

Examples

- These commands configure queues ***0*** through ***3*** (***interface Ethernet 19***) as round robin, then allocate bandwidth for three queues at ***30%*** and one queue at ***10%***.

```
switch(config)# interface Ethernet 19
switch(config-if-Et19)# tx-queue 3
switch(config-if-Et19-txq-3)# no priority
switch(config-if-Et19-txq-3)# bandwidth percent 10
switch(config-if-Et19-txq-3)# tx-queue 2
switch(config-if-Et19-txq-2)# bandwidth percent 30
switch(config-if-Et19-txq-2)# tx-queue 1
switch(config-if-Et19-txq-1)# bandwidth percent 30
switch(config-if-Et19-txq-1)# tx-queue 0
switch(config-if-Et19-txq-0)# bandwidth percent 30
switch(config-if-Et19-txq-0)# show qos interface ethernet 19

Ethernet19:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

 Tx    Bandwidth  Bandwidth          Shape Rate     Priority  ECN/
WRED
 Queue (percent)  Guaranteed (units)  units)

-------------------------------------------------------------------
-------
  7     - / -      - / -     ( - )     - / -  ( - )  SP / SP    D
  6     - / -      - / -     ( - )     - / -  ( - )  SP / SP    D
  5     - / -      - / -     ( - )     - / -  ( - )  SP / SP    D
  4     - / -      - / -     ( - )     - / -  ( - )  SP / SP    D
  3    10 / 10     - / -     ( - )     - / -  ( - )  RR / RR    D
  2    30 / 30     - / -     ( - )     - / -  ( - )  RR / SP    D
  1    30 / 30     - / -     ( - )     - / -  ( - )  RR / SP    D
  0    30 / 30     - / -     ( - )     - / -  ( - )  RR / SP    D

Note: Values are displayed as Operational/Configured

Legend:
RR -> Round Robin
SP -> Strict Priority
 -  -> Not Applicable / Not Configured
ECN/WRED: L -> Queue Length ECN Enabled     W -> WRED Enabled     D
 -> Disabled
```

- These commands re-configure the bandwidth share of ***transmit queue 3*** at ***30%***.

```
cp118.14:04:20# config
cp118.14:04:23(config)# interface ethernet 19
cp118.14:04:47(config-if-Et19-txq-0)# tx-queue 3
cp118.14:04:59(config-if-Et19-txq-3)# bandwidth percent 30
cp118.14:05:16(config-if-Et19-txq-3)# show qos interface ethernet 19

Ethernet19:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

 Tx   Bandwidth   Bandwidth          Shape Rate      Priority  ECN/WRED
Queue (percent)   Guaranteed (units)  (units)

-----------------------------------------------------------------------
  7     - / -      - / -     ( - )     - / -  ( - )   SP / SP    D
```

```
  6     - / -      - / -     ( - )     - / -  ( - )   SP / SP    D
  5     - / -      - / -     ( - )     - / -  ( - )   SP / SP    D
  4     - / -      - / -     ( - )     - / -  ( - )   SP / SP    D
  3    24 / 30     - / -     ( - )     - / -  ( - )   RR / RR    D
  2    24 / 30     - / -     ( - )     - / -  ( - )   RR / SP    D
  1    24 / 30     - / -     ( - )     - / -  ( - )   RR / SP    D
  0    24 / 30     - / -     ( - )     - / -  ( - )   RR / SP    D

Note: Values are displayed as Operational/Configured

Legend:
RR -> Round Robin
SP -> Strict Priority
 -  -> Not Applicable / Not Configured
ECN/WRED: L -> Queue Length ECN Enabled     W -> WRED Enabled     D -> Disabled
```

- These commands re-configure the bandwidth share of ***transmit queue 3*** at ***2%***.

```
cp118.14:09:37(config-if-Et19-txq-3)# bandwidth percent 2
cp118.14:12:56(config-if-Et19-txq-3)# show qos interface ethernet 19

Ethernet19:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

 Tx   Bandwidth  Bandwidth           Shape Rate    Priority  ECN/WRED
Queue (percent)  Guaranteed (units)  (units)

------------------------------------------------------------------------
  7     - / -      - / -     ( - )   - / -  ( - )   SP / SP   D
  6     - / -      - / -     ( - )   - / -  ( - )   SP / SP   D
  5     - / -      - / -     ( - )   - / -  ( - )   SP / SP   D
  4     - / -      - / -     ( - )   - / -  ( - )   SP / SP   D
  3     2 / 2      - / -     ( - )   - / -  ( - )   RR / RR   D
  2    30 / 30     - / -     ( - )   - / -  ( - )   RR / SP   D
  1    30 / 30     - / -     ( - )   - / -  ( - )   RR / SP   D
  0    30 / 30     - / -     ( - )   - / -  ( - )   RR / SP   D

Note: Values are displayed as Operational/Configured

Legend:
RR -> Round Robin
SP -> Strict Priority
 -  -> Not Applicable / Not Configured
ECN/WRED: L -> Queue Length ECN Enabled     W -> WRED Enabled     D -> Disabled
```

10.1.13.5 bandwidth percent (Petra)

The **`bandwidth percent`** command configures the bandwidth share of the transmit queue when configured for round robin priority. Bandwidth is allocated to all queues based on the cumulative configured bandwidth of all the port's round robin queues.

The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

The **`no bandwidth percent`** and **`default bandwidth percent`** commands restore the default bandwidth share of the transmit queue by removing the corresponding **`bandwidth percent`** command ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

`bandwidth percent` ***proportion***

no bandwidth percent

default bandwidth percent

Parameters

proportionBandwidth percentage assigned to queues. Values range from ***1*** to ***100***.

Related Command

tx-queue (Petra) places the switch in tx-queue configuration mode.

Examples

- These commands configure queues ***0*** through ***3*** (***interface ethernet 3/28***) as round robin, then allocate bandwidth for three queues at ***30%*** and one queue at ***10%***.

```
switch(config)# interface ethernet 3/28
switch(config-if-Et3/28)# tx-queue 3
switch(config-if-Et3/28-txq-3)# no priority
switch(config-if-Et3/28-txq-3)# bandwidth percent 10
switch(config-if-Et3/28-txq-3)# tx-queue 2
switch(config-if-Et3/28-txq-2)# bandwidth percent 30
switch(config-if-Et3/28-txq-2)# tx-queue 1
switch(config-if-Et3/28-txq-1)# bandwidth percent 30
switch(config-if-Et3/28-txq-1)# tx-queue 0
switch(config-if-Et3/28-txq-0)# bandwidth percent 30
switch(config-if-Et3/28-txq-0)# show qos interface ethernet 3/28

Ethernet3/28:

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          10     disabled   round-robin
          2          30     disabled   round-robin
          1          30     disabled   round-robin
          0          30     disabled   round-robin

switch(config-if-Et3/28-txq-0)#
```

- These commands re-configure the bandwidth share of the fourth queue at ***30%***.

```
switch(config-if-Et3/28-txq-0)# tx-queue 3
switch(config-if-Et3/28-txq-3)# bandwidth percent 30
switch(config-if-Et3/28-txq-3)# show qos interface ethernet 3/28

Ethernet3/28:
   Trust Mode: COS

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
---------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          24     disabled   round-robin
          2          24     disabled   round-robin
          1          24     disabled   round-robin
```

```
          0          24     disabled   round-robin

switch(config-if-Et3/28-txq-3)#
```

- These commands configure the bandwidth share of the fourth queue at ***2%***.

```
switch(config-if-Et3/28)# tx-queue 3
switch(config-if-Et3/28-txq-3)# bandwidth percent 2
switch(config-if-Et3/28-txq-3)# show qos interface ethernet 3/28

Ethernet3/28:
   Trust Mode: COS


   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3           2     disabled   round-robin
          2          30     disabled   round-robin
          1          30     disabled   round-robin
          0          30     disabled   round-robin

switch(config-if-Et3/28-txq-3)#
```

10.1.13.6 bandwidth percent (Trident and Tomahawk)

The **`bandwidth percent`** command configures the bandwidth share of the transmit queue when configured for round robin priority. Bandwidth is allocated to all queues based on the cumulative configured bandwidth of all the port's round robin queues.

The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

The **`no bandwidth percent`** and **`default bandwidth percent`** commands restore the default bandwidth share of the transmit queue by removing the corresponding **`bandwidth percent`** command ***running-config***.

Command Mode

Mc-Tx-Queue configuration

Uc-Tx-Queue configuration

Command Syntax

`bandwidth percent` ***proportion***

`no bandwidth percent`

`default bandwidth percent`

Parameters

proportion Bandwidth percentage assigned to queues. Values range from ***1*** to ***100***.

Related Commands

- mc-tx-queue places the switch in mc-tx-queue configuration mode.
- uc-tx-queue places the switch in uc-tx-queue configuration mode.

Examples

- These commands configure ***unicast transmit queue 3*** (and all other queues of lower priority) as round robin, then allocate bandwidth for unicast transmit queues ***1***, ***2***, and ***3*** at ***30%*** and ***multicast transmit queue 1*** at ***10%***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)# no priority
switch(config-if-Et7-uc-txq-3)# bandwidth percent 30
switch(config-if-Et7-uc-txq-3)# uc-tx-queue 2
switch(config-if-Et7-uc-txq-2)# bandwidth percent 30
switch(config-if-Et7-uc-txq-2)# uc-tx-queue 1
switch(config-if-Et7-uc-txq-1)# bandwidth percent 30
switch(config-if-Et7-uc-txq-1)# mc-tx-queue 1
switch(config-if-Et7-mc-txq-1)# bandwidth percent 10
switch(config-if-Et7-mc-txq-1)# show qos interfaces ethernet 7

Ethernet7:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3          30     disabled   round-robin                0
        UC2          30     disabled   round-robin                0
        MC1          10     disabled   round-robin                0
        UC1          30     disabled   round-robin                0
        UC0           0     disabled   round-robin                0
        MC0           0     disabled   round-robin                0

switch(config-if-Et7-mc-txq-1)#
```

- These commands re-configure the bandwidth share of ***unicast queue 3*** at ***55%***.

```
switch(config-if-Et7-mc-txq-1)# uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)# bandwidth percent 55
switch(config-if-Et7-uc-txq-3)# show qos interface ethernet 7

Ethernet7:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3          44     disabled   round-robin                0
        UC2          24     disabled   round-robin                0
        MC1           8     disabled   round-robin                0
        UC1          24     disabled   round-robin                0
        UC0           0     disabled   round-robin                0
        MC0           0     disabled   round-robin                0
```

```
switch(config-if-Et7-uc-txq-3)#
```

10.1.13.7 color

Use the **color** command for the ability to configure the color extended community in route-map set clauses and in an extcommunity-list for inbound and outbound policy application.

Command Mode

Route-maps and extcommunity-lists

Command Syntax

color COLOR-VALUE [color-only [exact-match | [endpoint-match [any | null]]]]

Parameters

- **COLOR-VALUE** A single policy color value, range ***0*** to ***4294967295***.
- **color-only** Allows configuration of color-only bits.
- **exact-match** Explicitly sets the color-only bits of the extended community to ***00*** (optional).
- **endpoint-match any** Sets the color-only bits of the extended community to ***10***.
- **endpoint-match null** Sets the color-only bits of the extended community to ***01***.

Example

```
switch(config)# route-map foo
switch(config0route-map-foo)# set community color 2 color-only endpoint-match
 any
```

10.1.13.8 dscp to traffic-class (DSCP map)

The **dscp to traffic-class** command configures the specified QoS map to map one or more DSCP classes to a traffic class. Only one traffic class may be specified per command. The command can be repeated to configure additional traffic classes. The configuration is modified immediately. Each **dscp to traffic-class** command overwrites the existing entries in the map.

The **default dscp to traffic-class** and **no dscp to traffic-class** commands restore the global map values for the given DSCP classes.

Command Mode

DSCP Map Configuration

Command Syntax

dscp ***dscp_classes*** **to traffic-class** ***traffic_class***

default dscp ***dscp_classes*** **to traffic-class**

no dscp ***dscp_classes*** **to traffic-class**

Parameters

- ***dscp_classes*** The DSCP class or classes to map to a new traffic-class value. These can be provided as a single value, a range given with a hyphen (such as 20-25), a comma separated list (such as 1,4,9), or a combination (such as 20-25, 35). The range for each value is 0-63.
- ***traffic_class*** The traffic class to map the specified DSCP class or classes to.

Example

These commands configure the DSCP-to-traffic-class map ***map1*** to map DSCP class ***35*** to traffic class ***7***, and DSCP classes ***20-25*** to traffic class ***6***.

```
switch(config)#qos map dscp to traffic-class name map1
switch(config-dscp-map-map1)#dscp 35 to traffic-class 7
switch(config-dscp-map-map1)#dscp 20-25 to traffic-class 6
```

10.1.13.9 hardware access-list qos resource sharing vlan in

The **`hardware access-list qos resource sharing vlan in`** command enables the ACL based QoS resources sharing on a VLAN interface.

The **`no hardware access-list qos resource sharing vlan in`** disables the ACL based QoS resources sharing on a VLAN interface. By default this function is disabled.

Command Mode

Global Configuration

Command Syntax

`hardware access-list qos resource sharing vlan in`

`no hardware access-list qos resource sharing vlan in`

Example

This commands enables the ACL based QoS resources sharing on a VLAN interface.

```
switch(config)# hardware access-list qos resource sharing vlan in
```

10.1.13.10 interface fabric (Trident II)

The **`interface fabric`** command places the switch in Fabric-interface configuration mode and allows the user to attach the QoS profile to the fabric interface of the switch.

Command Mode

Global Configuration

Command Syntax

`interface fabric`

Example

This command places the switch in Fabric-interface configuration mode.

```
switch(config)# interface fabric
switch(config-if-fabric)#
```

10.1.13.11 ip extcommunity-list

The **`ip extcommunity-list`** command adds a color extended community to an extcommunity-list. Multiple color extended communities can be added to the list. Negating the command removes the corresponding color extended community from the list.

Command Mode

Configuration mode

Command Syntax

`ip extcommunity-list` **WORD** [permit | deny][**COLOR-EXPRESSION**]

`no ip extcommunity-list` **WORD** [permit | deny] **COLOR-EXPRESSION**]

`default ip extcommunity-list` **WORD** [permit | deny][**COLOR-EXPRESSION**]

Parameters

- **WORD** Community list name.
- **permit** Specifies community to accept.
- **deny** Specifies comminity to reject.
- **COLOR-EXPRESSION** Color extended community.

Example

```
arista(config)# ip extcommunity-list foo permit color 1 color 2
arista(config)# ip extcommunity-list bar permit color 3 color-only endpoint-
match null color 4 color-only endpoint-match any
```

10.1.13.12 mc-tx-queue

The **`mc-tx-queue`** command places the switch in mc-tx-queue configuration mode to configure a multicast transmit queue on the configuration mode interface. Mc-tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **`exit`** command does not affect the configuration.

Trident and Tomahawk switches have four multicast queues (***MC0 – MC03***) and eight unicast queues (***UC0 – UC7***), categorized into two priority groups. All queues are exposed through the CLI and are user configurable.

- **Priority Group 1:** ***UC7***, ***UC6***, ***MC3***
- **Priority Group 0:** ***UC5***, ***UC4***, ***MC2***, ***UC3***, ***UC2***, ***MC1***, ***UC1***, ***UC0***, ***MC0***

The **`exit`** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **`no mc-tx-queue`** and **`default mc-tx-queue`** commands remove the configuration for the specified transmit queue by deleting the all corresponding **mc-tx-queue** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`mc-tx-queue` ***queue_level***

Parameters

queue_level The multicast transmit queue number. Values range from ***0*** to ***3***.

Commands Available in tx-queue Configuration Mode

- bandwidth percent (Trident and Tomahawk)
- priority (Trident and Tomahawk)
- shape rate (Tx-queue – Trident and Tomahawk)

Related Command

`uc-tx-queue` Configures unicast transmit queues on Trident and Tomahawk platform switches.

Example

This command enters mc-tx-queue configuration mode for ***multicast transmit queue 3*** of ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# mc-tx-queue 3
switch(config-if-Et5-mc-txq-3)#
```

10.1.13.13 platform petraA traffic-class

The **`platform petraA traffic-class`** command configures the default traffic class used by all ports on a specified chip. The default traffic class is implemented by Petra platform switches to replace qos cos and qos dscp commands. Traffic class values range from ***0*** to ***7***. The default traffic class is ***1***.

When **platform ?** returns **Petra**:

- **CoS trusted ports:** inbound untagged packets are assigned to the default traffic class. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- **DSCP trusted ports:** inbound non-IP packets are assigned to the default traffic class. IP packets are assigned to the traffic class that corresponds to the contents of its DSCP field.
- **Untrusted ports:** all inbound packets are assigned to the default traffic class.

The **`no platform petraA traffic-class`** and **`default platform petraA traffic-class`** commands restore the default traffic class of one for all ports on the specified chips by deleting the corresponding **`platform petraA traffic-class`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`platform petraA` [**CHIP_NAME**] traffic-class ***tc_value***

`no platform petraA` [**CHIP_NAME**] traffic-class

`default platform petraA` [**CHIP_NAME**] traffic-class

Parameters

CHIP_NAME Trust mode assigned to the specified ports. Port designation options include:

- ***no parameter*** All ports on the switch.
- **module *cardX*** All ports on specified linecard (7500 Series).
- **petra *cardX* / *chipY*** All ports on PetraA chip ***chipY*** on linecard ***cardX*** (7500 Series).
- **petra *-chipZ*** All ports on PetraA chip chipZ (7048 Series)

7500 Series

Switches can contain up to eight linecards. ***CardX*** varies from ***3*** to ***10***.

Each linecard contains six PetraA chips. Each chip controls eight ports. ***ChipY*** varies from ***0*** to ***5***:

- ***0*** controls ports ***1*** through ***8***:
  - ***1*** controls ports ***9*** through ***16***.
  - ***2*** controls ports ***17*** through ***24***.
  - ***3*** controls ports ***25*** through ***32***.
  - ***4*** controls ports ***33*** through ***40***.
  - ***5*** controls ports ***41*** through ***48***.

7048 Series

Each switch contains two PetraA chips. ***ChipZ*** varies from ***0*** to ***1***:

- ***0*** controls ports ***1*** through ***32***.
  - ***1*** controls ports ***33*** through ***52***.
- ***tc_value*** Traffic class value. Values range from ***0*** to ***7***. Default value is ***1***.

Related Command

show platform petraA traffic-class displays the traffic class assignment on all specified Petra chips.

Example

This command configures the default ***traffic class 6*** for ports ***25-32*** on linecard ***5***.

```
switch(config)# platform petraA petra5/3 traffic-class 6
switch(config)#
```

10.1.13.14 priority (Arad/Jericho)

The **`priority`** command specifies the priority of the transmit queue. The switch supports two queue priorities:

- **strict priority:** contents are removed from the queue - subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all queues is strict priority.
- **round robin priority:** contents are removed proportionately from all round robin queues - subject to maximum bandwidth limits assigned to the strict priority queues.

Tx-queue 7 is set to strict priority and is not configurable.

When a queue is configured as a round robin queue, all lower priority queues also function as round robin queues. A queue's numerical label denotes its priority: higher labels denote higher priority. ***Tx-queue 6*** has higher priority than ***Tx-queue 5***, and ***Tx-queue 0*** has the lowest priority.

The **`priority strict`** and **`default priority`** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **`no priority`** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Tx-Queue Configuration

Command Syntax

`priority strict`

`no priority`

`default priority`

Related Command

tx-queue (Arad/Jericho) places the switch in tx-queue configuration mode.

Example

These commands perform the following on ***interface ethernet 3/4/1***:

- Displays the default state of all transmit queues.
- Configures ***transmit queue 3*** as a round robin queue.

- Displays the effect of the **`no priority`** command on all transmit queues on the interface.

```
switch(config)# interface ethernet 3/4/1
switch(config-if-Et3/4/1)# show qos interfaces ethernet 3/4/1

Ethernet3/4/1:

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -      999 / 1000 ( Mbps )   SP / SP    D
   3      - / -      999 / 1000 ( Mbps )   SP / SP    D
   2      - / -        - / -    (  -  )    SP / SP    D
   1      - / -        - / -    (  -  )    SP / SP    D
   0      - / -        - / -    (  -  )    SP / SP    D

Note: Values are displayed as Operational/Configured

switch(config-if-Et3/4/1)# tx-queue 3
switch(config-if-Et3/4/1-txq-3)# no priority
switch(config-if-Et3/4/1-txq-3)# show qos interfaces ethernet 3/4/1

Ethernet3/4/1:

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -      999 / 1000 ( Mbps )   SP / SP    D
   3     25 / -      999 / 1000 ( Mbps )   RR / RR    D
   2     25 / -        - / -    (  -  )    RR / SP    D
   1     25 / -        - / -    (  -  )    RR / SP    D
   0     25 / -        - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured

switch(config-if-Et3/4/1-txq-3)#
```

10.1.13.15 priority (FM6000)

The **`priority`** command specifies the priority of the transmit queue. The switch supports two queue priorities:

- **strict priority:** contents are removed from the queue - subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all queues is strict priority.
- **round robin priority:** contents are removed proportionately from all round robin queues - subject to maximum bandwidth limits assigned to the strict priority queues.

When a queue is configured as a round robin queue, all lower priority queues also function as round robin queues. A queue's numerical label denotes its priority: higher labels denote higher priority. ***Tx-queue 6*** has higher priority than ***Tx-queue 5***, and ***Tx-queue 0*** has the lowest priority.

The **`priority strict`** and **`default priority`** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Tx-Queue Configuration

Command Syntax

```
priority strict

no priority

default priority
```

Related Command

tx-queue (FM6000) places the switch in tx-queue configuration mode.

Example

These commands perform the following on ***interface ethernet 19***:

- Displays the default state of all transmit queues.
- Configures ***transmit queue 3*** as a round robin queue.
- Displays the effect of the **no priority** command on all transmit queues on the interface.

```
switch(config)# interface ethernet 19
switch(config-if-Et19)# show qos interface ethernet 19

Ethernet19:
   Trust Mode: COS

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3         N/A     disabled        strict
          2         N/A     disabled        strict
          1         N/A     disabled        strict
          0         N/A     disabled        strict

switch(config-if-Et19)# tx-queue 3
switch(config-if-Et19-txq-3)# no priority
switch(config-if-Et19-txq-3)# show qos interface ethernet 19

Ethernet19:
   Trust Mode: COS

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          25     disabled   round-robin
          2          25     disabled   round-robin
          1          25     disabled   round-robin
          0          25     disabled   round-robin
```

```
switch(config-if-Et19-txq-3)#
```

10.1.13.16 priority (Petra)

The **priority** command specifies the priority of the transmit queue. The switch supports two queue priorities:

- **strict priority:** contents are removed from the queue - subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all queues is strict priority.
- **round robin priority:** contents are removed proportionately from all round robin queues - subject to maximum bandwidth limits assigned to the strict priority queues.

Tx-queue 7 is set to strict priority and is not configurable.

When a queue is configured as a round robin queue, all lower priority queues also function as round robin queues. A queue's numerical label denotes its priority: higher labels denote higher priority. ***Tx-queue 6*** has higher priority than ***Tx-queue 5***, and ***Tx-queue 0*** has the lowest priority.

The **priority strict** and **default priority** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Tx-Queue Configuration

Command Syntax

priority strict

no priority

default priority

Related Command

tx-queue (Petra) places the switch in tx-queue configuration mode.

Example

These commands perform the following on ***Ethernet interface 3/28***:

- Displays the default state of all transmit queues.
- Configures ***transmit queue 3*** as a round robin queue.
- Displays the effect of the **no priority** command on all transmit queues on the interface.

```
switch(config)# interface ethernet 3/28
switch(config-if-Et3/28)# show qos interface ethernet 3/28

Ethernet3/28:
   Trust Mode: COS

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3         N/A     disabled        strict
          2         N/A     disabled        strict
```

```
          1         N/A     disabled        strict
          0         N/A     disabled        strict

switch(config-if-Et3/28)# tx-queue 3
switch(config-if-Et3/28-txq-3)# no priority
switch(config-if-Et3/28-txq-3)# show qos interface ethernet 3/28

Ethernet3/28:
   Trust Mode: COS

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          25     disabled   round-robin
          2          25     disabled   round-robin
          1          25     disabled   round-robin
          0          25     disabled   round-robin

switch(config-if-Et3/28-txq-3)#
```

10.1.13.17 priority (Trident and Tomahawk)

The **priority** command specifies the priority of the transmit queue. The switch supports two queue priorities:

- **strict priority:** contents are removed from the queue - subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all other queues is strict priority.
- **round robin priority:** contents are removed proportionately from all round robin queues - subject to maximum bandwidth limits assigned to the strict priority queues.

Trident and Tomahawk switches have eight unicast queues (***UC0 – UC7***) and four multicast queues (M***C0 – MC03***), categorized into two priority groups. ***Priority group 1*** queues have priority over ***priority 0*** queues. The following lists display the priority group queues in order from higher priority to lower priority.

- **Priority Group 1:** ***UC7***, ***UC6***, ***MC3***
- **Priority Group 0:** ***UC5***, ***UC4***, ***MC2***, ***UC3***, ***UC2***, ***MC1***, ***UC1***, ***UC0***, ***MC0***

Priority group 1 queues are strict priority queues and are not configurable as round robin. ***Priority 0*** queues are strict priority by default and are configurable as round robin. When a queue is configured as a round robin queue, all lower priority queues automatically function as round robin queues.

The **priority strict** and **default priority** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Mc-Tx-Queue configuration

Uc-Tx-Queue configuration

Command Syntax

priority strict

no priority

`default priority`

Related Commands

- mc-tx-queue places the switch in mc-tx-queue configuration mode.
- uc-tx-queue places the switch in uc-tx-queue configuration mode.

Example

These commands perform the following on ***interface ethernet 7***:

- Displays the default state of all transmit queues.
- Configures ***transmit queue 3*** as a round robin queue.
- Displays the effect of the **`no priority`** command on all transmit queues on the interface.

```
switch(config) #interface ethernet 7
switch(config-if-Et7)# show qos interface ethernet 7

Ethernet7:
   Trust Mode: COS

   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3         N/A     disabled        strict                0
        UC2         N/A     disabled        strict                0
        MC1         N/A     disabled        strict                0
        UC1         N/A     disabled        strict                0
        UC0         N/A     disabled        strict                0
        MC0         N/A     disabled        strict                0

switch(config-if-Et7)# uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)# no priority
switch(config-if-Et7-uc-txq-3)# show qos interface ethernet 7

Ethernet7:
   Trust Mode: COS

   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3          20     disabled   round-robin                0
        UC2          16     disabled   round-robin                0
        MC1          16     disabled   round-robin                0
        UC1          16     disabled   round-robin                0
        UC0          16     disabled   round-robin                0
        MC0          16     disabled   round-robin                0

switch(config-if-Et7-uc-txq-3)#
```

10.1.13.18 qos cos

The **`qos cos`** command specifies the default class of service (CoS) value of the configuration mode interface. CoS values range from ***0*** to ***7***. Default value is ***0***.

Arad, Jericho, fm6000, Trident, Tomahawk, and **Trident II** platform switches:

- CoS trusted ports: the default CoS value determines the traffic class for inbound untagged packets. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- Untrusted ports: the default CoS value determines the traffic class for all inbound packets.

Petra platform switches:

- CoS trusted ports: inbound untagged packets are assigned to the default traffic class, as configured by platform petraA traffic-class. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- Untrusted ports: all inbound packets are assigned to the default traffic class.

The **`no qos cos`** and **`default qos cos`** commands restore the port's default CoS value to zero by deleting the corresponding **`qos cos`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`qos cos` ***cos_value***

`no qos cos`

`default qos cos`

Parameters

cos_valueCoS value assigned to port. Value ranges from ***0*** to ***7***. Default value is ***0***.

Example

This command configures the default ***CoS 4*** on ***interface ethernet 8***.

```
switch(config-if-Et8)# qos cos 4
switch(config-if-Et8)#
```

10.1.13.19 qos dscp

The **`qos dscp`** command specifies the default Differentiated Services Code Point (DSCP) value of the configuration mode interface. The default DSCP determines the traffic class for non-IP packets that are inbound on DSCP trusted ports. DSCP trusted ports determine the traffic class for inbound packets as follows:

- **Arad, Jericho, fm6000, Trident, Tomahawk,** and **Trident II** platform switches:
  - non-IP packets: default DSCP value specified by **`qos dscp`** determines the traffic class.
  - IP packets: assigned to the traffic class corresponding to its DSCP field contents.
- **Petra** platform switches:
  - non-IP packets: assigned to default traffic class configured by platform petraA traffic-class.
  - IP packets: assigned to the traffic class corresponding to its DSCP field contents.

The **`no qos dscp`** and **`default qos dscp`** commands restore the port's default DSCP value to zero by deleting the corresponding **`qos dscp`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`qos dscp` ***dscp_value***

`no qos dscp`

`default qos dscp`

Parameters

dscp_value DSCP value assigned to the port. Value ranges from ***0*** to ***63***. Default value is ***0***.

Example

This command sets the default DSCP of ***44*** on ***interfacee thernet 7***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# qos dscp 44
switch(config-if-Et7)
```

10.1.13.20 qos map cos

The **`qos map cos`** command associates a traffic class to a list of class of service (CoS) settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

The **`no qos map cos`** and **`default qos map cos`** commands restore the specified CoS values to their default traffic class setting by deleting the corresponding **`qos map cos`** statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`qos map cos` ***cos_value_1*** [***cos_value_2 ... cos_value_n***] **to traffic-class** ***tc_value***

`no qos map cos` ***cos_value_1*** [***cos_value_2 ... cos_value_n***]

`default qos map cos` ***cos_value_1*** [***cos_value_2 ... cos_value_n***]

Parameters

- ***cos_value_x*** Class of Service (CoS) value. Value ranges from ***0*** to ***7***.
- ***tc_value*** Traffic class value. Value range varies by platform.

Default CoS to traffic class map varies by platform (Table 40: Default CoS to Traffic Class Map).

Default Inbound CoS to Traffic Class Map

Table 40: Default CoS to Traffic Class Map displays the default CoS to traffic class map for each platform.

Table 40: Default CoS to Traffic Class Map

Inbound CoS	untagged	0	1	2	3	4	5	6	7
Traffic Class (Arad / Jericho)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7
Traffic Class (FM6000)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7
Traffic Class (Helix)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7
Traffic Class (Petra)	Assigned default traffic class	1	0	2	3	4	5	6	7
Traffic Class (Trident and Tomahawk)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7
Traffic Class (Trident II)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7

Related Commands

- qos cos specifies the default CoS.
- platform petraA traffic-class specifies the default traffic class.

Example

This command assigns the ***traffic class 5*** to the classes of service ***1***, ***3***, ***5***, and ***7***.

```
switch(config)# qos map cos 1 3 5 7 to traffic-class 5
switch(config)#
```

10.1.13.21 qos map dscp

The **qos map dscp** command associates a traffic class to a set of Differentiated Services Code Point (DSCP) values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

This command configures the global DSCP to traffic-class map. To create additional named maps that can be attached to specific VRFs, use the **qos map dscp to traffic-class** command with the **name** option.

The **no qos map dscp** and **default qos map dscp** commands restore the specified DSCP values to their default traffic class settings by deleting corresponding **qos map dscp** statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

qos map dscp ***dscpv_1*** [***dscpv_2...dscpv_n***] **to traffic-class** ***tc_value***

no qos map dscp ***dscpv_1*** [***dscpv_2...dscpv_n***]

default qos map dscp ***dscpv_1*** [***dscpv_2...dscpv_n***]

Parameters

- ***dscpv_x*** Differentiated Services Code Point (DSCP) value. Value ranges from ***0*** to ***63***.

- ***tc_value*** Traffic class value. Value range varies by platform.

Default map varies by platform (Table 41: Default DSCP to Traffic Class Map).

Default Inbound DSCP to Traffic Class Map

Table 41: Default DSCP to Traffic Class Map displays the default DSCP to traffic class map for each platform.

Table 41: Default DSCP to Traffic Class Map

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class (Arad / Jericho)	1	0	2	3	4	5	6	7
Traffic Class (FM6000)	1	0	2	3	4	5	6	7
Traffic Class (Helix)	1	0	2	3	4	5	6	7
Traffic Class (Petra)	1	0	2	3	4	5	6	7
Traffic Class (Trident and Tomahawk)	1	0	2	3	4	5	6	7
Traffic Class (Trident II)	1	0	2	3	4	5	6	7

Example

This command assigns the ***traffic class 3*** to the DSCP values of ***12***, ***13***, ***25***, and ***37***.

```
switch(config)# qos map dscp 12 13 25 37 to traffic-class 3
switch(config)#
```

10.1.13.22 qos map dscp to traffic-class

The **qos map dscp to traffic-class** command puts the switch in DSCP Map Configuration mode for the specified QoS map. If the map does not already exist, it is created as a copy of the global DSCP-to-traffic-class map. In DSCP Map Configuration mode, the **dscp to traffic-class** command is available to make changes to the map.

The **no** and **default** forms of the command remove the specified custom map.

Command Mode

Global Configuration

Command Syntax

qos map dscp to traffic-class name ***map_name***

no qos map dscp to traffic-class name ***map_name***

default qos map dscp to traffic-class name ***map_name***

Parameters

- ***map_name*** The name of the map to configure. If the map does not exist, it is created as a copy of the global DSCP-to-traffic-class map.

Example

This command creates the map ***map1*** and places the switch in DSCP Map Configuration mode.

```
switch(config)#qos map dscp to traffic-class name map1
switch(config-dscp-map-map1)#
```

10.1.13.23 qos map dscp to traffic-class (MPLS tunnel termination VRF)

The **`qos map dscp to traffic-class`** command assigns a DSCP-to-traffic-class map to a VRF, replacing the global map or previous custom map. The switch uses this map to assign a traffic class to data packets routed to this VRF on the basis of the DSCP fields of these packets.

The **`no qos map dscp to traffic-class`** and **`default qos map dscp to traffic-map`** commands remove the assignment of a custom map from the VRF, restoring the global map. default traffic class settings by deleting corresponding **qos map dscp** statements from ***running-config***.

Command Mode

MPLS Tunnel Termination VRF Configuration

Command Syntax

`qos map dscp to traffic-class` ***map_name***

`no qos map dscp to traffic-class` ***map_name***

`default qos map dscp to traffic-class` ***map_name***

Parameter

- ***map_name*** DSCP to traffic-class map name.

Example

These commands assign the ***map1*** QoS DSCP-to-traffic-class map to the ***newVRF1*** and ***newVRF2*** VRFs.

```
switch(config)#mpls tunnel termination
switch(config-mpls-tunnel-termination)#vrf newVRF1
switch(config-tunnel-termination-vrf-newVRF1)#qos map dscp to traffic-
class map1
switch(config-tunnel-termination-vrf-newVRF1)#exit
switch(config-mpls-tunnel-termination)#vrf newVRF2
switch(config-tunnel-termination-vrf-newVRF2)#qos map dscp to traffic-
class map1
switch(config-tunnel-termination-vrf-newVRF2)#exit
switch(config-tunnel-termination)#
```

10.1.13.24 qos map traffic-class to cos

The **`qos map traffic-class to cos`** command associates a class of service (CoS) to a list of traffic classes. Multiple commands create a complete traffic class to CoS map. The switch uses this map in CoS rewrite operations to fill the CoS field in outbound packets. This map is applicable to DSCP trusted ports and untrusted ports. CoS rewrite is disabled on CoS trusted ports. The show qos maps command displays the CoS to traffic class map.

The **`no qos traffic-class to cos`** and **`default qos traffic-class to cos`** commands restore the specified traffic class values to their default CoS settings by removing the corresponding **`qos map traffic-class to cos`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*] `to cos`** ***cos_value***

`no qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*] `to cos`**

`default qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*] `to cos`**

Parameters

- ***tc_num_x*** Traffic class value. Value range varies by switch platform.
- ***cos_value*** Class of Service (CoS) value. Value ranges from ***0*** to ***7***.

Default Inbound Traffic Class to CoS Map

Table 42: Default Traffic Class to CoS Rewrite Value Map displays the default traffic class to CoS map for each platform.

Table 42: Default Traffic Class to CoS Rewrite Value Map

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value (Arad and / Jericho)	1	0	2	3	4	5	6	7
CoS Rewrite Value (FM6000)	1	0	2	3	4	5	6	7
CoS Rewrite Value (Helix)	1	0	2	3	4	5	6	7
CoS Rewrite Value (Petra)	1	0	2	3	4	5	6	7
CoS Rewrite Value (Trident and Tomahawk)	1	0	2	3	4	5	6	7
CoS Rewrite Value (Trident II)	1	0	2	3	4	5	6	7

Example

This command assigns the ***CoS 2*** to traffic classes ***1***, ***3***, and ***5***.

```
switch(config)# qos map traffic-class 1 3 5 to cos 2
switch(config)#
```

10.1.13.25 qos map traffic-class to dscp

The **`qos map traffic-class to dscp`** command associates a Differentiated Services Code Point (DSCP) value to a list of traffic classes. Multiple commands create a complete traffic class to DSCP map. The switch uses this map in DSCP rewrite operations to fill the DSCP field in outbound packets. This map is applicable to CoS trusted ports and untrusted ports but disabled by default on these ports. DSCP rewrite is disabled on DSCP trusted ports. The show qos maps command displays the traffic class to DSCP map.

The **`no qos traffic-class to dscp`** and **`default qos traffic-class to dscp`** commands restore the specified traffic class values to their default DSCP settings by removing the corresponding **`qos map traffic-class to dscp`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

qos map traffic-class ***tc_num_1*** **[***tc_num_2 ... tc_num_n***] to dscp** ***dscp_value***

no qos map traffic-class ***tc_num_1*** **[***tc_num_2 ... tc_num_n***] to dscp**

default qos map traffic-class ***tc_num_1*** **[***tc_num_2 ... tc_num_n***] to dscp**

Parameters

- ***tc_num_x*** Traffic class value. Value range varies by switch platform.
- ***dscp_value*** Differentiated Services Code Point (DSCP) value. Value ranges from ***0*** to ***63***.

Default Inbound Traffic Class to DSCP Map

Table 43: Default Traffic Class to DSCP Rewrite Value Map displays the default traffic class to DSCP map for each platform.

Table 43: Default Traffic Class to DSCP Rewrite Value Map

Traffic Class	0	1	2	3	4	5	6	7
DSCP Rewrite Value (FM6000)	8	0	16	24	32	40	48	56
DSCP Rewrite Value (Helix)	8	0	16	24	32	40	48	56
DSCP Rewrite Value (Trident and Tomahawk)	8	0	16	24	32	40	48	56
DSCP Rewrite Value (Trident II)	8	0	16	24	32	40	48	56

Example

This command assigns the DSCP value of ***17*** to traffic classes ***1***, ***2***, and ***4***.

```
switch(config)# qos map traffic-class 1 2 4 to dscp 17
switch(config)#
```

10.1.13.26 qos map traffic-class to mc-tx-queue

The **qos map traffic-class to mc-tx-queue** command associates a multicast transmit queue to a list of traffic classes. Multiple commands create a complete traffic class to mc-tx-queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch. The show qos maps command displays the traffic class to multicast transmit queue map.

The **no qos traffic-class to mc-tx-queue** and **default qos traffic-class to mc-tx-queue** commands restore the default traffic class to multicast transmit queue map for the specified traffic class values by removing the corresponding **qos map traffic-class to mc-tx-queue** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

qos map traffic-class ***tc_num_1*** **[***tc_num_2 ... tc_num_n***] to mc-tx-queue** ***mtq_value***

no qos map traffic-class ***tc_num_1*** **[***tc_num_2 ... tc_num_n***] to mc-tx-queue**

default qos map traffic-class ***tc_num_1*** **[***tc_num_2 ... tc_num_n***] to mc-tx-queue**

Parameters

- ***tc_num_x*** Traffic class value. Value ranges from ***0*** to ***7***.
- ***mtq_value*** Multicast transmit queue number. Value ranges from ***0*** to ***3***.

Default Inbound Traffic Class to Multicast Transmit Queue Map

Table 44: Default Traffic Class to Multicast Transmit Queue Map displays the default traffic class to multicast transmit queue map for Trident and Tomahawk platform switches.

Table 44: Default Traffic Class to Multicast Transmit Queue Map

Traffic Class	0	1	2	3	4	5	6	7
Multicast Transmit Queue (Trident and Tomahawk)	0	0	1	1	2	2	3	3

Related Commands

- qos map traffic-class to uc-tx-queue (Trident and Tomahawk) associates traffic classes to a multicast transmit queue.
- qos map traffic-class to tx-queue (all other platforms) associates traffic classes to a transmit queue.

Example

This command maps traffic classes ***0***, ***4***, and ***5*** to ***mc-tx-queue 2***.

```
switch(config)# qos map traffic-class 0 4 5 to mc-tx-queue 2
switch(config)#
```

10.1.13.27 qos map traffic-class to tx-queue

The **`qos map traffic-class to tx-queue`** command associates a transmit queue (tx-queue) to a list of traffic classes. Multiple commands create a complete traffic to tx-queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch. The show qos maps command displays the transmit queue to traffic class map.

The **`no qos traffic-class to tx-queue`** and **`default qos traffic-class to tx-queue`** commands restore the specified traffic class values to their default transmit queue settings by removing the corresponding **`qos map traffic-class to tx-queue`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*] `to tx-queue`** ***txq_value***

`no qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*] `to tx-queue`**

`default qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*] `to tx-queue`**

Parameters

- ***tc_num_x*** Traffic class value. Value range varies by platform.
- ***txq_value*** Transmit queue value. Value range varies by platform.

Restrictions

FM6000: When Priority Flow Control (PFC) is enabled, traffic classes are mapped to their corresponding transmit queues, regardless of existing **`qos map traffic-class to tx-queue`** statements.

Arad, Jericho, and Petra: ***Traffic class 7*** always maps to ***transmit queue 7***. This association is not editable.

Default Inbound Traffic Class to Transmit Queue Map

Table 45: Default Traffic Class to Transmit Queue Map displays the transmit queue to traffic class map.

Table 45: Default Traffic Class to Transmit Queue Map

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue (Arad / Jericho)	0	1	2	3	4	5	6	7
Transmit Queue (FM6000)	0	1	2	3	4	5	6	7
Transmit Queue (Helix)	0	1	2	3	4	5	6	7
Transmit Queue (Petra)	0	1	2	3	4	5	6	7
Transmit Queue (Trident II)	0	1	2	3	4	5	6	7

Related Commands

- qos map traffic-class to mc-tx-queue (Trident and Tomahawk) associates traffic classes to a unicast transmit queue.
- qos map traffic-class to uc-tx-queue (Trident and Tomahawk) associates traffic classes to a multicast transmit queue.

Example

This command maps traffic classes ***0***, ***4***, and ***5*** to ***tx-queue 4***.

```
switch(config)# qos map traffic-class 0 4 5 to tx-queue 4
switch(config)#
```

10.1.13.28 qos map traffic-class to uc-tx-queue

The **`qos map traffic-class to uc-tx-queue`** command associates a unicast transmit queue to a list of traffic classes. Multiple commands create a complete traffic class to unicast transmit queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch. The show qos maps command displays the traffic class to unicast transmit queue map.

The **`no qos traffic-class to uc-tx-queue`** and **`default qos traffic-class to uc-tx-queue`** commands restore the default traffic class to unicast transmit queue map for the specified traffic class values by removing the corresponding **`qos map traffic-class to uc-tx-queue`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*]** **`to uc-tx-queue`** ***utq_value***

`no qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*]** **`to uc-tx-queue`**

`default qos map traffic-class` ***tc_num_1*** **[*tc_num_2 ... tc_num_n*]** **`to uc-tx-queue`**

Parameters

- ***tc_num_x*** Traffic class value. Value ranges from ***0*** to ***7***.
- ***utq_value*** Unicast transmit queue number. Value ranges from ***0*** to ***7***.

Default Inbound Traffic Class to Unicast Transmit Queue Map

Table 46: Default Traffic Class to Unicast Transmit Queue Map displays the default traffic class to Unicast transmit queue map for Trident and Tomahawk platform switches.

Table 46: Default Traffic Class to Unicast Transmit Queue Map

Traffic Class	0	1	2	3	4	5	6	7
Unicast Transmit Queue (Trident and Tomahawk)	0	1	2	3	4	5	6	7

Related Commands

- qos map traffic-class to mc-tx-queue (Trident and Tomahawk) associates traffic classes to a unicast transmit queue.
- qos map traffic-class to tx-queue (all other platforms) associates traffic classes to a transmit queue.

Example

This command maps traffic classes ***0***, ***4***, and ***5*** to ***unicast transmit queue 4***.

```
switch(config)# qos map traffic-class 0 4 5 to uc-tx-queue 4
switch(config)#
```

10.1.13.29 qos profile

The **`qos profile`** command places the switch in QoS profile configuration mode and for the specified profile and creates the profile if it does not already exist. QoS profiles are used to apply the same QoS configuration to multiple interfaces.

The **`no qos profile`** and **`default qos profile`** command deletes the QoS profile from the running configuration.

The **`exit`** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

`qos profile` ***profile_name***

`no qos profile` ***profile_name***

`default qos profile` ***profile_name***

Parameter

profile_name QoS profile name.

Note: Commands use a subset of the listed fields. Available subset depends on the specified parameter. Use CLI syntax assistance to view options for specific parameter when creating a QoS profile.

Example

This command places the switch in QoS profile configuration mode for policy map policy map ***TP*** and creates the policy map if it does not already exist.

```
switch(config)# qos profile TP
switch(config-qos-profile-TP)#
```

10.1.13.30 qos random-detect ecn allow non-ect chip-based (Tomahawk and Trident)

The **`qos random-detect ecn allow non-ect chip-based`** enables per color queue thresholds using color based queue thresholds and drop-precedence values along with drop of non-ect traffic by allowing non-ect and set drop-precedence 1 in a policy map simultaneously.

The **`no qos random-detect ecn allow non-ect chip-based`** and **`default qos random-detect ecn allow non-ect chip-based`** commands disbales the use of non-ect and set drop-precedence 1 simultaneously in a policy map in the **`running-config`**.

Command Mode

Global Configuration

Command Syntax

`qos random-detect ecn allow non-ect chip-based`

`no qos random-detect ecn allow non-ect chip-based`

`default qos random-detect ecn allow non-ect chip-based`

Example

The following command enables the use of non-ect and ***set drop-precedence 1*** simultaneously in a policy map.

```
switch(config)# qos random-detect ecn allow non-ect chip-based
```

10.1.13.31 qos random-detect ecn global-buffer (Helix)

The **`qos random-detect ecn global-buffer`** command enables ECN marking for globally shared packet memory and specifies minimum and maximum queue threshold sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- **00** Non ECN Capable transport.
- **10** ECN Capable transport.
- **01** ECN Capable transport.
- **11** Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked **`congestion encountered`**.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked **`congestion encountered`**. The proportion of marked packets varies linearly with average queue size:
  - 0% are marked when average queue size is less than or equal to minimum threshold.
  - 100% are marked when average queue size is greater than or equal to maximum threshold.

  When transmitted packets are marked **`Non ECN Capable`**, congestion packets are dropped, not marked.

The **no qos random-detect ecn global-buffer** and **default qos random-detect ecn global-buffer** commands disables ECN marking for the shared buffer by removing the **qos random-detect ecn global-buffer** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

qos random-detect ecn global-buffer minimum-threshold **MIN** maximum-threshold **MAX**

no qos random-detect ecn global-buffer

default qos random-detect ecn global-buffer

Guidelines

Packet memory is divided into 46080 208-byte cells, whose allocation is managed by the memory management unit (MMU). The MMU tracks the cells that each entity uses and determines the number of cells that can be allocated to an entity.

Parameters

MIN and **MAX** parameters must use the same data unit.

- **MIN** Minimum threshold. Options include:
  - ***1*** to ***19456*** **segments** 208-byte segments units.
  - ***1*** to ***4*** **mbytes** Megabyte units.
  - ***1*** to ***4046*** **kbytes** Kilobyte units.
  - ***1*** to ***4046848*** **bytes** Byte units.
- **MAX** Maximum threshold. Options include:
  - ***1*** to ***19456*** **segments** 208-byte segments units.
  - ***1*** to ***4*** **mbytes** Megabyte units.
  - ***1*** to ***4046*** **kbytes** Kilobyte units.
  - ***1*** to ***4046848*** **bytes** Byte units.

Related Command

random-detect ecn (Helix) enables ECN marking for a unicast transmit queue.

Examples

- This command enables ECN marking of unicast packets from the global data pool and sets the minimum and maximum thresholds at ***20*** and ***500*** segments.

```
switch(config)# qos random-detect ecn global-buffer minimum-thres
hold 20 segments
maximum-threshold 500 segments
switch(config)#
```

- This command disables ECN marking of unicast packets from the global data pool.

```
switch(config)# no qos random-detect ecn global-buffer
switch(config)#
```

10.1.13.32 qos random-detect ecn global-buffer (Trident and Tomahawk)

The **qos random-detect ecn global-buffer** command enables ECN marking for globally shared packet memory and specifies minimum and maximum queue threshold sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- **00** Non ECN Capable transport.
- **10** ECN Capable transport.
- **01** ECN Capable transport.
- **11** Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked **congestion encountered**.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked **congestion encountered**. The proportion of marked packets varies linearly with average queue size:
  - 0% are marked when average queue size is less than or equal to minimum threshold.
  - 100% are marked when average queue size is greater than or equal to maximum threshold.

  When transmitted packets are marked **Non ECN Capable**, congestion packets are dropped, not marked.

The **no qos random-detect ecn global-buffer** and **default qos random-detect ecn global-buffer** commands disables ECN marking for the shared buffer by removing the **qos random-detect ecn global-buffer** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

qos random-detect ecn global-buffer minimum-threshold **MIN** maximum-threshold **MAX**

no qos random-detect ecn global-buffer

default qos random-detect ecn global-buffer

Guidelines

Packet memory is divided into 46080 208-byte cells, whose allocation is managed by the memory management unit (MMU). The MMU tracks the cells that each entity uses and determines the number of cells that can be allocated to an entity.

Parameters

MIN and **MAX** parameters must use the same data unit.

- **MIN** Minimum threshold. Options include:
  - ***1*** to ***46080*** **segments** 208-byte segments units.
  - ***1*** to ***9*** **mbytes** Megabyte units.
  - ***1*** to ***9584*** **kbytes** Kilobyte units.
  - ***1*** to ***9584640*** **bytes** Byte units.
- **MAX** Maximum threshold. Options include:
  - ***1*** to ***46080*** **segments** 208-byte segments units.
  - ***1*** to ***9*** **mbytes** Megabyte units.
  - ***1*** to ***9584*** **kbytes** Kilobyte units.
  - ***1*** to ***9584640*** **bytes** Byte units.

Related Command

random-detect ecn (Trident and Tomahawk) enables ECN marking for a unicast transmit queue.

Examples

- This command enables ECN marking of unicast packets from the global data pool and sets the minimum and maximum thresholds at ***20*** and ***500*** segments.

```
switch(config)# qos random-detect ecn global-buffer minimum-thres
hold 20 segments maximum-threshold 500 segments
switch(config)#
```

- This command disables ECN marking of unicast packets from the global data pool.

```
switch(config)# no qos random-detect ecn global-buffer
switch(config)#
```

10.1.13.33 qos rewrite cos

The **`qos rewrite cos`** command enables the rewriting of the CoS field for outbound tagged packets that were received on DSCP trusted ports and untrusted ports. CoS rewrite is always disabled on CoS trusted ports. The CoS value that is written into the packet is based on the data stream's traffic class. CoS rewriting is active by default.

The **`no qos rewrite cos`** command disables CoS rewriting on the switch. The default **qos rewrite cos** command restores the default setting of enabling CoS rewriting by removing the **`no qos rewrite cos`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`qos rewrite cos`

`no qos rewrite cos`

`default qos rewrite cos`

Related Command

qos map traffic-class to cos configures the traffic class to CoS rewrite map.

Example

This command enables CoS rewrite.

```
switch(config)# qos rewrite cos
switch(config)#
```

10.1.13.34 qos rewrite dscp

The **`qos rewrite dscp`** command enables the rewriting of the DSCP field for outbound tagged packets that were received on CoS trusted ports and untrusted ports. DSCP rewrite is always disabled on DSCP trusted ports. The DSCP value that is written into the packet is based on the data stream's traffic class. DSCP rewriting is disabled by default.

The **`no qos rewrite dscp`** and **`default qos rewrite dscp`** commands disable DSCP rewriting on the switch by removing the **`no qos rewrite dscp`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`qos rewrite dscp`

`no qos rewrite dscp`

`default qos rewrite dscp`

Related Command

qos map traffic-class to dscp configures the traffic class to DSCP rewrite map.

Example

This command enables DSCP rewrite.

```
switch(config)# qos rewrite dscp
switch(config)#
```

10.1.13.35 qos trust

The **`qos trust`** command configures the quality of service port trust mode for the configuration mode interface. Trust-enabled ports classify traffic by examining the traffic's CoS or DSCP value. Port trust mode default setting is **cos** for switched interfaces and **dscp** for routed interfaces.

The **`default qos trust`** command restores the default trust mode on the configuration mode interface by removing the corresponding **`qos trust`** or **`no qos trust`** statement from ***running-config***.

The **`no qos trust`** command performs the following:

- **`no qos trust`** places the port in ***untrusted*** mode.
- **`no qos trust cos`** removes the corresponding **`qos trust cos`** statement.
- **`no qos trust dscp`** removes the corresponding **`qos trust dscp`** statement.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`qos trust` [**MODE**]

`no qos trust` [**MODE**]

`default qos trust`

Parameters

- ***MODE*** Trust mode assigned to the port. Options include:
  - **`cos`** Enables cos trust mode.
  - **`dscp`** Enables dscp trust mode.
- **`no qos trust`** Enables untrusted mode on the port.

Examples

- This command configures trust mode of dscp for ***interface ethernet 7***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# qos trust dscp
switch(config-if-Et7)# show active
interface Ethernet7
   qos trust dscp
switch(config-if-Et7)#
```

- This command configures trust mode of untrusted for ***Port Channel interface 23***.

```
switch(config)# interface port-channel 23
switch(config-if-Po23)# no qos trust
switch(config-if-Po23)# show active
interface Port-Channel23
   no qos trust
switch(config-if-Po23)#
```

10.1.13.36 random-detect ecn (Arad/Jericho)

The **random-detect ecn** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- **00** Non ECN Capable transport.
- **10** ECN Capable transport.
- **01** ECN Capable transport.
- **11** Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked **congestion encountered**.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked **congestion encountered**. The proportion of marked packets varies linearly with average queue size:
  - 0% are marked when average queue size is less than or equal to minimum threshold.
  - 100% are marked when average queue size is greater than or equal to maximum threshold.

  When transmitted packets are marked **Non ECN Capable**, congestion packets are dropped, not marked.

The **no random-detect ecn** and **default qos random-detect ecn** commands disables ECN marking for the shared buffer by removing the **qos random-detect ecn** command from ***running-config***.

Command Mode

Tx-Queue configuration

Command Syntax

random-detect ecn minimum-threshold **MIN** maximum-threshold **MAX**

no random-detect ecn

default random-detect ecn

Parameters

MIN and **MAX** parameters must use the same data unit.

- **MIN** Minimum threshold. Options include:

- ***1*** to ***256*** **mbytes** Megabyte units.
- ***1*** to ***256000*** **kbytes** Kilobyte units.
- ***1*** to ***256000000*** **bytes** Byte units.

- **MAX** Maximum threshold. Options include:
  - ***1*** to ***256*** **mbytes** Megabyte units.
  - ***1*** to ***256000*** **kbytes** Kilobyte units.
  - ***1*** to ***256000000*** **bytes** Byte units.

Related Command

tx-queue (Arad/Jericho) places the switch in tx-queue configuration mode.

Example

These commands enable ECN marking of unicast packets from ***unicast transmit queue 4*** of ***interface Ethernet 3/5/1***, setting thresholds at ***128*** kbytes and ***1280*** kbytes.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# tx-queue 4
switch(config-if-Et3/5/1-txq-4)# random-detect ecn minimum-threshold
 128 kbytes
maximum-threshold 1280 kbyte
switch(config-if-Et3/5/1-txq-4)# show active
interface Ethernet3/5/1
   tx-queue 4
      random-detect ecn minimum-threshold 128 kbytes maximum-threshold
 1280 kbytes
switch(config-if-Et3/5/1-txq-4)#
```

10.1.13.37 random-detect ecn (Helix)

The **`random-detect ecn`** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- **00** Non ECN Capable transport.
- **10** ECN Capable transport.
- **01** ECN Capable transport.
- **11** Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked **`congestion encountered`**.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked **`congestion encountered`**. The proportion of marked packets varies linearly with average queue size:
  - ***0%*** are marked when average queue size is less than or equal to minimum threshold.
  - ***100%*** are marked when average queue size is greater than or equal to maximum threshold.

  When transmitted packets are marked Non ECN Capable, congestion packets are dropped, not marked.

  Average queue length is tracked for transmit queues and the global pool independently. When either entity reaches its maximum threshold, all subsequent packets are marked.

The **no random-detect ecn** and **default random-detect ecn** commands disable ECN marking on the configuration mode queue, deleting the corresponding **random-detect ecn** command from ***running-config***.

Command Mode

Tx-Queue configuration

Command Syntax

random-detect ecn minimum-threshold **MIN** maximum-threshold **MAX**

no random-detect ecn

default random-detect ecn

Parameters

MIN and **MAX** parameters must use the same data unit.

- **MIN** Minimum threshold. Options include:
  - ***1*** to ***46080*** **segments** 208-byte segments units.
  - ***1*** to ***9*** **mbytes** Megabyte units.
  - ***1*** to ***9584*** **kbytes** Kilobyte units.
  - ***1*** to ***9584640*** **bytes** Byte units.
- **MAX** Maximum threshold. Options include:
  - ***1*** to ***46080*** **segments** 208-byte segments units.
  - ***1*** to ***9*** **mbytes** Megabyte units.
  - ***1*** to ***9584*** **kbytes** Kilobyte units.
  - ***1*** to ***9584640*** **bytes** Byte units.

Related Commands

- tx-queue (Helix) places the switch in tx-queue configuration mode.
- qos random-detect ecn global-buffer (Helix) enables ECN marking for globally shared packet memory.

Examples

- These commands enable ECN marking of unicast packets from ***transmit queue 4*** of ***interface ethernet 15***, setting thresholds at ***10*** and ***100*** segments.

```
switch(config)# interface ethernet 15
switch(config-if-Et15)# uc-tx-queue 4
switch(config-if-Et15-txq-4)# random-detect ecn minimum-threshold 10
 segments
maximum-threshold 100 segments
switch(config-if-Et15-txq-4)# show active
interface Ethernet15
   tx-queue 4
      random-detect ecn minimum-threshold 10 segments maximum-thres
hold 100
segments
switch(config-if-Et15-txq-4)# exit
switch(config-if-Et15)
```

- This command disables ECN marking of unicast packets from ***transmit queue 4*** of ***interface ethernet 15***.

```
switch(config-if-Et15-txq-4)# no random-detect ecn
switch(config-if-Et15-txq-4)# show active
interface Ethernet15
switch(config-if-Et15-txq-4)# exit
switch(config-if-Et15)#
```

10.1.13.38 random-detect ecn (Trident and Tomahawk)

The **`random-detect ecn`** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- **00** Non ECN Capable transport.
- **10** ECN Capable transport.
- **01** ECN Capable transport.
- **11** Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked **`congestion encountered`**.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked **`congestion encountered`**. The proportion of marked packets varies linearly with average queue size:
  - ***0%*** are marked when average queue size is less than or equal to minimum threshold.
  - ***100%*** are marked when average queue size is greater than or equal to maximum threshold.

  When transmitted packets are marked **`Non ECN Capable`**, congestion packets are dropped, not marked.

  Average queue length is tracked for transmit queues and the global pool independently. When either entity reaches its maximum threshold, all subsequent packets are marked.

The **`no random-detect ecn`** and **`default random-detect ecn`** commands disable ECN marking on the configuration mode queue, deleting the corresponding **`random-detect ecn`** command from ***running-config***.

Command Mode

Uc-Tx-Queue configuration

Command Syntax

`random-detect ecn minimum-threshold` **MIN** maximum-threshold **MAX**

`no random-detect ecn`

`default random-detect ecn`

Parameters

MIN and **MAX** parameters must use the same data unit.

- **MIN** Minimum threshold. Options include:
  - ***1*** to ***46080*** **segments** 208-byte segments units.
  - ***1*** to ***9*** **mbytes** Megabyte units.
  - ***1*** to ***9584*** **kbytes** Kilobyte units.

  - ***1*** to ***9584640*** **bytes** Byte units.
- **MAX** Maximum threshold. Options include:
  - ***1*** to ***46080*** **segments** 208-byte segments units.
  - ***1*** to ***9*** **mbytes** Megabyte units.
  - ***1*** to ***9584*** **kbytes** Kilobyte units.
  - ***1*** to ***9584640*** **bytes** Byte units.

Related Commands

- uc-tx-queue places the switch in the ***uc-tx-queue*** configuration mode.
- qos random-detect ecn global-buffer (Trident and Tomahawk) enables ECN marking for globally shared packet memory.

Examples

- These commands enable ECN marking of unicast packets from ***unicast transmit queue 4*** of ***interface ethernet 15***, setting thresholds at ***10*** and ***100*** segments.

```
switch(config)# interface ethernet 15
switch(config-if-Et15)# uc-tx-queue 4
switch(config-if-Et15-uc-txq-4)# random-detect ecn minimum-threshold
 10 segments
maximum-threshold 100 segments
switch(config-if-Et15-uc-txq-4)#show active
interface Ethernet15
   uc-tx-queue 4
      random-detect ecn minimum-threshold 10 segments maximum-thres
hold 100
segments
switch(config-if-Et15-uc-txq-4)# exit
switch(config-if-Et15)#
```

- This command disables ECN marking of unicast packets from ***unicast transmit queue 4*** of ***interface ethernet 15***.

```
switch(config-if-Et15-uc-txq-4)# no random-detect ecn
switch(config-if-Et15-uc-txq-4)# show active
interface Ethernet15
switch(config-if-Et15-uc-txq-4)# exit
switch(config-if-Et15)#
```

10.1.13.39 service-policy type qos input

The **`service-policy type qos input`** command applies the specified policy map to a QoS profile. The profile is then applied to an interface in interface configuration mode using the service-profile command.

The **`no service-policy`** type qos and **`default service-policy type qos`** command deletes the policy map from the profile.

The **`exit`** command returns the switch to global configuration mode.

Command Mode

QoS Profile Configuration

Command Syntax

`service-policy type qos input` ***policy_map_name***

`no service-policy type qos input` ***policy_map_name***

`default service-policy type qos input` ***policy_map_name***

Parameter

policy_map_name QoS policy map name.

Example

This command applies the policy map ***PM-1*** to the QoS profile ***TP***.

```
switch(config-qos-profile-TP)# service-policy type qos input PM-1
switch(config-qos-profile-TP)#
```

10.1.13.40 service-profile

The command applies the QoS profile to the configuration mode interface.

The **`no service-profile`** and the **`default service-profile`** command removes the QoS profile from the interface.

The **`exit`** command returns the switch to global configuration mode.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`service-profile` ***profile_name***

`no service-profile` ***profile_name***

`default service-profile` ***profile_name***

Parameter

profile_name QoS profile name.

Example

This commands applies the QoS profile ***TP*** to ***interfaceethernet 13***.

```
switch(config)# interface ethernet 13
switch(config-if-Et13)# service-profile TP
```

10.1.13.41 set extcommunity

The **`set extcommunity`** command adds a color extended community to be applied to routes affected by the route-map. Multiple set clauses can be applied to a single route-map to configure multiple colors for routes. Negating the command removes the entry from the route-map.

Command Mode

Config-route-map mode

Command Syntax

`set extcommunity` **COLOR-EXPRESSION** [additive | delete]

`no set extcommunity` **COLOR-EXPRESSION** [additive | delete]

`default set extcommunity` **COLOR-EXPRESSION** [additive | delete]

Parameters

- **COLOR-EXPRESSION** The color extended community to be applied to routes affected by the route-map.
- **additive** Adds the extended communities to those received.
- **delete** Deletes any matching extended color communities.

Example

```
arista(config)# route-map foo
arista(config-route-map foo)# set extcommunity color 1
arista(config-route-map foo)# set extcommunity color 2 color-only exact-match
arista(config-route-map foo)# set extcommunity color 3 color-only endpoint-matc
h null
arista(config-route-map foo)# set extcommunity color 4 color-only endpoint-matc
h any
```

10.1.13.42 shape rate (Interface – Arad/Jericho)

The **`shape rate`** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the shape rate (Tx-queue – Arad/Jericho) command. By default, outbound transmission rate is not bounded by a shape rate.

The **`no shape rate`** and **`default shape rate`** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **`shape rate`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`shape rate` ***byte_limit*** [kbps]

`no shape rate`

`default shape rate`

Parameters

byte_limit Shape rate applied to interface (Kbps). Value ranges from ***162*** to ***100000000***.

Example

This command configures a port shape rate of ***5*** Gbps on ***interface ethernet 3/5/1***.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# shape rate 5000000
switch(config-if-Et3/5/1)# show qos interfaces ethernet 3/5/1

Ethernet3/5/1:

   Port shaping rate: 5000012 / 5000000 kbps

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
```

```
   4       - / -          - / -     (  -  )     SP / SP     D
   3       - / -          - / -     (  -  )     SP / SP     D
   2       - / -          - / -     (  -  )     SP / SP     D
   1       - / -          - / -     (  -  )     SP / SP     D
   0       - / -          - / -     (  -  )     SP / SP     D

switch(config-if-Et3/5/1)#
```

10.1.13.43 shape rate (Interface – FM6000)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the shape rate (Tx-queue – FM6000) command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

shape rate ***byte_limit*** [kbps]

no shape rate

default shape rate

Parameters

byte_limit Shape rate applied to interface (Kbps). Value ranges from ***7000*** to ***10000000***.

Guidelines

Enabling port shaping on an FM6000 interface disables queue shaping internally. Disabling port shaping restores queue shaping as specified in ***running-config***.

Example

This command configures a port shape rate of ***5*** Gbps on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# shape rate 5000000
switch(config-if-Et5)#
```

10.1.13.44 shape rate (Interface – Helix)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the shape rate (Interface – Helix) command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`shape rate` DATA_LIMIT

`no shape rate`

`default shape rate`

Parameters

- **DATA_LIMIT** Shape rate applied to interface. Value range varies with data unit:
  - ***8*** to ***40000000*** ***8*** to ***40000000*** kbytes per second.
  - ***8*** to ***40000000*****kbps** ***8*** to ***40000000*** kbytes per second.
  - ***8*** to ***60000000*****pps** ***8*** to ***60000000*** packets per second.

Guidelines

Shaping rates of at least ***8*** kbps are supported. At shaping rates smaller than ***1*** Mbps, granularity and rounding errors may skew the actual shaping rate by ***20%*** from the specified rate.

Example

This command configures a port shape rate of ***5*** Gbps on ***interface ethernet 17***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# shape rate 5000000 kbps
switch(config-if-Et17)# show qos interface ethernet 17/3

Ethernet17:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000 / 5000000 kbps

  Tx       Bandwidth                  Shape Rate         Priority
 Queue     Guaranteed (units)          (units)
   ---------------------------------------------------------------
   7         - / -    (  -  )       - / -    (  -  )    SP / SP
   6         - / -    (  -  )       - / -    (  -  )    SP / SP

switch(config-if-Et17)#
```

10.1.13.45 shape rate (Interface – Petra)

The **`shape rate`** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the shape rate (Tx-queue – Petra) command. By default, outbound transmission rate is not bounded by a shape rate.

The **`no shape rate`** and **`default shape rate`** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **`shape rate`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`shape rate` ***data_limit*** [kbps]

`no shape rate`

`default shape rate`

Parameters

data_limit Shape rate applied to interface (Kbps). Value ranges from ***100*** to ***10000000***.

Guidelines

The following port shaping rates are supported:

- 1G ports: above 100 kbps.
- 10G ports: above 7900 kbps.

Commands that specify a smaller shape rate disable port shaping on the interface.

Example

This command configures a port shape rate of ***5*** Gbps on ***interface ethernet 3/3***.

```
switch(config)# interface ethernet 3/3
switch(config-if-Et3/3)# shape rate 5000000
switch(config-if-Et3/3)# show active
interface Ethernet3/3
   shape rate 5000000
switch(config-if-Et3/3)#
```

10.1.13.46 shape rate (Interface – Trident and Tomahawk)

The **`shape rate`** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the shape rate (Tx-queue – Trident and Tomahawk) command. By default, outbound transmission rate is not bounded by a shape rate.

The **`no shape rate`** and **`default shape rate`** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **`shape rate`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`shape rate` **DATA_LIMIT**

`no shape rate`

`default shape rate`

Parameters

- **DATA_LIMIT** Shape rate applied to interface. Value range varies with data unit:
  - ***8*** to ***40000000*** ***8*** to ***40000000*** kbytes per second.
  - ***8*** to 40000000**kbps** ***8*** to ***40000000*** kbytes per second.
  - ***8*** to ***60000000*****pps** ***8*** to ***60000000*** packets per second.

Guidelines

Shaping rates of at least ***8*** kbps are supported. At shaping rates smaller than ***1*** Mbps, granularity and rounding errors may skew the actual shaping rate by ***20%*** from the specified rate.

Example

This command configures a port shape rate of ***5*** Gbps on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# shape rate 5000000
switch(config-if-Et5)#
```

10.1.13.47 shape rate (Interface – Trident II)

The **`shape rate`** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the shape rate (Tx-queue – Trident II) command. By default, outbound transmission rate is not bounded by a shape rate.

The **`no shape rate`** and **`default shape rate`** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **`shape rate`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`shape rate` **DATA_LIMIT**

`no shape rate`

`default shape rate`

Parameters

DATA_LIMIT Shape rate applied to interface. Value range varies with data unit:

- • ***8*** to ***40000000*** **kbps** ***8*** to ***40000000*** kbytes per second.
  - ***8*** to ***40000000*** **kbps** ***8*** to ***40000000*** kbytes per second.
  - ***8*** to ***60000000*** **pps** ***8*** to ***60000000*** packets per second.

Guidelines

Shaping rates of at least ***8*** kbps are supported. At shaping rates smaller than 1 Mbps, granularity and rounding errors may skew the actual shaping rate by ***20%*** from the specified rate.

Example

This command configures a port shape rate of ***5*** Gbps on ***interface ethernet 17/3***.

```
switch(config)# interface ethernet 17/3
switch(config-if-Et17/3)# shape rate 5000000 kbps
switch(config-if-Et17/3)# show qos interface ethernet 17/3

Ethernet17/3:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000 / 5000000 kbps
```

```
  Tx       Bandwidth                  Shape Rate         Priority
 Queue     Guaranteed (units)          (units)
   ------------------------------------------------------------
   7         - / -    (  -  )       - / -    (  -  )    SP / SP
   6         - / -    (  -  )       - / -    (  -  )    SP / SP

switch(config-if-Et17/3)#
```

10.1.13.48 shape rate (Tx-queue – Arad/Jericho)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the shape rate (Interface – Arad/Jericho) command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

Shaping rates greater than ***50000*** kbps are supported. At lower shaping rates (less than ***10*** Mbps), granularity and rounding errors may skew the actual shaping rate by ***20%*** from the specified rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

shape rate ***byte_limit*** [kbps]

no shape rate

default shape rate

Parameters

byte_limit Shape rate applied to interface (Kbps). Value ranges from ***50000*** to ***100000000***.

Related Command

tx-queue (Arad/Jericho) places the switch in tx-queue configuration mode.

Related Information

shape rate (Interface – Arad/Jericho)

Example

These commands configure a shape rate of ***1*** Gbps on transmit queues ***3*** and ***4*** of ***interface ethernet 3/4/1***.

```
switch(config)# interface ethernet 3/4/1
switch(config-if-Et3/4/1)# tx-queue 4
switch(config-if-Et3/4/1-txq-4)# shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-4)# tx-queue 3
switch(config-if-Et3/4/1-txq-3)# shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-3)# show qos interface ethernet 3/4/1

Ethernet3/4/1:

   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
```

```
   ------------------------------------------------------
   7       - / -          - / -     (  -  )     SP / SP     D
   6       - / -          - / -     (  -  )     SP / SP     D
   5       - / -          - / -     (  -  )     SP / SP     D
   4       - / -       999 / 1000 ( Mbps )     SP / SP     D
   3       - / -       999 / 1000 ( Mbps )     SP / SP     D
   2       - / -          - / -     (  -  )     SP / SP     D
   1       - / -          - / -     (  -  )     SP / SP     D
   0       - / -          - / -     (  -  )     SP / SP     D

switch(config-if-Et3/4/1-txq-3)#
```

10.1.13.49 shape rate (Tx-queue – FM6000)

The **shape rate**command specifies the maximum bandwidth for outbound traffic on the transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the shape rate (Interface – FM6000) command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

Queue shaping on an FM6000 port is supported only when port shaping is not enabled on the interface. Enabling port shaping on a port disables queue shaping internally. Disabling port shaping restores queue shaping as specified by ***running-config***.

Shaping rates greater than ***460*** kbps are supported. At lower shaping rates (less than ***10*** Mbps), granularity and rounding errors may skew the actual shaping rate by ***20%*** from the specified rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the transmit queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

shape rate ***byte_limit*** [kbps]

no shape rate

default shape rate

Parameters

byte_limit Shape rate applied to interface (Kbps). Value ranges from ***464*** to ***10000000***.

Related Commands

- tx-queue (FM6000) places the switch in tx-queue configuration mode.
- shape rate (Interface – FM6000) configures the shape rate for a configuration mode interface.

Example

These commands configure a shape rate of ***1*** Gbps (***1,000,000*** Kbps) on transmit queues ***3*** and ***4*** of ***interface ethernet 19***.

```
switch(config)# interface ethernet 19
switch(config-if-Et19)# tx-queue 4
switch(config-if-Et19-txq-4)# shape rate 1000000
switch(config-if-Et19-txq-4)# tx-queue 3
switch(config-if-Et19-txq-3)# shape rate 1000000
switch(config-if-Et19-txq-3)# show qos interface ethernet 19

Ethernet19:
```

```
   Trust Mode: COS

   Tx-Queue    Bandwidth     Shape Rate      Priority
               (percent)        (Kbps)
   ---------------------------------------------------
          6          N/A     disabled         strict
          5          N/A     disabled         strict
          4          N/A      1000000         strict
          3           25      1000000    round-robin
          2           25     disabled    round-robin
          1           25     disabled    round-robin
          0           25     disabled    round-robin

switch(config-if-Et19-txq-3)#
```

10.1.13.50 shape rate (Tx-queue – Helix)

The **`shape rate`** command specifies the maximum bandwidth for outbound traffic on the transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the shape rate (Interface – Helix) command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

The **`no shape rate`** and **`default shape rate`** commands remove the shape rate bandwidth limit on the configuration mode transmit queue by deleting the corresponding **`shape rate`** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

`shape rate` ***byte_limit*** [kbps]

`no shape rate`

`default shape rate`

Parameters

DATA_LIMIT Shape rate applied to the queue. Value range varies with data unit:

- • ***8*** to ***40000000*** ***8*** to 40,000,000 kbytes per second.
  - ***8*** to ***40000000*** **kbps** ***8*** to ***40000000*** kbytes per second.
  - ***8*** to ***60000000*** **pps** ***8*** to ***60000000*** packets per second.

Restrictions

Queue shaping is not supported in cut-through mode.

Related Commands

- tx-queue (Helix) places the switch in ***tx-queue*** configuration mode.
- shape rate (Interface – Helix) configures the shape rate for a configuration mode interface.

Example

These commands configure a shape rate of ***1*** Gbps (***1,000,000*** Kbps) on transmit queues ***3*** and ***4*** of ***interface Eethernet 17/3***.

```
switch(config)# interface ethernet 17/3
switch(config-if-Et17/3)# tx-queue 4
switch(config-if-Et17/3-txq-4)# shape rate 1000000 kbps
```

```
switch(config-if-Et17/3-txq-4)# tx-queue 3
switch(config-if-Et17/3-txq-3)# shape rate 1000000 kbps
switch(config-if-Et17/3-txq-3)# show qos interface ethernet 17/3

Ethernet17/3:

  Tx       Bandwidth                  Shape Rate         Priority
 Queue     Guaranteed (units)          (units)
   ------------------------------------------------------------
   7         - / -    (  -  )       - / -    (  -  )    SP / SP
   6         - / -    (  -  )       - / -    (  -  )    SP / SP
   5         - / -    (  -  )       - / -    (  -  )    SP / SP
   4         - / -    (  -  )       1 / 1    ( Gbps )    SP / SP
   3         - / -    (  -  )       1 / 1    ( Gbps )    SP / SP
   2         - / -    (  -  )       - / -    (  -  )    SP / SP
   1         - / -    (  -  )       - / -    (  -  )    SP / SP
   0         - / -    (  -  )       - / -    (  -  )    SP / SP

switch(config-if-Et17/3-txq-3)#
```

10.1.13.51 shape rate (Tx-queue – Petra)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the shape rate (Interface – Petra) command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

Queue shaping applies only to unicast traffic. Shaping rates of at least 162 Kbps are supported.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

shape rate DATA_LIMIT

no shape rate

default shape rate

Parameters

DATA_LIMIT Shape rate applied to the queue. Value range varies with data unit:

- • ***8*** to ***40000000*** **kbps** Range is from ***8*** to ***40000000*** kbytes per second.
  • ***8***to ***60000000*****pps** Range is from ***8*** to ***60000000*** packets per second.

  Shaping rates greater than ***460*** kbps are supported. At lower shaping rates (less than ***10*** Mbps), granularity and rounding errors may skew the actual shaping rate by ***20%*** from the specified rate.

Related Commands

- tx-queue (Petra) places the switch in ***tx-queue*** configuration mode.
- shape rate (Interface – Petra) configures the shape rate for a configuration mode interface.

Example

These commands configure a shape rate of ***1*** Gbps (***1,000,000*** Kbps) on transmit queues ***3*** and ***4*** of ***interface ethernet 3/28***.

```
switch(config)# interface ethernet 3/28
switch(config-if-Et3/28)# tx-queue 4
switch(config-if-Et3/28-txq-4)# shape rate 1000000
switch(config-if-Et3/28-txq-4)# tx-queue 3
switch(config-if-Et3/28-txq-3)# shape rate 1000000
switch(config-if-Et3/28-txq-3)# show qos interface ethernet 3/28

Ethernet3/28:

   Tx-Queue    Bandwidth     Shape Rate      Priority
               (percent)        (Kbps)
   ---------------------------------------------
          7          N/A     disabled         strict
          6          N/A     disabled         strict
          5          N/A     disabled         strict
          4          N/A      1000000         strict
          3           25      1000000    round-robin
          2           25     disabled    round-robin
          1           25     disabled    round-robin
          0           25     disabled    round-robin

switch(config-if-Et3/28-txq-3)#
```

10.1.13.52 shape rate (Tx-queue – Trident and Tomahawk)

The **`shape rate`** command specifies the maximum bandwidth for outbound traffic on the configuration mode transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the shape rate (Interface – Trident and Tomahawk) command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

The **`no shape rate`** and **`default shape rate`** commands remove the shape rate limit from the configuration mode transmit queue by deleting the corresponding **`shape rate`** command from ***running-config***.

Command Mode

Mc-Tx-Queue configuration

Uc-Tx-Queue configuration

Command Syntax

`shape rate` **DATA_LIMIT**

`no shape rate`

`default shape rate`

Parameters

- **DATA_LIMIT** Shape rate applied to the queue. Value range varies with data unit:
  - ***8*** to ***40000000*** **kbps** Range is from ***8*** to ***40000000*** kbytes per second.
  - ***8*** to ***60000000*** **pps** Range is from ***8*** to ***60000000*** packets per second.

Related Commands

- mc-tx-queue places the switch in ***mc-tx-queue*** configuration mode.
- uc-tx-queue places the switch in ***uc-tx-queue*** configuration mode.

- shape rate (Interface – Trident and Tomahawk) configures the shape rate for a configuration mode interface.

Guidelines

Shaping rates of at least ***8*** kbps are supported. At shaping rates smaller than ***1*** Mbps, granularity and rounding errors may skew the actual shaping rate by ***20%*** from the specified rate.

When two queues source traffic from the same traffic class and the higher priority queue is shaped, that queue consumes all internal buffers, starving the lower priority queue even if bandwidth is available.

Example

These commands configure a shape rate of ***1*** Gbps (***1,000,000*** Kbps) on ***unicast transmit queues 3*** and ***multicast transmit 4*** of ***interface ethernet 7***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)# shape rate 1000000
switch(config-if-Et7-uc-txq-3)# mc-tx-queue 2
switch(config-if-Et7-mc-txq-2)# shape rate 1000000
switch(config-if-Et7-mc-txq-2)# show qos interface ethernet 7

Ethernet7:
   Tx-Queue    Bandwidth     Shape Rate      Priority    Priority Group
               (percent)        (Kbps)
   ------------------------------------------------------------------
        UC7          N/A     disabled         strict                 1
        UC6          N/A     disabled         strict                 1
        MC3          N/A     disabled         strict                 1
        UC5          N/A     disabled         strict                 0
        UC4          N/A     disabled         strict                 0
        MC2          N/A      1000000         strict                 0
        UC3           20      1000000    round-robin                 0
        UC2           16     disabled    round-robin                 0
        MC1           16     disabled    round-robin                 0
        UC1           16     disabled    round-robin                 0
        UC0           16     disabled    round-robin                 0
        MC0           16     disabled    round-robin                 0

switch(config-if-Et7-mc-txq-2)#
```

10.1.13.53 shape rate (Tx-queue – Trident II)

The **`shape rate`** command specifies the maximum bandwidth for outbound traffic on the configuration mode transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the shape rate (Interface – Trident II) command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

The **`no shape rate`** and **`default shape rate`** commands remove the shape rate bandwidth limit on the configuration mode transmit queue by deleting the corresponding **`shape rate`** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

`shape rate` ***byte_limit*** [*kbps*]

no shape rate

default shape rate

Parameters

DATA_LIMIT Shape rate applied to the queue. Value range varies with data unit:

- • ***8*** to ***40000000*** **kbps** Range is from ***8*** to ***40000000*** kbytes per second.
  - ***8*** to ***60000000*** **pps** Range is from ***8*** to ***60000000*** packets per second.

Restrictions

Queue shaping is not supported in cut-through mode

Related Commands

- tx-queue (Trident II) places the switch in ***tx-queue*** configuration mode.
- shape rate (Interface – Trident II) configures the shape rate for a configuration mode interface.

Example

These commands configure a shape rate of ***1*** Gbps (***1,000,000*** Kbps) on transmit queues ***3*** and ***4*** of ***interface ethernet 17/3***.

```
switch(config)# interface ethernet 17/3
switch(config-if-Et17/3)# tx-queue 4
switch(config-if-Et17/3-txq-4)# shape rate 1000000 kbps
switch(config-if-Et17/3-txq-4)# tx-queue 3
switch(config-if-Et17/3-txq-3)# shape rate 1000000 kbps
switch(config-if-Et17/3-txq-3)# show qos interface ethernet 17/3

Ethernet17/3:

  Tx       Bandwidth                  Shape Rate         Priority
 Queue     Guaranteed (units)          (units)
   ------------------------------------------------------------
   7         - / -    (  -  )       - / -    (  -  )    SP / SP
   6         - / -    (  -  )       - / -    (  -  )    SP / SP
   5         - / -    (  -  )       - / -    (  -  )    SP / SP
   4         - / -    (  -  )       1 / 1    ( Gbps )   SP / SP
   3         - / -    (  -  )       1 / 1    ( Gbps )   SP / SP
   2         - / -    (  -  )       - / -    (  -  )    SP / SP
   1         - / -    (  -  )       - / -    (  -  )    SP / SP
   0         - / -    (  -  )       - / -    (  -  )    SP / SP

switch(config-if-Et17/3-txq-3)#
```

10.1.13.54 show interface counters queue drop-precedence

The **show interface counters queue drop-precedence** command displays the drop-precedence counters.

Command Mode

EXEC

Command Syntax

show interface counters queue drop-precedence

Example

This command displays the drop precedence counts on two interfaces.

```
switch# show interface counters queue drop-precedence
intf        0          1          2
Et1/1     100          0        200
Et1/2     200          0        300 switch#
```

10.1.13.55 show platform petraA traffic-class

The **show platform petraA traffic-class** command displays the traffic class assignment on all specified Petra chips. Each chip controls eight Ethernet interfaces. The default traffic class of an interface is specified by the traffic class assigned to the chip that controls the interface.

Traffic class assignments are configured with the platform petraA traffic-class command.

Valid command options include:

- **show platform petraA traffic-class** Traffic class of all chips on all linecard.
- **show platform petraA CHIP_NAME traffic-class** Traffic class of specified chip.
- **show platform petraA MODULE_NAME traffic-class** Traffic class of all chips on specified linecard.

Command Mode

EXEC

Command Syntax

show platform petraA traffic-class

show platform petraA CHIP_NAME traffic-class

show platform petraA MODULE_NAME traffic-class

Parameters

- **CHIP_NAME** Name of Petra chip on linecard that control Ethernet ports. Options include:
  - **petra *cardX* / *chipY*** All ports on PetraA chip ***chipY*** on linecard ***cardX*** (7500 Series).
  - **petra *chipZ*** All ports on PetraA chip ***chipZ*** (7048 Series).

  7500 Series

  Switches can contain up to eight linecards. ***cardX*** varies from ***3*** to ***10***.

  Each linecard contains six PetraA chips. Each chip controls eight ports. ***chipY*** varies from ***0*** to ***5***:

  - 0 controls ports 1 through 8
  - 1 controls ports 9 through 16
  - 2 controls ports 17 through 24
  - 3 controls ports 25 through 32
  - 4 controls ports 33 through 40
  - 5 controls ports 41 through 48

  7048 Series

  Each switch contains two PetraA chips. ***chipZ*** varies from ***0*** to ***1***:

  - 0 controls ports 1 through 32
  - 1 controls ports 33 through 52
- **MODULE_NAME** Name and number of linecard (7500 Series). Options include:
  - **module linecard *mod_num*** Linecard number (***3*** to ***10***).
  - **module *mod_num*** Linecard number (***3*** to ***10***).

Related Command

platform petraA traffic-class configures the default traffic class used by all ports on a specified chip.

Example

This command displays the traffic class of all chips on ***linecard 3***.

```
switch# show platform petraA module linecard 3 traffic-class
Petra3/0 traffic-class: 1
Petra3/1 traffic-class: 1
Petra3/2 traffic-class: 1
Petra3/3 traffic-class: 1
Petra3/4 traffic-class: 5
Petra3/5 traffic-class: 1 switch#
```

10.1.13.56 show platform trident tcam qos detail

The **`show platform trident tcam qos detail`** command displays the list of all the SVIs that are sharing the TCAM entries.

Command Mode

EXEC

Command Syntax

`show platform trident tcam qos detail`

Example

This command displays the list of all the SVIs that are sharing the TCAM entries.

```
switch(config)# show platform trident tcam qos detail

=== Policy-map p01 type qos on switch Linecard0/0 ===
Interfaces : Vlan2 Vlan1
=-= Interface BitMap =-=
0x00000000000000001FFFFFE
```

10.1.13.57 show platform trident tcam shared vlan interface-class-id

The **`show platform trident tcam shared vlan interface-class-id`** command displays what SVIs are currently sharing the QoS policy-map in the below output under QoS PMAP Data.

Command Mode

EXEC

Command Syntax

`show platform trident tcam shared vlan interface-class-id`

Example

This command displays what SVIs are currently sharing the QoS policy-map in the below output under QoS PMAP Data.

```
switch(config)# show platform trident tcam shared vlan interface-cla
ss-id

=== Shared RACL Data on switch Linecard0/0 ===
=== Shared QoS Policy-map Data on switch Linecard0/0 ===
Interface Class Id             VLANs
1                               1 2
```

10.1.13.58 show platform xp qos tcam hit

The **`show platform xp qos tcam hit`** command displays the TCAM entries programmed for each policy-map as well as the traffic hits. The hits option is used to see the TCAM entries with nonzero traffic hits.

Command Mode

EXEC

Command Syntax

`show platform xp qos tcam hit`

Example

This command displays the QoS TCAM hits on ***interface ethernet 10/1***.

```
switch# show platform xp qos tcam hit

=== Policy-map test type qos on switch 0 ===
Assigned to ports: Ethernet10/1
=-= Class-map test type qos =-=
=== ACL test
================================================================================================
|Seq|AclId|Prot|Port|SPort|Ecn|FFlg|DPort|Vlan|Action|Hits|Src Ip|Dest Ip|hwId    | | |     |      | | | |dscp|cos  |
tc|PolId| | | |
================================================================================================
| 10| 0x01|    |    |     |   |0x04|     |    | 4    |-   | -    | -     |91852787| | | | 0 | 0x00 | | | |     |0xfb|
  |     |  | | |
------------------------------------------------------------------------------------------------
```

10.1.13.59 show policy-map interface

The **`show policy-map interface`** command displays contents of the policy map applied to specified the interface.

Command Mode

EXEC

Command Syntax

`show policy-map interface` ***interface_name***

Parameters

interface_name Interface for which command returns data. Options include:

- ***no parameter*** Returns data for all interfaces.
- ethernet ***e_range*** Ethernet interfaces specified by ***e_range***.
- **port-channel *p_range*** Port channel interfaces specified by ***p_range***.

Example

This command displays the name and contents of the policy map applied to ***interface Ethernet 1***.

```
switch# show policy-map interface ethernet 1
Service-policy input: p1
 Hardware programming status: Successful

 Class-map: c2001 (match-any)
    Match: vlan 2001 0xfff
       set dscp 4

 Class-map: c2002 (match-any)
    Match: vlan 2002 0xfff
       set dscp 8

 Class-map: c2003 (match-any)
    Match: vlan 2003 0xfff
       set dscp 12
```

10.1.13.60 show policy-map

The **`show policy-map`** command displays the policy map information for the configured policy map.

Command Mode

EXEC

Command Syntax

`show policy-map` ***policy_map_name*** [counters][interface | summary]

Parameters

- ***policy_map_name*** QoS policy map name.
- **counters** Specifies the policy map traffic match count (This parameter is applicable only on DCS-7010, DCS-7050X, DCS7250X, DCS-7300X and DCS-7280(E/R), DCS-7500(E/R) series switches.)
- **interface** Specifies the service policy on an interface.
- **summary** Policy map summary.

Examples

- The **`show policy-map command`** displays the information for the policy map ***policy1***.

```
switch# show policy-map policy1
Service-policy policy1
Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes
Class-map: class-default (match-any)
```

- The **`show policy-map counters`** command displays the policy map traffic match count for the policy map configured.

```
switch# show policy-map policy1 counters

Service-policy input: policy1
 Hardware programming status: Successful
 Class-map: class1 (match-any)
    Match: vlan 20-40,1000-1250
       police rate 100 mbps burst-size 100 kbytes
        Interface: Ethernet16/1
          Conformed 28621 packets, 7098008 bytes -------------- packet match count

Class-map: class-default (match-any)
```

```
       Matched Packets: 19 -------------- packet match count
```

10.1.13.61 show qos interfaces random-detect ecn

The **show qos interfaces random-detect ecn** command displays the Explicit Congestion Notification (ECN) configuration for each transmit queue on the specified interfaces.

Command Mode

EXEC

Command Syntax

show qos interfaces [**INTERFACE_NAME**] random-detect ecn

Parameters

INTERFACE_NAME Interface for which command returns data. Options include:

- ***no parameter*** Returns data for all interfaces.
- **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
- **port-channel *p_range*** Port-Channel Interfaces specified by ***p_range***.

Example

This command configures ECN parameters for transmit queues ***0*** through ***3*** on ***interface ethernet 3/5/1***, then displays that configuration.

```
switch(config)# interface ethernet 3/5/1
switch(config-if-Et3/5/1)# tx-queue 0
switch(config-if-Et3/5/1-txq-0)# random-detect ecn minimum-threshold 2560 kbytes
maximum-threshold 256000 kbytes
switch(config-if-Et3/5/1-txq-0)# tx-queue 1
switch(config-if-Et3/5/1-txq-1)# random-detect ecn minimum-threshold 25600 kbytes
maximum-threshold 128000 kbytes
switch(config-if-Et3/5/1-txq-1)# tx-queue 2
switch(config-if-Et3/5/1-txq-2)# random-detect ecn minimum-threshold 25600 bytes
maximum-threshold 128000 bytes
switch(config-if-Et3/5/1-txq-2)# tx-queue 3
switch(config-if-Et3/5/1-txq-3)# random-detect ecn minimum-threshold 25 mbytes
maximum-threshold 128 mbytes
switch(config-if-Et3/5/1-txq-3)# show qos interfaces ethernet 3/5/1 random-detect
ecn

Ethernet3/5/1:

   Tx-Queue   Mininimum Threshold    Maximum Threshold    Threshold Unit
   ------------------------------------------------------------------
          7                     -                    -                 -
          6                     -                    -                 -
          5                     -                    -                 -
          4                     -                    -                 -
          3                    25                  128            mbytes
          2                 25600               128000             bytes
          1                 25600               128000            kbytes
          0                  2560               256000            kbytes

switch(config-if-Et3/5/1-txq-3)#
```

10.1.13.62 show qos interfaces trust

The **show qos interfaces trust** command displays the configured and operational QoS trust mode of all specified interfaces.

Command Mode

EXEC

Command Syntax

`show qos interfaces` [**INTERFACE_NAME**] **`trust`**

Parameters

INTERFACE_NAME Interface for which command returns data. Options include:

- ***no parameter*** Returns data for all interfaces.
- **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
- **port-channel *p_range*** Port-Channel Interfaces specified by ***p_range***.

Example

These commands configure a variety of QoS trust settings on a set of interfaces, then displays the QoS trust mode on these interfaces.

```
switch(config)# interface ethernet 1/1
switch(config-if-Et1/1)# qos trust cos
switch(config-if-Et1/1)# interface ethernet 1/2
switch(config-if-Et1/2)# qos trust dscp
switch(config-if-Et1/2)# interface ethernet 1/3
switch(config-if-Et1/3)# no qos trust
switch(config-if-Et1/3)# interface ethernet 1/4
switch(config-if-Et1/4)# default qos trust
switch(config-if-Et1/4)# interface ethernet 2/1
switch(config-if-Et2/1)# no switchport
switch(config-if-Et2/1)# default qos trust
switch(config-if-Et2/1)# show qos interface ethernet 1/1 - 2/4 trust

Port                                    Trust Mode
                                Operational           Configured
-------------------------------------------------------------------
Ethernet1/1                     COS                   COS
Ethernet1/2                     DSCP                  DSCP
Ethernet1/3                     UNTRUSTED             UNTRUSTED
Ethernet1/4                     COS                   DEFAULT
Ethernet2/1                     DSCP                  DEFAULT
Ethernet2/2                     COS                   DEFAULT
Ethernet2/3                     COS                   DEFAULT
Ethernet2/4                     COS                   DEFAULT

switch(config-if-Et2/1)#
```

10.1.13.63 show qos interfaces

The **`show qos interfaces`** command displays the QoS, DSCP, and transmit queue configuration on a specified interface. Information provided by this command includes the ports trust setting, the default CoS value, and the DSCP value.

Command Mode

EXEC

Command Syntax

`show qos interfaces` **INTERFACE_NAME**

Parameters

INTERFACE_NAME Interface For which command returns data. Options include:

- ***no parameter*** Returns data for all interfaces.
- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.

Example

This command lists the QoS configuration for ***interface ethernet 4***.

```
switch> show qos interfaces ethernet 4

Ethernet4:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000Kbps

   Tx-Queue   Bandwidth    ShapeRate      Priority
              (percent)       (Kbps)
   ---------------------------------------------
          0          50     disabled   round-robin
          1          50     disabled   round-robin
          2         N/A     disabled        strict
          3         N/A      1000000        strict
          4         N/A      1000000        strict
          5         N/A      1500000        strict
          6         N/A      2000000        strict switch>
```

10.1.13.64 show qos interfaces latency maximum

The **`show qos interfaces latency maximum`** command displays the maximum latency tail-drop threshold active on each Tx queue for a specified interface.

Command Mode

EXEC

Command Syntax

`show qos interfaces` ***INTERFACE_NAME*** **`latency maximum`**

Parameters

INTERFACE_NAME: Name of the interface.

This command lists the maximum latency for VOQ tail drop on ***interface ethernet 23/1***.

```
switch# show qos interfaces ethernet 23/1 latency maximum

Ethernet23/1:
Tx Queue            Maximum Latency
--------            ---------------
       7                          -
       6                          -
       5                          -
       4                          -
       3                      10 ms
       2                          -
       1                          -
       0                          -

switch# show qos profile
qos profile latency
    tx-queue 3
```

```
        latency maximum 4000 microseconds
    Tx-queue 4
        latency maximum 30 milliseconds switch#
```

10.1.13.65 show qos maps

The **show qos maps** command lists the number of traffic classes that the switch supports and displays the CoS-Traffic Class, DSCP-Traffic Class, Traffic Class-CoS, and Traffic Class-Transmit Queue maps.

Command Mode

EXEC

Command Syntax

show qos maps

Example

This command displays the QoS maps that are configured on the switch.

```
switch> show qos maps
Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
   Cos Rewrite:  Disabled
   Dscp Rewrite:  Disabled

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  0  2  3  4  5  6  7

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  3  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  5  5  5  5  5  5  5  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ----------------------------
     cos:  1  0  2  3  4  5  6  7

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ------------------------------
     dscp:  8  0 16 24 32 40 48 56

   Tc - tx-queue map:
     tc:         0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:  0  1  2  3  4  5  6  7 switch>
```

10.1.13.66 show qos map dscp to traffic-class

The **show qos map dscp to traffic-class** command shows all DSCP to traffic-class maps, or one specified DSCP to TC map.

Command Mode

Privileged EXEC mode

Command Syntax

show qos map dscp to traffic-class [***map_name***]

Parameters

- ***map_name*** The name of the DSCP to traffic-class map. If this is not specified, all DSCP to TC maps are shown.

Example

This command shows the DSCP to TC map named ***map1.***

```
switch#show qos map dscp to traffic-class map1
   DSCP to TC map: map1
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     6  6  6  6  6  6  3  3  3  3
      3 :     3  3  4  4  4  7  4  4  4  4
      4 :     5  5  5  5  5  5  5  5  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7 switch#
```

10.1.13.67 show qos profile summary

The **show qos profile summary** command displays the QoS profile summary of those which are part of the running configuration.

Command Mode

EXEC

Command Syntax

show qos profile summary

Example

This command shows a summary of all QoS profiles configured on the switch.

```
switch(config)# show qos profile summary
Qos Profile: p
Configured on: Et13,7
Fabric
Po12
Qos Profile: p2
Configured on: Et56
```

10.1.13.68 show qos profile

The **show qos profile** command displays the contents of the specified QoS profile or of all QoS profiles in the running configuration.

Command Mode

EXEC

Command Syntax

`show qos profile` ***profile_name***

Parameter

profile_name QoS profile name.

Examples

- This command displays the contents of all QoS profiles configured on the switch.

```
switch(config)# show qos profile
qos profile p
qos cos 1
no priority-flow-control pause watchdog
priority-flow-control priority 1 no-drop
priority-flow-control priority 2 no-drop
qos profile p2
qos cos 3
priority-flow-control priority 0 no-drop
```

- This command displays the configuration attached and information specific to ***QoS profile p2***.

```
switch# show qos profile p2
qos profile p2
qos cos 3
priority-flow-control priority 0 no-drop
```

10.1.13.69 show qos random-detect ecn

The command displays the global Explicit Congestion Notification (ECN) configuration.

Command Mode

EXEC

Command Syntax

`show qos random-detect ecn`

Example

These commands configure global ECN parameters, then displays that configuration.

```
switch(config)# qos random-detect ecn global-buffer minimum-threshold
 2 mbytes
maximum-threshold 5 mbytes
switch(config)# show qos random-detect ecn

   Minimum Threshold:  2
   Maximum Threshold:  5
   Threshold Unit:  mbytes

switch(config)#
```

10.1.13.70 show run|grep sharing

The **show run|grep sharing** command displays whether the QoS policy-map sharing on SVIs is enabled or disabled.

Command Mode

EXEC

Command Syntax

show run|grep sharing

Example

This command displays whether the QoS policy-map sharing on SVIs is enabled or disabled.

```
switch# show run|grep sharing
hardware access-list qos resource sharing vlan in ----
```

If this message is displayed then QoS policy-map sharing on SVIs is enabled.

10.1.13.71 tx-queue (Arad/Jericho)

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Arad and Jericho platform switches have eight queues, ***0*** through ***7***, and all queues are exposed through the CLI. However, ***queue 7*** is not user-configurable. ***Queue 7*** is always mapped to ***traffic class 7***, which is reserved for control traffic.

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no tx-queue** and **default tx-queue** commands remove the configuration for the specified transmit queue by deleting all corresponding **tx-queue** mode statements from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

tx-queue ***queue_level***

Parameters

queue_level The transmit queue. Values range from ***0*** to ***7***.

Commands Available in tx-queue Configuration Mode

- bandwidth percent (Arad/Jericho)
- priority (Arad/Jericho)
- shape rate (Tx-queue – Arad/Jericho)

Guidelines

Arad and Jericho platform switch queues handle unicast traffic. Queues for multicast traffic are not supported.

Example

This command enters Tx-queue configuration mode for ***transmit queue 4*** of ***interface ethernet 3/3/3***.

```
switch(config)# interface ethernet 3/3/3
switch(config-if-Et3/3/3)# tx-queue 4
switch(config-if-Et3/3/3-txq-4)#
```

10.1.13.72 tx-queue (FM6000)

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

FM6000 platform switches have eight queues, ***0*** through ***7***. All queues are exposed through the CLI and are user configurable.

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no tx-queue** and **default tx-queue** commands remove the configuration for the specified transmit queue by deleting the all corresponding **tx-queue** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

tx-queue ***queue_level***

Parameters

queue_level The transmit queue. Values range from ***0*** to ***7***.

Commands Available in tx-queue Configuration Mode

- bandwidth percent (FM6000)
- priority (FM6000)
- shape rate (Tx-queue – FM6000)

Guidelines

FM6000 platform switch queues handle unicast and multicast traffic.

Example

This command enters Tx-queue configuration mode for ***transmit queue 3*** of ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# tx-queue 3
switch(config-if-Et5-txq-3)#
```

10.1.13.73 tx-queue (Helix)

The **tx-queue** command places the switch in ***tx-queue*** configuration mode to configure a transmit queue on the configuration mode interface. The ***tx-queue*** configuration mode is not a group change mode;

running-config is changed immediately after commands are executed. The **`exit`** command does not affect the configuration.

Helix platform switches have eight unicast (***UC0 – UC7***) and eight multicast (***MC0 – MC7***) queues. Each UCx-MCx queue set is combined into a single queue group (L1.x), which is exposed to the CLI through this command.

The **`exit`** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **`no tx-queue`** and **`default tx-queue`** commands remove the configuration for the specified transmit queue by deleting the all corresponding **`tx-queue`** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`tx-queue` ***queue_level***

Parameters

queue_level Transmit queue group number. Values range from ***0*** to ***7***.

Commands Available in tx-queue Configuration Mode

- bandwidth guaranteed (Helix)
- shape rate (Tx-queue – Helix)

Guidelines

Helix platform switch queues handle unicast and multicast traffic.

Example

This command enters Tx-queue configuration mode for ***transmit queue 4*** of ***interface ethernet 17/3***.

```
switch(config)# interface ethernet 17/3
switch(config-if-Et17/3)# tx-queue 4
switch(config-if-Et17/3-txq-4)#
```

10.1.13.74 tx-queue (Petra)

The **`tx-queue`** command places the switch in ***tx-queue*** configuration mode to configure a transmit queue on the configuration mode interface. The ***tx-queue*** configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **`exit`** command does not affect the configuration.

Petra platform switches have eight queues, ***0*** through ***7***, and all queues are exposed through the CLI. However, ***queue 7*** is not user-configurable. ***Queue 7*** is always mapped to ***traffic class 7***, which is reserved for control traffic.

The **`exit`** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **`no tx-queue`** and **`default tx-queue`** commands remove the configuration for the specified transmit queue by deleting the all corresponding **`tx-queue`** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`tx-queue` ***queue_level***

Parameters

queue_level The transmit queue. Values range from ***0*** to ***7***.

Commands Available in tx-queue Configuration Mode

- bandwidth percent (Petra)
- priority (Petra)
- shape rate (Tx-queue – Petra)

Guidelines

Petra platform switch queues handle unicast traffic. Queues for multicast traffic are not supported.

Example

This command enters the ***tx-queue*** configuration mode for ***transmit queue 3*** of ***interface ethernet 3/3***.

```
switch(config)# interface ethernet 3/3
switch(config-if-Et3/3)# tx-queue 3
switch(config-if-Et3/3-txq-3)#
```

10.1.13.75 tx-queue (Trident II)

The **`tx-queue`** command places the switch in ***tx-queue*** configuration mode to configure a transmit queue on the configuration mode interface. The ***tx-queue*** configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **`exit`** command does not affect the configuration.

Trident II platform switches have eight unicast (***UC0 – UC7***) and eight multicast (***MC0 – MC7***) queues. Each UCx-MCx queue set is combined into a single queue group (L1.x), which is exposed to the CLI through this command.

The **`exit`** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **`no tx-queue`** and **`default tx-queue`** commands remove the configuration for the specified transmit queue by deleting the all corresponding **`tx-queue`** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`tx-queue` ***queue_level***

Parameters

queue_level Transmit queue group number. Values range from ***0*** to ***7***.

Commands Available in tx-queue Configuration Mode

- bandwidth guaranteed (Trident II)

- shape rate (Tx-queue – Trident II)

Trident II platform switch queues handle unicast and multicast traffic.

Example

This command enters the ***tx-queue*** configuration mode for ***transmit queue 4*** of ***interface ethernet 17/3***.

```
switch(config)# interface ethernet 17/3
switch(config-if-Et17/3)# tx-queue 4
switch(config-if-Et17/3-txq-4)#
```

10.1.13.76 uc-tx-queue

The **uc-tx-queue** command places the switch in the ***uc-tx-queue*** configuration mode to configure a unicast transmit queue on the configuration mode interface. The ***uc-tx-queue*** configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Trident and Tomahawk switches have eight unicast queues (***UC0 – UC7***) and four multicast queues (***MC0 – MC03***), categorized into two priority groups. All queues are exposed through the CLI and are user-configurable.

- **Priority Group 1:** ***UC7***, ***UC6***, ***MC3***
- **Priority Group 0:** ***UC5***, ***UC4***, ***MC2***, ***UC3***, ***UC2***, ***MC1***, ***UC1***, ***UC0***, ***MC0***

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no uc-tx-queue** and **default uc-tx-queue** commands remove the configuration for the specified transmit queue by deleting the all corresponding **uc-tx-queue** mode commands from ***running-config.***

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

uc-tx-queue ***queue_level***

Parameters

queue_level The multicast transmit queue number. Values range from ***0*** to ***7***.

Commands Available in uc-tx-queue Configuration Mode

- bandwidth percent (Trident and Tomahawk)
- priority (Trident and Tomahawk)
- shape rate (Tx-queue – Trident and Tomahawk)

Related Command

mc-tx-queue: Configures multicast transmit queues on Trident and Tomahawk platform switches.

Example

This command enters the ***mc-tx-queue*** configuration mode for ***multicast transmit queue 4*** of ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# uc-tx-queue 4
switch(config-if-Et5-mc-txq-4)#
```

10.1.14 Chipset Mapping for QoS

Table 47: Chipset Names by Model Number

Model Number	Chipset Name
7010T-48	Helix4
7010T-48-DC	Helix4
7020SR-24C2	QumranAX
7020SR-32C2	QumranAX
7020SRG-24C2	QumranAX
7020TR-48	QumranAX
7020TRA-48	QumranAX
7048T-4S	Petra
7048T-A	Petra
7050CX3-32S	Trident3X7
7050CX3M-32S	Trident3X7
7050Q-16	Trident+
7050QX2-32S	Trident2+
7050QX-32	Trident2
7050QX-32S	Trident2
7050S-52	Trident+
7050S-64	Trident+
7050SX-128	Trident2
7050SX2-128	Trident2+
7050SX2-72Q	Trident2+
7050SX3-48C8	Trident3X5
7050SX3-48YC	Trident3X5
7050SX3-48YC12	Trident3X7
7050SX3-96YC8	Trident3X7
7050SX-64	Trident2
7050SX-72	Trident2
7050SX-72Q	Trident2
7050SX-96	Trident2
7050T-36	Trident+
7050T-36 (HwRev4)	Trident+
7050T-52	Trident+
7050T-52 (HwRev4)	Trident+

Model Number	Chipset Name
7050T-64	Trident+
7050T-64 (HwRev4)	Trident+
7050TX-128	Trident2
7050TX2-128	Trident2+
7050TX3-48C8	Trident3X5
7050TX-48	Trident2
7050TX-64	Trident2
7050TX-72	Trident2
7050TX-72Q	Trident2
7050TX-96	Trident2
7060CX2-32S	Tomahawk+
7060CX-32S	Tomahawk
7060DX4-32	Tomahawk3
7060PX4-32	Tomahawk3
7060SX2-48YC6	Tomahawk+
7120T-4S	Bali
7124FX	Bali
7124S	Bali
7124SX	Bali
7140T-8S	Bali
7148S	Bali
7148SX	Bali
7150S-24	Alta
7150S-24-CL	Alta
7150S-52-CL	Alta
7150S-64-CL	Alta
7150SC-24-CLD	Alta
7150SC-64-CLD	Alta
7170-32C	Tofino
7170-32CD	Tofino
7170-64C	Tofino
720XP-24Y6	Trident3X3
720XP-24ZY4	Trident3X3
720XP-48Y6	Trident3X3

Model Number	Chipset Name
720XP-48ZC2	Trident3X3
720XP-96ZC2	Trident3X3
7250QX-64	Trident2
7260CX3-64	Tomahawk2
7260CX3-64E	Tomahawk2
7260CX-64	Tomahawk
7260QX-64	Tomahawk
7280CR2-60	Jericho+
7280CR2A-60	Jericho+
7280CR2AK-30	Jericho+
7280CR2K-30	Jericho+
7280CR2K-60	Jericho+
7280CR2M-30	Jericho+
7280CR3-32D4	Jericho2
7280CR3-32P4	Jericho2
7280CR3-96	Jericho2
7280CR3K-32D4	Jericho2
7280CR3K-32P4	Jericho2
7280CR3K-96	Jericho2
7280CR3MK-32D4	Jericho2
7280CR3MK-32P4	Jericho2
7280CR-48	Jericho
7280DR3-24	Jericho2
7280DR3K-24	Jericho2
7280PR3-24	Jericho2
7280PR3K-24	Jericho2
7280QRA-C36S	Jericho
7280QR-C36	QumranMX
7280QR-C72	Jericho
7280SE-64	Arad+
7280SE-68	Arad+
7280SE-72	Arad+
7280SR2-48YC6	Jericho+
7280SR2A-48YC6	Jericho+

Model Number	Chipset Name
7280SR2K-48C6	Jericho+
7280SR3-40YC6	Jericho2C Q2A
7280SR-48C6	QumranMX
7280SRA-48C6	QumranMX
7280SRAM-48C6	QumranMX
7280SRM-40CX2	QumranMX
7280TR3-40C6	Jericho2C Q2A
7280TR-48C6	QumranMX
7280TRA-48C6	QumranMX
DCS-7304	Trident3
DCS-7304	Tomahawk
DCS-7304	Trident2
DCS-7308	Trident3
DCS-7308	Tomahawk
DCS-7308	Trident2
DCS-7316	Trident2
DCS-7368X4	Tomahawk3
DCS-7504	Petra
DCS-7504E	Arad/Arad+
DCS-7504N	Jericho 2
DCS-7504N	Jericho/Jericho+
DCS-7508	Petra
DCS-7508E	Arad/Arad+
DCS-7508N	Jericho 2
DCS-7508N	Jericho/Jericho+
DCS-7512N	Jericho 2
DCS-7512N	Jericho/Jericho+
DCS-7516N	Jericho/Jericho+
DCS-7804-CH	Jericho 2
DCS-7808-CH	Jericho 2

10.2 Traffic Management

This chapter describes Arista's Traffic Management, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Traffic Management Conceptual Overview
- Traffic Management Configuration Arad Platform Switches
- Traffic Management Configuration FM6000 Platform Switches
- Traffic Management Configuration Petra Platform Switches
- Traffic Management Configuration Trident Platform Switches
- Traffic Management Configuration Trident II Platform Switches
- Traffic Management Configuration Commands

10.2.1 Traffic Management Conceptual Overview

Traffic is managed through policy maps that apply data shaping methods to specific data streams. A policy map is a data structure that identifies specific data streams and then defines shaping parameters that modify packets within the streams. The switch defines four types of policies:

- Control Plane Policies: Control plane policy maps are applied to the control plane.
- QoS Policies: QoS policy maps are applied to Ethernet and port channel interfaces.
- Segment Routing Traffic Engineering Policy (SR-TE).
- PBR Policies: PBR policy maps are applied to Ethernet interfaces, port channel interfaces and switch virtual interfaces (SVIs).

A policy map consists of classes. Each class contains an eponymous class map and traffic resolution commands.

- A class map is a data structure that defines a data stream by specifying characteristics of data packets that comprise that stream. Each class map is typed as either QoS, control plane, or PBR and is available only to identically typed policy maps.
- Traffic resolution commands specify data handling methods for traffic that matches a class map. Traffic resolution options vary by policy map type.

Data packets that enter an entity to which a policy map is assigned are managed with traffic resolution commands of the first class that matches the packets.

10.2.1.1 Control Plane Policies

The switch defines one control plane policy map named ***copp-system-policy***. The ***copp-system-policy*** policy map is always applied to the control plane and cannot be removed from the switch. Other control plane policy maps cannot be added. **Copp-system-policy** consists of preconfigured classes, each containing a static class map and traffic resolution commands. Preconfigured classes cannot be removed from ***copp-system-policy***.

Static class maps are provided by the switch and cannot be modified or deleted. The naming convention of static class maps is ***copp-system-*** *name*, where *name* differentiates the class maps. Static class maps have pre-defined internal conditions, are not based on ACLs, and are only listed in ***running-config*** as components of ***copp-system-policy***. The sequence of static class maps in the policy map is not significant. Traffic resolution commands define minimum (bandwidth) and maximum (shape) transmission rates for data streams matching the corresponding class map.

Copp-system-policy can be modified through the following steps:

1. Add classes consisting of an eponymous dynamic class map and traffic resolution commands.

   Dynamic class maps are user created, can be edited or deleted, filter traffic with a single IPv4 ACL, and are listed in ***running-config***.
2. Change traffic resolution commands for a preconfigured class.

These sections describe control plane traffic policy configuration procedures:

- Configuring Control Plane Traffic Policies Arad Platform Switches
- Configuring Control Plane Traffic Policies FM6000 Platform Switches

- Configuring Control Plane Traffic Policies Petra Platform Switches
- Configuring Control Plane Traffic Policies Trident Platform Switches

10.2.1.2 QoS Policies

QoS policy maps are user defined. The switch does not provide preconfigured QoS policy maps and in the default configuration, policy maps are not applied to any Ethernet or port channel interface. Policy maps and class maps are created and applied to interfaces through configuration commands.

A QoS policy map is composed of one or more classes. Each class contains an eponymous dynamic class map and traffic resolution commands. Dynamic class maps are user created, can be edited or deleted, filter traffic with a single IPv4 ACL, and are listed in ***running-config***.

QoS traffic resolution commands perform one of the following:

- Set the Layer 2 CoS field
- Set the DSCP value in the ToS byte
- Specify a traffic class queue

The last class in all QoS policy maps is ***class-default***, which is composed as follows:

- The ***class-default*** class map matches all traffic except IPv4 or IPv6 traffic and is not editable.
- By default, ***class-default*** class contains no traffic resolution commands. Traffic resolution commands can be added through configuration commands.

Data packets that enter an interface to which a policy map is assigned are managed with traffic resolution commands that correspond to the first class that matches the packet.

These sections describe QoS traffic policy configuration procedures:

- Configuring QoS Traffic Policies Arad Platform Switches
- Configuring QoS Traffic Policies FM6000 Platform Switches
- Configuring QoS Traffic Policies Petra Platform Switches
- Configuring QoS Traffic Policies Trident Platform Switches

10.2.1.3 Segment Routing Traffic Engineering Policy (SR-TE)

Segment Routing Traffic Engineering Policy (SR-TE) policy uses Segment Routing (SR) to enable a headend to steer traffic along any path without maintaining per flow state in every node based on the policy. Configuring SR policy for the MPLS dataplane (SR-MPLS) for Type-1 SR policy segments with BGP and locally configured policies as sources of SR policy is available on DCS-7500 and DCS-7280 family of switches.

SR Policy Overview
SR Policy Identification

The following identifies an SR policy.

- **Endpoint**: an IPv4 or IPv6 address which refers to the destination of the policy. (0/0, 0:: are allowed and called “null endpoints”)
- **Color**: an unsigned 32-bit opaque numerical quantity. The semantic of a color is up to the operator. It can refer to, for instance, an application or a type of traffic (low latency) or a geographical location, etc.

SR Policy Constituents

The SR policy consists of *candidate paths*. Each candidate path has the following.

- **SID-lists (SLs)**: an ordered list of Segment Identifiers. (Each SID is a MPLS label in the MPLS instantiation of SR). An SL encodes one path from the headend to the destination. Each SL has an optional weight attached to it for the purpose of Unequal Cost Multipath (UCMP) traffic distribution. The default value for SL weight is ***1***.

- **Preference**: an optional, unsigned 32-bit integer used in the candidate path selection algorithm to select the “active” candidate path. The default value for preference is ***100***.
- **Binding SID (BSID)**: an optional, SID.



Note: In EOS, a BSID is mandatory for each candidate path.

SR Policy Sources

A headend learns SR policies using the following.

- **BGP**
  - Single agent routing model (Ribd)
  - Multi-agent routing model
- **Local configuration using CLI**
  - Single agent routing model (Ribd)
  - Multi-agent routing model
  - Openconfig YANG models
- **PCEP**



Note: PCEP is not supported in EOS.

Identity of a Candidate Path

A candidate path within an SR policy is identified by a 3-tuple of {Protocol-Origin, Originator, Discriminator}. In EOS, for locally configured policies:

- the ASN in the Originator is set to 0.
- the node address in the Originator is set to ***0.0.0.0***.
- discriminator is set to the Preference configured.



Note: EOS CLI allows configuring only one candidate path at a given preference and does not allow configuring the discriminator for a candidate path.

State of an SID List (SL)

The following describes the state of an SL.

- **Valid**: the top label of the SL is resolvable within the LFIB to outgoing next hop(s), interface(s) and a label action.
- **Invalid**: top label of the SL is not resolvable to outgoing next hop(s), interface(s) and a label action. An SL is also marked as invalid when the SL is resolvable, but the resolved labeled stack exceeds the platform’s maximum SID depth (SID), that is, exceeds the maximum number of labels the platform can push in to the outgoing packet.



Note: The state is either valid or invalid.

State of a Candidate Path

The following describes the states of a candidate path.

- **Invalid**: not eligible to participate in the best/active candidate path selection algorithm because of one of the reasons below.
  - all constituent SLs are invalid
  - Binding SID is not present
  - Binding SID is present but outside SRLB range
- **Valid**: At least one SL is valid and has lost out to some other candidate path in the best / active candidate path selection algorithm.

- **Active**: The candidate path is valid and is the winner in the best / active candidate path selection algorithm. The active candidate path is installed in the switch hardware and used for forwarding traffic.

State of an SR Policy

An SR policy is “valid” when at least one of its candidate paths is valid as described above. Otherwise, the SR policy is said to be “invalid”.

Resolution of an SL

An SL is resolved if the top label (first SID) can be resolved in the system Labeled FIB (LFIB) to yield a nexthop and outgoing interface(s). The other labels in the SID-List do not play a part in resolution.

Best Candidate Path (Active Candidate Path) Selection Algorithm

EOS overrides selection based on discriminator by retaining the current active candidate path even when current active path has a lower discriminator value. This reduces the active path flap when a new path is learnt of the same significance. The following is a summary of valid candidate paths ordering for a given policy.

1. Candidate path with higher preference is chosen.
2. Locally configured candidate path is chosen over BGP learnt path
3. Lower originator is chosen.
   a. Lower AS number of Originator field is chosen
   b. Lower Node address of Originator field is chosen
4. Current active candidate path is chosen

The following displays the reason for a path not getting selected as an active path for a specified policy.

```
switch# show traffic-engineering segment-routing policy endpoint <endpoint>
 color <color>
```

Binding SID

The use cases for Binding SID are the following.

- Stitch together multiple domains
- Stitch together different traffic tunnels
- Overcome label stack imposition limitation in hardware.

BSID Conflict Handling

Examples

1. **(Between Policies)**: If the policy (E1, C1) becomes eligible to be active first then it would be installed in the LFIB and the policy (E2,C2) whose best path(CP1) is in conflict with the Policy (E1, C1) and will not become active.
   - Policy(E1, C1): CP1: Binding-SID 965536 (wins best path)
   - Policy(E2, C2): CP1: Binding-SID 965536 (wins best path)
   - CP2: Binding-SID 965537
2. **(with other Application)**: The SR-TE policies have the lowest preference when there is a conflict with any other application in EOS using the SRLB range. The candidate paths which have the binding-SID as that of an LFIB entry by another application (for example, static adjacency segment) will be kept as “invalid”.

In both the cases, when the conflict no longer exists, the candidate paths are re-evaluated and may become active.

BGP as a Source of Policies

SR Policies from a BGP peer (a controller, route reflector) is received for installation at the headend by EOS. It does not propagate the received policies to BGP peers nor does it originate SR Policies for transmission to BGP peers.

The following are supported over IPv4 or IPv6 peers which can be single hop or multi-hop iBGP or eBGP peers.

1. **SAFI 73 for AFI 1 and AFI 2**: IPv4 and IPv6 policy endpoints, with the encoding defined in section 2.1 of ***Advertising Segment Routing Policies in BGP***.

   Note: The nexthop address-family must match the AFI of the NLRI.

2. **Sub-TLVs of Tunnel Encapsulation TLV of type 15 (SR-TE Policy Type) of the Tunnel Encapsulation Path Attribute**:

   a. Preference (Sub-TLV Type 12)
   b. Binding SID (Sub-TLV Type 13) of length ***2*** or ***6*** bytes
   c. Segment List (Sub-TLV Type 128). The following Segment List sub-TLVs are supported:

      1. Type 1 Segment (Sub-TLV type 1)
      2. Weight (Sub-TLV type 9)

   d. Explicit NULL Label Policy (Sub-TLV Type 14)

Note: All other sub-TLVs of the Tunnel Encapsulation TLV and Segment List sub-TLVs are ignored.

Route-Target and NO_ADVERTISE Community in SR-TE SAFI Updates

EOS implements the Acceptance and Usability checks as defined in sections 4.2.1 and 4.2.2 of the IETF draft Advertising Segment Routing Policies in BGP. However EOS skips matching the Route-Target with the router-ID of the headend if the SR-TE NLRI is tagged with **NO_ADVERTISE** community.

ECMP is not supported for SR-TE SAFI Paths

EOS does not support ECMP for BGP SR-TE SAFI. Only one best candidate path is chosen by BGP path selection and published to SR-TE Policy Agent for candidate path selection.

Note: ECMP of BGP next hops where each next hop resolves to an SR-TE policy is supported.

Path Selection within BGP

The IETF draft **Advertising Segment Routing Policies in BGP** supports passing multiple candidate paths from a single protocol source for an SR-TE policy path selection. Hence it includes a field distinguisher in the NLRI which can be unique for each controller to make BGP pass through the policies. However when multiple sources use the same distinguisher, BGP performs a path selection for the tuple: Endpoint, Color and Distinguisher. The best path for that tuple is published to the SR-TE Policy Agent for selecting an Active path. The best **bgp-best-path** selection applies to SR-TE SAFI as well.

Error Handling / Edge Cases

- **Weight 0**: The IETF draft does not limit the range of SL weight to exclude weight 0. A SID-List with weight 0 is not used for forwarding so BGP module in EOS does not pass on SID-Lists with weight 0 to the SR-TE policy agent. Such SID-Lists will be visible in **`show bgp sr-te`** commands but not in **`show traffic-engineering segment-routing policy`** commands.
- **Empty SLs**: Given the TLV encoding used to propagate SR Policies in BGP, it is possible to receive SID-Lists that have no SIDs (SID-Lists are empty). The BGP module in EOS does not pass such empty SID-Lists to SR-TE policy agent. Such SID-Lists will be visible in **`show bgp sr-te`** commands but not in **`show traffic-engineering segment-routing policy`** commands.
- **Non Type 1 segments**: EOS supports only Type-1 segments. If a BGP update is received with a SID-List that has non Type-1 segments, the entire SID-List is ignored and **`BGP-4-SRTE_IGNORED_SEGMENT_LIST_UNSUPPORTED_SEGMENTS`** syslog is emitted. Such SID-Lists are not stored locally, and **`show bgp sr-te`** command does not display them.

  Note: The SID-Lists that are made up of all Type-1 segments are passed on to the SR-TE policy agent.

Steering Traffic into a Policy

Incoming label is BSID - “Labelled Steering”

At the headend when a packet is received with a label stack that has a BSID of an active CP of a valid SR Policy as the top label, the headend pops the label, and imposes the resolved label stack on the outgoing packet.

Example

Consider an SR Policy with an active candidate path with BSID ***965536*** and SL with label stack [***965540***, ***900001***, ***900002***]. Assume that ***965540*** is an IS-IS SR Adjacency SID. An incoming packet has label stack [***965536***, ***100000***] then the outgoing label stack is [***900001***, ***900002***, ***100000***].

Steering BGP learnt IP(v6) prefixes - “IP Steering”

Incoming label is BSID - “Labelled Steering”

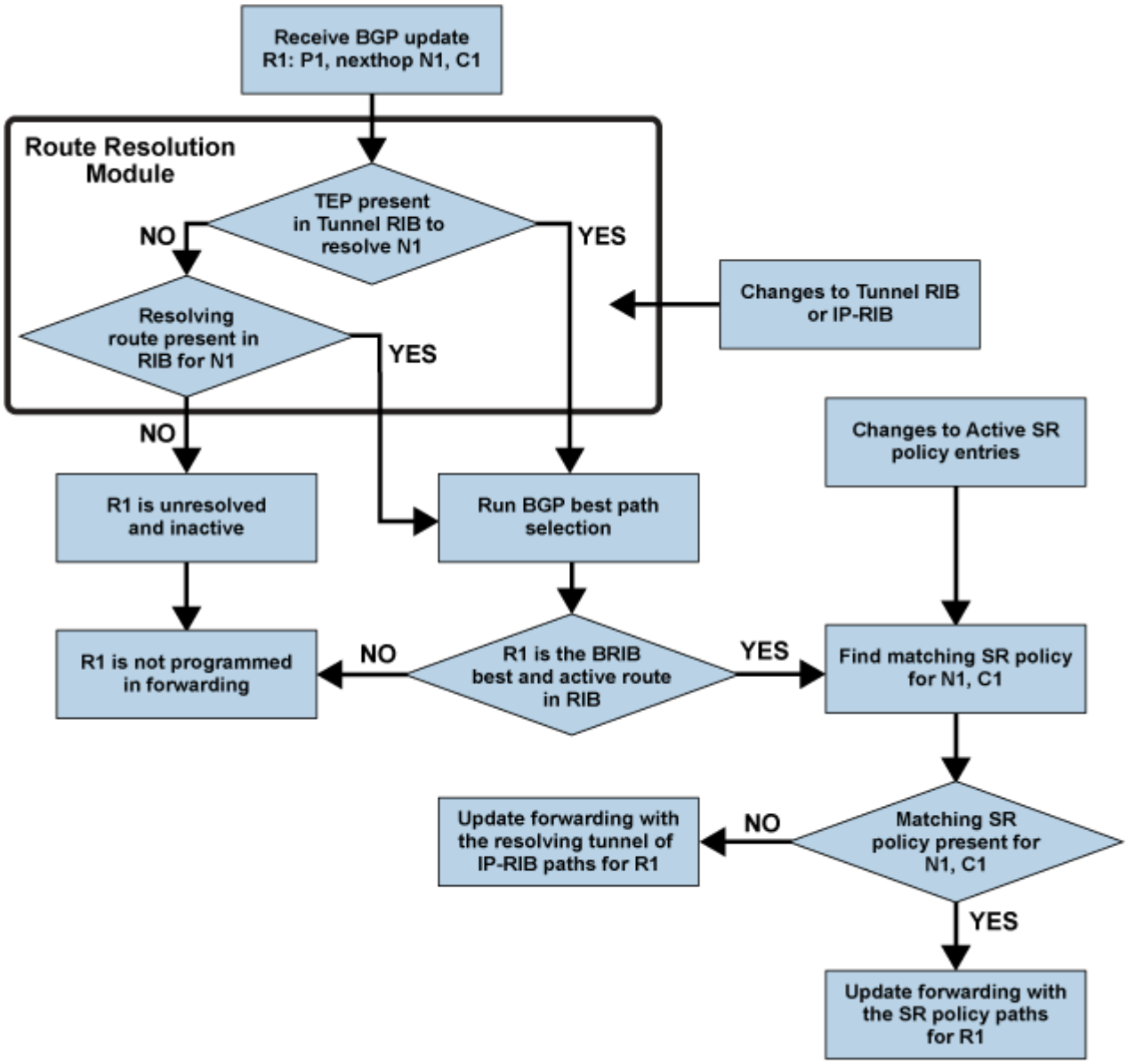
At the headend, BGP IPv4 and IPv6 routes received with one or more extended color communities are recursively resolved through any active SR Policy that matches the BGP routes’ nexthop and color. When an IPv4 or IPv6 packet is received that is forwarded using this policy, the SL’s resolved label stack is imposed in the outgoing packet.

For BGP routes received with color community to be steered via an SR policy, the route’s nexthop must already be resolvable through IGP. If there is no resolving route in IGP, the route is considered unresolvable and will not be programmed in hardware even if there is a matching SR policy for the corresponding nexthop and color.

If there is no matching SR policy for the received BGP nexthop and color, the route will be resolved through the IGP route in IP RIB. If an active SR policy that matches the BGP nexthop and color gets instantiated at a later time, the BGP route will change from resolving through IGP to the new active SR policy.



Note: The recursion through SR policy is only applicable for BGP routes that are active in RIB.



Color only IP steering using CO bits

It is possible to relax the requirement of an exact match of the BGP route's nexthop with the endpoint of the SR Policy using the "CO" (Color Only) bits in the color extended community. The "CO" bits are 2 reserved bits repurposed for color only steering as defined in section 3 of Advertising Segment Routing Policies in BGP. The exact match of the nexthop is done with the CO bits set to 00 or 11.

CO = 01 Steering: relaxes the nexthop to match the null endpoint of a policy. For a BGP route with nexthop N and color C, the following order is used for resolution. If there is no IGP route resolving the BGP nexthop, the route is not programmed in hardware.

1. Active SR policy with endpoint N and color C
2. Active SR policy with null endpoint (from the same AFI as the BGP route) and color C
3. Active SR policy with null endpoint from any AFI and color C
4. IGP route

CO = 10 Steering: in addition to the steps in CO = ***01*** steering, CO = ***10*** additionally relaxes the nexthop to match *any* endpoint. The following order is used for resolving a BGP route with nexthop N and color C. The behavior described is in accordance with section 8.8.1 of the IETF draft Segment Routing Policy for Traffic Engineering.

1. Active SR policy with endpoint N and color C
2. Active SR policy with null endpoint (from the same AFI as the BGP route) and color C
3. Active SR policy with null endpoint from any AFI and color C
4. Active SR policy for any endpoint from the same AFI as the BGP route and color C
5. Active SR policy for any endpoint from any AFI and color C
6. IGP route

ECMP of IPv4/IPv6 Prefixes that Resolve over SR-TE Policies

When multiple BGP paths of BGP unicast prefixes resolve through active SR policies form ECMP, the resulting FIB entry for the BGP route has an ECMP of segment list paths which is a union of all the segments-list entries present in each of the resolving SR policies for the BGP paths.

Example

The following table displays four paths for prefix ***192.1.0.0/31***, and each of the four paths resolves via SR-TE policies.

Table 48: List of Paths Resolved via SR-TE Policies

Path	Nexthop	Color	Policy EP	Policy Color	Segment Lists	Per SL Traffic Distribution
1	1.0.0.2	CO(00):1000	1.0.0.2	1000	[2500 500], Weight: 1 [2501 500], Weight: 2	8.33% 16.66%
2	1.0.2.2	CO(00):2000	1.0.2.2	2000	[2502 500], Weight: 1 [2503 500], Weight: 1	12.5% 12.5%
3	1.0.4.2	CO(00):3000	1.0.4.2	3000	[2504 500], Weight: 1 [2505 500], Weight: 1	12.5% 12.5%
4	1.0.6.2	CO(00):4000	1.0.6.2	4000	[2506 500], Weight: 1 [2507 500], Weight: 1	12.5% 12.5%

```
B I    192.1.0.0/31 [200/0] via SR-TE Policy 1.0.4.2, color 3000
                              via SR-TE tunnel index 6, weight 1
                                 via 1.0.4.2, Ethernet1, label 2505 500
                              via SR-TE tunnel index 5, weight 1
                                 via 1.0.4.2, Ethernet1, label 2504 500
                           via SR-TE Policy 1.0.0.2, color 1000
                              via SR-TE tunnel index 2, weight 1
                                 via 1.0.0.2, Ethernet2, label 2501 500
                              via SR-TE tunnel index 1, weight 1
                                 via 1.0.0.2, Ethernet2, label 2500 500
                           via SR-TE Policy 1.0.2.2, color 2000
                              via SR-TE tunnel index 4, weight 1
                                 via 1.0.2.2, Ethernet3, label 2503 500
```

```
                                via SR-TE tunnel index 3, weight 1
                                   via 1.0.2.2, Ethernet3, label 2502 500
                             via SR-TE Policy 1.0.6.2, color 4000
                                via SR-TE tunnel index 8, weight 1
                                   via 1.0.6.2, Ethernet6, label 2507 500
                                via SR-TE tunnel index 7, weight 1
                                   via 1.0.6.2, Ethernet6, label 2506 500
```

The traffic distribution honors the weights of the SID-Lists. In the example, each of the four SR Policies will get ***25%*** of the total traffic meant for prefix ***192.1.0.0/31***. Within each policy, the distribution is based on the weights of the SID-Lists.

ECMP Group when some BGP unicast paths resolve over SR Policies and some via non SR Policy IGP paths

If some BGP paths resolve via SR Policy paths and some BGP paths resolve via non SR Policy IGP, then the ECMP group formed programmed as the active route in FIB, only considers the SR Policy paths. ECMP in the FIB is not formed between paths that resolve over SR Policy and paths that resolve via non SR Policy IGP routes. In the example above, if SR Policy with endpoint ***1.0.6.2*** and color ***4000*** becomes inactive or is removed, the FIB path for ***192.1.0.0/31*** resolves via 3 SR Policies as shown below.

```
B I    192.1.0.0/31 [200/0] via SR-TE Policy 1.0.4.2, color 3000
                               via SR-TE tunnel index 6, weight 1
                                  via 1.0.4.2, Ethernet1, label 2505 500
                               via SR-TE tunnel index 5, weight 1
                                  via 1.0.4.2, Ethernet1, label 2504 500
                            via SR-TE Policy 1.0.0.2, color 1000
                               via SR-TE tunnel index 2, weight 1
                                  via 1.0.0.2, Ethernet2, label 2501 500
                               via SR-TE tunnel index 1, weight 1
                                  via 1.0.0.2, Ethernet2, label 2500 500
                            via SR-TE Policy 1.0.2.2, color 2000
                               via SR-TE tunnel index 4, weight 1
                                  via 1.0.2.2, Ethernet3, label 2503 500
                               via SR-TE tunnel index 3, weight 1
                                  via 1.0.2.2, Ethernet3, label 2502 500
```

Note: **`show ip bgp`** still shows a 4-way ECMP. The FIB paths switch to resolving via the (non SR Policy) IGP paths when there are no BGP paths in the ECMP group that resolve via an SR Policy.

UCMP of IPv4/IPv6 prefixes using LinkBandwidth (LBW) Extended Community that resolve over SR-TE policies not supported

When multiple BGP paths of BGP unicast prefixes resolve through active SR policies form ECMP, and the unicast paths also contain the LBW extended community, EOS does not form UCMP amongst the unicast paths. Only ECMP is formed at the unicast prefix level. The LBW is ignored the behavior is identical to the behavior explained in the previous section.

Resolution of BGP unicast prefixes that resolve over other BGP unicast prefixes resolved via SR Policies

A BGP unicast prefix P1, that is recursively resolved via another BGP prefix P2, such that P2 resolves via an SR Policy, then in the FIB, P1 is programmed with the resolved nexthop pointing to the non SR Policy resolution of P2. P1 does not use P2s SR Policy for forwarding.

Explicit Null Label Imposition

When the address family of the BGP unicast prefix is not the same as the address family of the endpoint of the SR Policy that the unicast prefixes resolves via, an explicit null label is automatically imposed in the outgoing label stack.

Example

If an IPv4 unicast prefix ***P1*** resolves over a policy whose endpoint ***EP1*** is an IPv6 address (this can happen due to color only CO=01/10 steering with ***P1*** having an IPv4 nexthop) and the SR Policy had a SID-List whose resolved label stack is [***1001***, ***1002***, ***1003***], the outgoing packet is imposed with [***1001***, ***1002***, ***1003***, ***2***] where ***0*** is the IPv4 explicit null label.

If an IPv6 prefix ***P2***, resolves over a policy whose endpoint ***EP2*** is an IPv4 address (this can happen with color only CO=01/10 steering with ***P2*** having a IPv6 nexthop) and the SR Policy had a SID-List whose resolved label stack is [***1001***, ***1002***, ***1003***], the outgoing packet is imposed with [***1001***, ***1002***, ***1003***, ***2***] where ***2*** is the IPv6 explicit null label.

The following table lists the configurations which result in having explicit-null label in the resolved label stack.

Table 49: Configurations resulting in Explicit-Null Label in Resolved Label Stack

ENLP configuration for the resolving SR Policy	IPv4 Prefixes	IPv6 Prefixes
None	-	-
IPv4	IPv4 explicit null appended to the end of label stack	-
IPv6	-	IPv6 explicit null appended to the end of label stack
Both	IPv4 explicit null appended to the end of label stack	IPv6 explicit null appended to the end of label stack
No/Default config (incase of BGP learnt policies ENLP Sub-TLV is not received)	Resolving SR Policy has IPv4 Endpoint address: No explicit-null	Resolving SR Policy has IPv4 Endpoint address: IPv6 explicit null appended to the end of label stack
	Resolving SR Policy has IPv6 Endpoint address: IPv4 explicit null appended to the end of label stack	Resolving SR Policy has IPv6 Endpoint address: No explicit-null

Traffic Accounting

All egress tunnel counters (MPLS/GRE/MPLSoGRE using SR-TE/Nexthop-group/BGP-LU tunnel types) share the same hardware resource.

- **7280E/7500E systems**: Up to ***16k*** tunnels
- **7280R/7500R systems**: Up to ***8k*** tunnels

Tunnel counters are allocated on a first-come, first-served basis. Configurations using GRE/MPLSoGRE, GRE, and MPLS further limit a maximum of 4k countable egress MPLS tunnels on 7280R/7500R.

FEC Optimizations

The hardware FEC usage could be reduced as the underlying FEC is shared among different routes.

- Programming of the active candidate path of an SR-TE policy in hardware is shared between the BSID route and IP steering route.
- If all of the following conditions are met, ISIS-SR MPLS routes and tunnel entries directly point to the next hop FEC generated by the routing agent (IGP FEC).

  - All the next hops of the MPLS route either point to pop or forward (i.e. swapping to the same label) label action.
  - The switch is either a 7280 or a 7500 platform.
- The corresponding SR-TE policy BSID routes (and corresponding Segment List tunnels) that resolve over ISIS-SR MPLS routes, will directly point to the IGP FEC.

Configuring SR-TE

The following commands start the **SrTePolicy** agent and enter the switch into the Traffic Engineering configuration sub-mode.

```
switch(config)# router traffic-engineering
switch(config-te)# segment-routing
```

Note: The agent must be running even if the only source of policies is BGP.

Static Policy Configuration

The following commands set the policy using endpoint and color value, and define the BSID for the policy.

```
switch(config-te-sr)# policy endpoint v4Address|v6Address color color-value
switch(config-te-sr-policy)# binding-sid mpls-label
switch(config-te-sr-policy)# path-group preference value
```

The following commands enter the policy path configuration sub mode, and adds a segment list to the candidate path.

```
switch(config-te-sr-policy)# path-group preference value
switch(config-te-sr-policy-path)# segment-list label-stack label1 label2 …
 weight value
```

Note: The default weight value is ***1***. Adding weight is optional. Repear the configuration statement for multiple segment lists per candidate path.

The following commands configures a null label policy.

```
switch(config-te-sr-policy-path)# explicit-null [none|ipv4|ipv6|both]
```

Note: The null label policy configuration is optional.

BGP configuration for SR-TE SAFI

The following commands configures a BGP router to activate a neighbor to negotiate and accept SR-TE address-family with this peer.

```
switch(config)# router bgp <as>
switch(config-router-bgp)# address-family ipv4|ipv6 sr-te
switch(config-router-bgp-af-srte)# neighbor neighbor activate
```

The following command configures an inbound route-map to filter or modify attributes on incoming SR-TE prefixes from the peer.

```
switch(config-router-bgp-af-srte)# neighbor neighbor route-map routeMapName in
```

Configuring Egress SR-TE Traffic Accounting

The following command enables egress traffic accounting for SR policies (also known as MPLS tunnels).

```
switch(config)# hardware counter feature mpls tunnel
```

The following command displays current status of the MPLS counters.

```
switch#show hardware counter feature
Feature             Direction        Counter Resource (Engine)
------------------- ---------------- -------------------------
ACL-IPv4            out              Jericho: 2, 3
ACL                 in               Jericho: 4, 5, 6, 7
MPLS tunnel         out              Jericho: 8, 9
```

The following command disables egress traffic accounting for SR policies.

```
switch(config)# no hardware counter feature mpls tunnel
```

The following command displays a summary information of SR-TE SAFI.

```
switch# show bgp sr-te summary
BGP summary information for VRF default
Router identifier 100.1.1.2, local AS number 100
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State  PfxRcd PfxAcc
  100.1.1.1        4  100              407       413    0    0 00:18:57 Estab  1      1
  1000::1          4  100              407       413    0    0 00:18:57 Estab  1      1
```

The following command displays a summary information of candidate paths received from neighbors which have negotiated AFI=1 for SR-TE SAFI.

```
switch# show bgp sr-te ipv4 summary
BGP summary information for VRF default
Router identifier 100.1.1.2, local AS number 100
Neighbor Status Codes: m - Under maintenance
  Neighbor    V  AS   MsgRcvd  MsgSent InQ OutQ  Up/Down  State  PfxRcd PfxAcc
  100.1.1.1   4  100  407      413     0   0     00:18:57 Estab  0      0
```

The following command displays a summary information of candidate paths received from neighbors which have negotiated AFI=2 for SR-TE SAFI.

```
switch# show bgp sr-te ipv6 summary
BGP summary information for VRF default
Router identifier 100.1.1.2, local AS number 100
Neighbor Status Codes: m - Under maintenance
  Neighbor     V  AS   MsgRcvd  MsgSent InQ OutQ  Up/Down  State  PfxRcd PfxAcc
  1000::1      4  100  407      413     0   0     00:18:57 Estab  0      0
```

The following command displays all the SR-TE candidate paths.

```
switch# show bgp sr-te
BGP routing table information for VRF default
Router identifier 100.1.1.1, local AS number 100
Policy status codes: * - valid, > - active, E - ECMP head, e - ECMP
                     c - Contributing to ECMP
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

     Endpoint    Color   Distinguisher   Next Hop    Metric  LocPref Weight  Path
*>   133.1.1.1   0       1               130.1.1.3   0       100     0       ?
*>   133.1.1.1   0       2               130.1.1.3   0       100     0       ?
*>   1330::1     0       1               1300::3     0       100     0       ?
*>   1330::1     0       2               1300::3     0       100     0       ?
```

The following command displays all the SR-TE candidate paths with IPv4 endpoints.

```
switch# show bgp sr-te ipv4
BGP routing table information for VRF default
Router identifier 100.1.1.1, local AS number 100
Policy status codes: * - valid, > - active, E - ECMP head, e - ECMP
                     c - Contributing to ECMP
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

     Endpoint   Color   Distinguisher   Next Hop    Metric  LocPref Weight  Path
*>   133.1.1.1  0       1               130.1.1.3   0       100     0       ?
*>   133.1.1.1  0       2               130.1.1.3   0       100     0       ?
```

The following command displays all the SR-TE candidate paths with IPv6 endpoints.

```
switch# show bgp sr-te ipv6
BGP routing table information for VRF default
Router identifier 100.1.1.1, local AS number 100
Policy status codes: * - valid, > - active, E - ECMP head, e - ECMP
                     c - Contributing to ECMP
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

     Endpoint   Color   Distinguisher   Next Hop   Metric  LocPref Weight  Path
*>   1330::1    0       1               1300::3    0       100     0       ?
*>   1330::1    0       2               1300::3    0       100     0       ?
```

The following command displays information about a specific candidate path.

```
switch# show bgp sr-te endpoint 133.1.1.1 color 0 distinguisher 1
BGP routing table information for VRF default
Router identifier 100.1.1.1, local AS number 100
BGP routing table entry for Endpoint: 133.1.1.1 Color: 0 Distinguisher: 1
 Paths: 1 available
  Local
    130.1.1.3 from 100.1.1.2 (100.1.1.2)
      Origin INCOMPLETE, metric 0, localpref 100, IGP metric 0, weight 0,
      received 00:01:29 ago, valid, internal, best
      Community: no-advertise
      Rx SAFI: SR TE Policy
```

The following command displays information about a specific candidate path including the contents of the Tunnel encapsulation path attribute TLV of type SR policy.

```
switch# show bgp sr-te endpoint 133.1.1.1 color 0 distinguisher 1 detail
BGP routing table information for VRF default
Router identifier 100.1.1.1, local AS number 100
BGP routing table entry for Endpoint: 133.1.1.1 Color: 0 Distinguisher: 1
 Paths: 1 available
  Local
    130.1.1.3 from 100.1.1.2 (100.1.1.2)
      Origin INCOMPLETE, metric 0, localpref 100, IGP metric 0, weight 0,
      received 00:01:29 ago, valid, internal, best
      Community: no-advertise
      Rx SAFI: SR TE Policy
      Tunnel encapsulation attribute: SR Policy
         Preference: 200
         Binding SID: 965536
         Explicit null label policy: IPv4
         Segment-List: Label Stack: [ 16004 16003 ], Weight: 10
         Segment-List: Label Stack: [ 2000 3000 ]
```

The following command displays information about SR candidate paths received from the specified neighbor. The “policies” keyword displays only the candidate paths that are accepted. “received-policies” additionally also displays the rejected candidate paths.

```
switch# show bgp neighbors 100.1.1.2 ipv4 sr-te policies
BGP routing table information for VRF default
Router identifier 100.1.1.1, local AS number 100
Policy status codes: * - valid, > - active
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

     Endpoint   Color   Distinguisher   Next Hop    Metric  LocPref Weight  Path
*>   133.1.1.1  0       1               133.1.1.3   0       100     0       ?
*>   133.1.1.1  0       2               133.1.1.3   0       100     0       ?
```

The following command displays information about SR candidate paths received from the specified neighbor along with the contents of the Tunnel Encapsulation path attribute’s TLV of type SR Policy. The **policies** keyword displays only the candidate paths that are accepted. **received-policies** additionally also displays the rejected candidate paths..

```
switch# show bgp neighbors 100.1.1.2 ipv4 sr-te policies detail
BGP routing table information for VRF default
Router identifier 100.1.1.1, local AS number 100
BGP routing table entry for Endpoint: 133.1.1.1 Color: 0 Distinguisher: 2
 Paths: 1 available
  Local
    130.1.1.3 from 100.1.1.2 (100.1.1.2)
      Origin INCOMPLETE, metric 0, localpref 100, IGP metric 0, weight 0,
      received 00:01:29 ago, invalid, internal
      Rx SAFI: SR TE Policy
      Tunnel encapsulation attribute: SR Policy
         Preference: 200
         Binding SID: 965536
         Explicit null label policy: IPv4
         Segment-List: Label Stack: [ 16004 16003 ], Weight: 10
         Segment-List: Label Stack: [ 2000 3000 ]
```

10.2.1.4 PBR Policies

Policy-Based Routing (PBR) allows the operator to specify the next hop for selected incoming packets on an L3 interface, overriding the routing table. Incoming packets are filtered through a policy map referencing one or more ACLs, and matching packets are routed to the next hop specified.

A PBR policy map is composed of one or more classes and can include next-hop information for each class. It can also include single-line raw match statements, which have the appearance and function of a single line from an ACL. Each class contains an eponymous class map. Class maps are user-created, can be edited or deleted, filter traffic using IPv4 ACLs, and are listed in ***running-config***.

These sections describe PBR policy configuration procedures:

- Configuring PBR Policies Arad Platform Switches
- Configuring PBR Policies FM6000 Platform Switches
- Configuring PBR Policies Petra Platform Switches
- Configuring PBR Policies Trident Platform Switches

10.2.2 Traffic Management Configuration Arad Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane, or to L3 interfaces for Policy-Based Routing (PBR). Policy maps contain classes, which are composed of class maps and traffic resolution commands.

Traffic Management Conceptual Overview describes traffic policies.

10.2.2.1 Configuring Control Plane Traffic Policies Arad Platform Switches

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by associating traffic resolution commands with static classes that comprise the control plane policy map.

Static Class Maps

Control plane traffic policies utilize static class maps, which are provided by the switch, are not editable, and cannot be deleted.

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in class (policy-map (control-plane) Arad). Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Policy maps are modified in policy-map configuration mode. The `policy-map type copp` command enters policy-map configuration mode.

Example

This command enters policy-map configuration mode for editing copp-system-policy.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)#
```

The `class (policy-map (control-plane) Arad)` command enters policy-map-class configuration mode, where traffic resolution commands are modified for the configuration mode class.

Example

This command enters policy-map-class configuration mode for the copp-system-lacp static class.

```
switch(config-pmap-copp-system-policy)# class copp-system-lacp
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- bandwidth (policy-map-class (control-plane) Arad) specifies the minimum bandwidth.
- shape (policy-map-class (control-plane) Arad) specifies the maximum bandwidth.

Example

These commands configure a bandwidth range of ***2000*** to ***4000*** kilobits per seconds (kbps) for traffic filtered by the copp-system-lacp class map:

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)# bandwidth
 kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lacp)# shape kbps
 4000
```

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Example

These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)# exit
switch(config-pmap-copp-system-policy)# show pending
policy-map type copp copp-system-policy
  class copp-system-bpdu

  class copp-system-lldp

  class copp-system-lacp
    shape kbps 4000
    bandwidth kbps 2000

  class copp-system-l3ttl1

  class copp-system-l3slowpath


switch(config-pmap-copp-system-policy)# exit
switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

Displaying Policy Maps

The show policy-map interface type qos command displays the configured values of the policy maps classes and the number of packets filtered and dropped as a result of the class maps.

Example

These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config)# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy
  Hardware programming status: InProgress

  Class-map: copp-system-mlag (match-any)
       shape : 10000001 kbps
       bandwidth : 10000001 kbps
      Out Packets : 0
      Drop Packets : 0
```

```
  Class-map: copp-system-bpdu (match-any)
       shape : 2604 kbps
       bandwidth : 1302 kbps
      Out Packets : 0
      Drop Packets : 0

  Class-map: copp-system-lacp (match-any)
       shape : 4230 kbps
       bandwidth : 2115 kbps
      Out Packets : 0
      Drop Packets : 0

switch(config)#

switch(config-pmap-c-copp-system-policy-copp-system-lacp)# exit
```

10.2.2.2 Configuring QoS Traffic Policies Arad Platform Switches

QoS traffic policies are implemented by creating class maps and policy maps, then applying the policy maps to Ethernet and port channel interfaces.

Creating Class Maps

QoS traffic policies utilize dynamic class maps that are created and modified in class-map configuration mode. The `class-map type qos` command enters class-map configuration mode.

Example

This command enters class-map configuration mode to create QoS class map named ***Q-CMap_1***.

```
switch(config)# class-map type qos match-any Q-CMap_1
switch(config-cmap-Q-CMap_1)#
```

A class map contains one IPv4 access control list (ACL). The **`match ip access-group`** command assigns an ACL to the class map. Subsequent **`match`** commands replace the existing **`match`** command. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded.

Example

This command adds the IPv4 ACL named ***ACL_1*** to the class map.

```
switch(config-cmap-Q-CMap_1)# match ip access-group ACL_1
switch(config-cmap-Q-CMap_1)#
```

Class-map configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **`show active`** command displays the saved version of class map. The **`show pending`** command displays the unsaved class map.

Example

The **show active** command indicates that the configuration mode class map is not stored in ***running-config***. The **show pending** command displays the class map to be stored upon exiting class-map configuration mode.

```
switch(config-cmap-Q-CMap_1)# show active
switch(config-cmap-Q-CMap_1)# show pending
class-map type qos match-any Q-CMap_1
   match ip access-group ACL_1

switch(config-cmap-Q-CMap_1)#
```

The **exit** command returns the switch to global configuration mode and saves pending class map changes. The **abort** command returns the switch to global configuration mode and discards pending changes.

Example

This command exits class-map configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-CP-CMAP_1)# exit
switch(config)# show class-map type control-plane CP-CMAP_1
  Class-map: CP-CMAP_1 (match-any)
    Match: ip access-group name ACLv4_1
switch(config)#
```

Creating Policy Maps

Policy maps are created and modified in policy-map configuration mode. The `policy-map type quality-of-service` command enters policy-map configuration mode.

Example

This command places the switch in policy-map configuration mode and creates a QoS policy map named ***Q-PMAP_1***.

```
switch(config)# policy-map type quality-of-service Q-PMAP_1
switch(config-pmap-Q-PMAP_1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; traffic resolution commands are added or edited in ***policy-map-class*** configuration mode. The **below** command adds a class to the configuration mode policy map and places the switch in ***policy-map-class*** configuration mode, where traffic resolution commands are added to the class.

Example

This command adds the ***Q-CMap_1*** class to the ***Q-PMAP_1*** policy map and places the switch in ***policy-map-class*** configuration mode.

```
switch(config-pmap-Q-PMAP_1)# class Q-CMap_1
```

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

The **set cos** commands configure traffic resolution methods for data that passes the class map:

- **set cos** sets the Layer 2 CoS field.
- **set dscp** sets the DSCP value in the ToS byte.
- **set traffic class** specifies a traffic class queue.

Example

These commands configure the policy map to set the ***CoS field 7*** on packets filtered by the class map, then assigns those packets to ***traffic class 4***.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# set cos 7
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# set traffic-class 4
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

Policy-map and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** and ***show pending*** commands display the saved and modified policy map versions, respectively.

Example

These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# exit
switch(config-pmap-Q-PMAP_1)# show pending
policy-map type quality-of-service Q-PMAP_1
  class Q-CMap_1
    set cos 7
    set traffic-class 4

  class class-default

switch(config-pmap-Q-PMAP_1)# exit
switch(config)#
```

The last class in all QoS policy maps is ***class-default***. The ***class-default*** class map matches all traffic except IPv4 or IPv6 traffic and provides no traffic resolution commands. The ***class-default*** class map is not editable; traffic resolution commands can be added to the ***class-default*** class.

To modify traffic resolution commands for the ***class-default*** class, enter ***policy-map-class*** configuration mode for the class, then enter the desired **set** commands.

Example

These commands enter ***policy-map-class*** configuration mode for ***class-default***, configures the stream to enter ***traffic class 2***, and saves the altered policy map to ***running-config***.

```
switch(config)# policy-map type quality-of-service Q-PMap_1
switch(config-pmap-Q-PMap_1)# class class-default
switch(config-pmap-c-Q-PMap_1-class-default)# set traffic-class 2
```

```
switch(config-pmap-c-Q-PMap_1-class-default)# exit
switch(config-pmap-Q-PMap_1)# exit
switch(config)# show policy-map type qos Q-PMap_1
Service-policy Q-PMap_1

  Class-map: Q-CMap_1 (match-any)
    Match: ipv6 access-group name ACLv6_1
       set cos 7
       set traffic-class 4

  Class-map: class-default (match-any)
       set traffic-class 2

switch(config)#
```

Applying Policy Maps to an Interface

The service-policy type qos (Interface mode) command applies a specified policy map to the configuration mode interface.

These commands apply ***PMAP-1*** policy map to ***interfaceEthernet 8***.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# show active
switch(config-if-Et8)# service-policy input PMAP-1
switch(config-if-Et8)# show active
interface Ethernet8
   service-policy type qos input PMAP-1
switch(config-if-Et8)#
```

10.2.2.3 Configuring PBR Policies Arad Platform Switches

Policy-Based Routing (PBR) is implemented by creating class maps and policy maps, then applying the policy maps to Ethernet interfaces, port channel interfaces or switch virtual interfaces (SVIs).

Creating PBR Class Maps

PBR policies utilize class maps that are created and modified in the ***class-map*** configuration mode. The class-map type pbr command enters the ***class-map*** configuration mode.

Example

This command enters the ***class-map*** configuration mode to create a PBR class map named CMAP1.

```
switch(config)# class-map type pbr match-any CMAP1
switch(config-cmap-PBR-CMAP1)#
```

A class map contains one or more access control lists (ACLs). The match (policy-map (pbr)) command assigns an ACL to the class map. Subsequent **match** commands add additional ACLs to the class map. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded; if a class map includes ACLs with deny rules, the configuration reverts to its previous state.

Example

This command adds the ACL named ***ACL1*** to the class map.

```
switch(config-cmap-PBR-CMAP1)# match ip access-group ACL1
switch(config-cmap-PBR-CMAP1)#
```

The ***class-map*** configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **show active** command displays the saved version of class map.

The **show active** command indicates that the configuration mode class map is not stored in ***running-config***.

```
switch(config-cmap-PBR-CMAP1)# show active
switch(config-cmap-PBR-CMAP1)#
```

The **exit** command returns the switch to the ***global*** configuration mode and saves pending class map changes. The **abort** command returns the switch to the ***global*** configuration mode and discards pending changes.

Example

This command exits class-map configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-PBR-CMAP1)# exit
switch(config)# show class-map type pbr CMAP1
class-map type pbr match-any CMAP1
   10 match ip access-group ACL1
switch(config)#
```

Creating PBR Policy Maps

Policy maps are created and modified in policy-map configuration mode. The policy-map type pbr command enters the ***policy-map*** configuration mode.

Example

This command enters the ***policy-map*** configuration mode for creating a PBR policy map named ***PMAP1***.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; next-hop commands are added or edited in the ***policy-map-class*** configuration mode. The class (policy-map (pbr)) command adds a class to the configuration mode policy map and places the switch in the ***policy-map-class*** configuration mode, where next-hop commands are added to the class.

Example

- This command adds the *CMAP1* class to the policy map and places the switch into the ***policy-map-class*** configuration mode.

```
switch(config-pmap-PMAP1)# class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

  The set nexthop (policy-map-class pbr) command configures the next hop for data that passes the class map.
- This command configures the policy map to set the next hop to ***10.12.0.5*** on packets filtered by the class map.

```
switch(config-pmap-c-PMAP1-CMAP1)# set nexthop 10.12.0.5
switch(config-pmap-c-PMAP1-CMAP1)#
```

  The set nexthop-group (policy-map-class(pbr) Arad) command configures a nexthop group as the next hop for data that passes the class map.
- These commands configure the policy map ***PMAP1*** to set the next hop to a nexthop group named ***GROUP1*** for traffic defined by class map ***CMAP1***.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)# class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)# set nexthop-group GROUP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

  The ***policy-map*** and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the currently saved map version.
- These commands exits the ***policy-map-class*** configuration mode, then exits the ***policy-map*** configuration mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-PMAP1-CMAP1)# exit
switch(config-pmap-PMAP1)# exit
switch(config)#
```

Applying a PBR Policy Map to an Interface

The service-policy type pbr (Interface mode) command applies the specified PBR policy map to the configuration mode interface. Only one PBR service policy is supported per interface.

These commands apply the PMAP1 PBR policy map to ***interface ethernet 8***.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# service-policy type pbr input PMAP1
switch(config-if-Et8)#
```

Hardware Decapsulation

When hardware decapsulation takes place, PBR policy maps on Arad platform switches match on outer packet headers (i.e., they match based on the attributes of the packet before it is decapsulated).

10.2.3 Traffic Management Configuration FM6000 Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane or an interface. Policy maps contain classes, which are composed of class maps and traffic resolution commands. Traffic Management Conceptual Overview describes traffic policies.

FM6000 platform switches support the following traffic policies:

- Control plane policies manage control plane traffic.
- QoS traffic policies manage traffic on Ethernet and port channel interfaces.

These sections describe the construction and application of policy maps on FM6000 platform switches:

- Configuring Control Plane Traffic Policies FM6000 Platform Switches
- Configuring QoS Traffic Policies FM6000 Platform Switches
- Configuring PBR Policies FM6000 Platform Switches

10.2.3.1 Configuring Control Plane Traffic Policies FM6000 Platform Switches

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by associating traffic resolution commands with static classes that comprise the control plane policy map.

Static Class Maps

Control plane traffic policies utilize static class maps, which are provided by the switch, are not editable, and cannot be deleted.

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in copp-system-policy default classes: FM6000 Platform Switches. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 50: Copp-system-policy Default Classes: FM6000 Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-arp	10000	1000
copp-system-default	8000	1000
copp-system-ipmcrsvd	10000	1000
copp-system-ipmcmiss	10000	1000
copp-system-igmp	10000	1000
copp-system-l2rsvd	10000	10000
copp-system-l3slowpath	10000	1000
copp-system-pim-ptp	10000	1000
copp-system-ospf-isis	10000	1000
copp-system-selfip	5000	5000
copp-system-selfip-tc6to7	5000	5000
copp-system-sflow	25000	1000

Policy maps are modified in the ***policy-map*** configuration mode. The `policy-map type copp` command enters the ***policy-map*** configuration mode.

Example

This command enters the ***policy-map*** configuration mode for editing ***copp-system-policy***.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)#
```

The `class (policy-map (control-plane) FM6000)` command enters the ***policy-map-class*** configuration mode, where traffic resolution commands are modified for the configuration mode class.

Example

This command enters the ***policy-map-class*** configuration mode for the copp-system-arp static class.

```
switch(config-pmap-copp-system-policy)# class copp-system-arp
switch(config-pmap-c-copp-system-policy-copp-system-arp)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- bandwidth (policy-map-class (control-plane) FM6000)
- shape (policy-map-class (control-plane) FM6000)

Example

These commands configure a bandwidth range of ***2000*** to ***4000*** packets per seconds (pps) for traffic filtered by the copp-system-arp class map:

```
switch(config-pmap-c-copp-system-policy-copp-system-arp)# bandwidth
 pps 2000
switch(config-pmap-c-copp-system-policy-copp-system-arp)# shape pps
 4000
switch(config-pmap-c-copp-system-policy-copp-system-arp)#
```

The policy-map and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **`exit`** command or discarded with the **`abort`** command. The **`show active`** command displays the saved version of policy map. The **`show pending`** command displays the modified policy map.

Example

These commands exit the ***policy-map-class*** configuration mode, display the pending policy-map, then exits the ***policy-map*** configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)# exit
switch(config-pmap-copp-system-policy)# show pending
policy-map type copp copp-system-policy
  class CP-CMAP_1
    shape pps 4000
    bandwidth pps 2000
```

```
  class copp-system-bpdu

  class copp-system-lldp

  class copp-system-lacp

  class copp-system-arp


  class copp-system-arpresolver

  class copp-system-default

switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

10.2.3.2 Configuring QoS Traffic Policies FM6000 Platform Switches

QoS traffic policies are implemented by creating class maps and policy maps, then applying the policy maps to Ethernet and port channel interfaces.

Creating Class Maps

QoS traffic policies utilize dynamic class maps that are created and modified in the ***class-map*** configuration mode. The `class-map type qos` command enters the ***class-map*** configuration mode.

Example

This command enters the ***class-map*** configuration mode to create QoS class map named Q-CMap_1.

```
switch(config)# class-map type qos match-any Q-CMap_1
switch(config-cmap-Q-CMap_1)#
```

A class map contains one IPv4 access control list (ACL). The `match (class-map (qos) FM6000)` command assigns an ACL to the class map. Subsequent **match** commands replace the existing **match** command. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded.

Example

This command adds the IPv4 ACL named ***ACL_1*** to the class map.

```
switch(config-cmap-Q-CMap_1)# match ip access-group ACL_1
switch(config-cmap-Q-CMap_1)#
```

The ***class-map*** configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **`show active`** command displays the saved version of class map. The **`show pending`** command displays the unsaved class map.

Example

The **`show active`** command indicates that the configuration mode class map is not stored in ***running-config***. The **`show pending`** command displays the class map to be stored upon exiting the ***class-map*** configuration mode.

```
switch(config-cmap-Q-CMap_1)# show active
switch(config-cmap-Q-CMap_1)# show pending
class-map type qos match-any Q-CMap_1
   match ip access-group ACL_1

switch(config-cmap-Q-CMap_1)#
```

The **`exit`** command returns the switch to the ***global*** configuration mode and saves pending class map changes. The **`abort`** command returns the switch to the ***global*** configuration mode and discards pending changes.

Example

This command exits the ***class-map*** configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-CP-CMAP_1)# exit
switch(config)# show class-map type control-plane CP-CMAP_1
  Class-map: CP-CMAP_1 (match-any)
    Match: ip access-group name ACLv4_1
switch(config)#
```

Creating Policy Maps

Policy maps are created and modified in the ***policy-map*** configuration mode. The `policy-map type quality-of-service` command enters the ***policy-map*** configuration mode.

Example

This command places the switch in the ***policy-map*** configuration mode and creates a QoS policy map named ***Q-PMAP_1***.

```
switch(config)# policy-map type quality-of-service Q-PMAP_1
switch(config-pmap-Q-PMAP_1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; traffic resolution commands are added or edited in the ***policy-map-class*** configuration mode. The `class (policy-map (qos) FM6000)` command adds a class to the configuration mode policy map and places the switch in the *policy-map-class* configuration mode, where traffic resolution commands are added to the class.

Example

This command adds the ***Q-CMap_1*** class to the ***Q-PMAP_1*** policy map and places the switch in the ***policy-map-class*** configuration mode.

```
switch(config-pmap-Q-PMAP_1)# class Q-CMap_1
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

`set (policy-map-class (qos) FM6000)` commands configure traffic resolution methods for data that passes the class map:

- **`set cos`** sets the Layer 2 CoS field.
- **`set dscp`** sets the DSCP value in the ToS byte.
- **`set traffic class`** specifies a traffic class queue.

Example

These commands configure the policy map to set the ***CoS field 7*** on packets filtered by the class map, then assigns those packets to ***traffic class 4***.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# set cos 7
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# set traffic-class 4
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

The ***policy-map*** and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **`exit`** command or discarded with the **`abort`** command. The **`show active`** and **`show pending`** commands display the saved and modified policy map versions, respectively.

Example

These commands exit the ***policy-map-class*** configuration mode, display the pending policy-map, then exits the ***policy-map*** configuration mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# exit
switch(config-pmap-Q-PMAP_1)# show pending
policy-map type quality-of-service Q-PMAP_1
  class Q-CMap_1
    set cos 7
    set traffic-class 4

  class class-default

switch(config-pmap-Q-PMAP_1)# exit
switch(config)#
```

The last class in all QoS policy maps is ***class-default***. The ***class-default*** class map matches all traffic except IPv4 or IPv6 traffic and provides no traffic resolution commands. The ***class-default*** class map is not editable; traffic resolution commands can be added to the ***class-default*** class.

To modify traffic resolution commands for the ***class-default*** class, enter the ***policy-map-class*** configuration mode for the class, then enter the desired **`set`** commands.

Example

These commands enter the ***policy-map-class*** configuration mode for ***class-default***, configures the stream to enter ***traffic class 2***, and saves the altered policy map to ***running-config***.

```
switch(config)# policy-map type quality-of-service Q-PMap_1
switch(config-pmap-Q-PMap_1) #class class-default
switch(config-pmap-c-Q-PMap_1-class-default)# set traffic-class 2
switch(config-pmap-c-Q-PMap_1-class-default)# exit
switch(config-pmap-Q-PMap_1)# exit
switch(config)# show policy-map type qos Q-PMap_1
Service-policy Q-PMap_1

  Class-map: Q-CMap_1 (match-any)
    Match: ipv6 access-group name ACLv6_1
       set cos 7
       set traffic-class 4

  Class-map: class-default (match-any)
       set traffic-class 2

switch(config)#
```

Applying Policy Maps to an Interface

The service-policy type qos (Interface mode) command applies a specified policy map to the configuration mode interface.

These commands apply ***PMAP-1*** policy map to ***interface ethernet 8***.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# show active
switch(config-if-Et8)# service-policy input PMAP-1
switch(config-if-Et8)# show active
interface Ethernet8
   service-policy type qos input PMAP-1
switch(config-if-Et8)#
```

10.2.3.3 Configuring PBR Policies FM6000 Platform Switches

Policy-Based Routing (PBR) is implemented by creating class maps and policy maps, then applying the policy maps to Ethernet interfaces, port channel interfaces or Switch Virtual Interfaces (SVIs).

Creating PBR Class Maps

PBR policies utilize class maps that are created and modified in the ***class-map*** configuration mode. The `class-map type pbr` command enters the ***class-map*** configuration mode.

Example

This command enters the ***class-map*** configuration mode to create a PBR class map named ***CMAP1***.

```
switch(config)# class-map type pbr match-any CMAP1
switch(config-cmap-PBR-CMAP1)#
```

A class map contains one or more IPv4 access control lists (ACLs). The `match (policy-map (pbr))` command assigns an ACL to the class map. Subsequent **`match`** commands add additional ACLs to the class map. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded; if a class map includes ACLs with deny rules, the configuration reverts to its previous state.

On FM6000 platform switches, counters are not supported, so a `counters per-entry (ACL configuration modes)` command in an ACL is ignored.

Example

This command adds the IPv4 ACL named ***ACL1*** to the class map.

```
switch(config-cmap-PBR-CMAP1)# match ip access-group ACL1
switch(config-cmap-PBR-CMAP1)#
```

The ***class-map*** configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **`show active`** command displays the saved version of class map.

The **`show active`** command indicates that the configuration mode class map is not stored in ***running-config***.

```
switch(config-cmap-PBR-CMAP1)# show active
switch(config-cmap-PBR-CMAP1)#
```

The **`exit`** command returns the switch to ***global*** configuration mode and saves pending class map changes. The **`abort`** command returns the switch to ***global*** configuration mode and discards pending changes.

Example

This command exits the ***class-map*** configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-PBR-CMAP1)# exit
switch(config)# show class-map type pbr CMAP1
class-map type pbr match-any CMAP1
   10 match ip access-group ACL1
switch(config)#
```

Creating PBR Policy Maps

Policy maps are created and modified in the ***policy-map*** configuration mode. The `policy-map type pbr` command enters the ***policy-map*** configuration mode.

Example

This command enters the ***policy-map*** configuration mode for creating a PBR policy map named ***PMAP1***.

```
switch(config)# policy-map type pbr PMAP1
```

```
switch(config-pmap-PMAP1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; next-hop commands are added or edited in the ***policy-map-class*** configuration mode. The `class (policy-map (pbr))` command adds a class to the configuration mode policy map and places the switch in the ***policy-map-class*** configuration mode, where next-hop commands are added to the class.

Examples

- This command adds the ***CMAP1*** class to the policy map and places the switch in the ***policy-map-class*** configuration mode.

  ```
  switch(config-pmap-PMAP1)# class CMAP1
  switch(config-pmap-c-PMAP1-CMAP1)#
  ```

  The **`set nexthop (policy-map-class pbr)`** command configures the next hop for data that passes the class map.
- This command configures the policy map to set the next hop to ***10.12.0.5*** on packets filtered by the class map.

  ```
  switch(config-pmap-c-PMAP1-CMAP1)# set nexthop 10.12.0.5
  switch(config-pmap-c-PMAP1-CMAP1)#
  ```

  The ***policy-map*** and ***policy-map-clas***s configuration modes are group-change modes. Changes are saved with the **`exit`** command or discarded with the **`abort`** command. The **`show active`** command displays the currently saved map version.

Example

These commands exits the ***policy-map-class*** configuration mode, then exits the ***policy-map configuration*** mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-PMAP1-CMAP1)# exit
switch(config-pmap-PMAP1)# exit
switch(config)#
```

Applying a PBR Policy Map to an Interface

The `service-policy type pbr (Interface mode)` command applies the specified PBR policy map to the configuration mode interface. Only one PBR service policy is supported per interface.

These commands apply the ***PMAP1*** PBR policy map to ***interface ethernet 8***.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# service-policy type pbr input PMAP1
switch(config-if-Et8)#
```

Hardware Decapsulation

When hardware decapsulation takes place, PBR policy maps on FM6000 platform switches match on outer packet headers (i.e., they match based on the attributes of the packet before it is decapsulated).

10.2.4 Traffic Management Configuration Petra Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane. Policy maps contain classes, which are composed of class maps and traffic resolution commands. QoS traffic policies are not supported on 7500 Series switches.

Traffic Management Conceptual Overview describes traffic policies.

10.2.4.1 Configuring Control Plane Traffic Policies Petra Platform Switches

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by associating traffic resolution commands with static classes that comprise the control plane policy map.

Static Class Maps

Control plane traffic policies utilize static class maps, which are provided by the switch, are not editable, and cannot be deleted.

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in copp-system-policy default classes: Petra Platform Switches. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 51: copp-system-policy default classes: Petra Platform Switches

Class Name	shape (kbps)	bandwidth (kbps)
copp-system-bpdu	2500	1250
copp-system-default	2500	250
copp-system-igmp	2500	250
copp-system-ipbroadcast	2500	250
copp-system-ipmc	2500	250
copp-system-ipmcmiss	2500	250
copp-system-ipmcrsvd	2500	250
copp-system-ipunicast	NO LIMIT	250
copp-system-l3destmiss	2500	250
copp-system-l3slowpath	2500	250
copp-system-l3ttl0	2500	250
copp-system-l3ttl1	2500	250
copp-system-lacp	2500	1250
copp-system-lldp	2500	250
copp-system-unicast-arp	2500	250

Policy maps are modified in the ***policy-map*** configuration mode. The `policy-map type copp` command enters the ***policy-map*** configuration mode.

Example

This command enters the ***policy-map*** configuration mode for editing **copp-system-policy**.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)#
```

The `class (policy-map (control-plane) Petra)` command enters the ***policy-map-class*** configuration mode, where traffic resolution commands are modified for the configuration mode class.

Example

- This command enters the ***policy-map-class*** configuration mode for the **copp-system-lldp** static class.

  ```
  switch(config-pmap-copp-system-policy)# class copp-system-lldp
  switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
  ```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- `bandwidth (policy-map-class (control-plane) Petra)` specifies the minimum bandwidth.
- `shape (policy-map-class (control-plane) Petra)` specifies the maximum bandwidth.

Example

These commands configure a bandwidth range of ***2000*** to ***4000*** kilobits per seconds (kbps) for traffic filtered by the **copp-system-arp** class map:

```
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# bandwidth
 kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# shape kbps
 4000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

The ***policy-map*** and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the configured policy map.

Petra platform switches do not support all discrete rate values. When a **bandwidth** or **shape** command specifies a value that is not supported, the switch converts the rate to the next highest discrete value that it supports. The show policy-map interface type qos command displays the converted rate and not the user configured rate.

Example

These commands exits the ***policy-map-class*** configuration mode, display the pending policy-map, then exits the ***policy-map*** configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)# exit
switch(config-pmap-copp-system-policy)# show pending
policy-map type copp copp-system-policy
  class copp-system-bpdu

  class copp-system-lldp
    shape kbps 4000
    bandwidth kbps 2000

  class copp-system-lacp

switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Displaying Policy Maps

The show policy-map interface type qos command displays the traffic resolution rates of the policy maps classes and the number of packets filtered and dropped as a result of the class maps. The shape and bandwidth rates may differ from configured values, because the switch does not support all discrete rate values.

Example

These commands exits the ***policy-map-class*** configuration mode, display the pending policy-map, then exits the *policy-map* configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config)# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy
  Hardware programming status: InProgress

  Class-map: copp-system-mlag (match-any)
       shape : 10000001 kbps
       bandwidth : 10000001 kbps
      Out Packets : 0
      Drop Packets : 0

  Class-map: copp-system-lacp (match-any)
       shape : 2604 kbps
       bandwidth : 1302 kbps
      Out Packets : 0
      Drop Packets : 0

switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

10.2.4.2 Configuring QoS Traffic Policies Petra Platform Switches

QoS traffic policies are not supported on Petra platform switches.

10.2.4.3 Configuring PBR Policies Petra Platform Switches

PBR policies are not supported on Petra platform switches.

10.2.5 Traffic Management Configuration Trident Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane or an interface. Policy maps contain classes, which are composed of class maps and traffic resolution commands. Traffic Management Conceptual Overview describes traffic policies.

Trident platform switches support the following traffic policies:

- Control plane policies manage control plane traffic.
- QoS traffic policies manage traffic on Ethernet and port channel interfaces.

These sections describe the construction and application of policy maps:

- Configuring Control Plane Traffic Policies Trident Platform Switches
- Configuring QoS Traffic Policies Trident Platform Switches
- Configuring PBR Policies Trident Platform Switches

10.2.5.1 Configuring Control Plane Traffic Policies Trident Platform Switches

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by creating class maps and editing the policy map to include the new class maps.

Creating Class Maps

Control plane traffic policies utilize static and dynamic class maps. Static class maps are provided by the switch, are not editable, and cannot be deleted. Dynamic class maps are created and modified in the ***class-map*** configuration mode. The `class-map type copp` command enters the ***class-map*** configuration mode.

Example

This command enters the ***class-map*** configuration mode for creating or editing a control plane dynamic class map named ***CP-CMAP_1***.

```
switch(config)# class-map type copp match-any CP-CMAP_1
switch(config-cmap-CP-CMAP_1)#
```

Class maps contain one IPv4 or IPv6 access control list (ACL). The match (class-map (control-plane) Trident) command assigns an ACL to the class map. Subsequent **match** commands replace the existing **match** command. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded.

Example

This command assigns the IPv4 ACL named ***ACLv4_1*** to the class map.

```
switch(config-cmap-CP-CMAP_1)# match ip access-group ACLv4_1
switch(config-cmap-CP-CMAP_1)#
```

The ***class-map*** configuration mode is a group-change mode. Changes are saved by exiting the mode. The **`show active`** command displays the saved version of class map. The **`show pending`** command displays the unsaved class map.

Example

The **`show active`** command indicates that the configuration mode class map is not stored in ***running-config***. The **`show pending`** command displays the class map to be stored upon exiting the ***class-map*** configuration mode.

```
switch(config-cmap-CP-CMAP_1)# show active
switch(config-cmap-CP-CMAP_1)# show pending
class-map type copp match-any CP-CMAP_1
   match ip access-group ACLv4_1

switch(config-cmap-CP-CMAP_1)#
```

The **`exit`** command returns the switch to the ***global*** configuration mode and saves pending class map changes. The **`abort`** command returns the switch to the ***global*** configuration mode and discards pending class map changes.

Example

This command exits the ***class-map*** configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-CP-CMAP_1)# exit
switch(config)# show class-map type control-plane CP-CMAP_1
  Class-map: CP-CMAP_1 (match-any)
    Match: ip access-group name ACLv4_1
switch(config)#
```

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in copp-system-policy default classes: Trident Platform Switches. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 52: copp-system-policy default classes: Trident Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-bpdu	5000	5000
copp-system-lacp	5000	5000
copp-system-selfip-tc6to7	5000	5000
copp-system-selfip	5000	5000
copp-system-tc6to7	10000	1000
copp-system-lldp	10000	1000
copp-system-ipmcrsvd	10000	1000
copp-system-igmp	10000	1000
copp-system-ipmcmiss	10000	1000
copp-system-glean	10000	1000
copp-system-tc3to5	10000	1000
copp-system-arp	10000	1000
copp-system-arpresolver	10000	1000
copp-system-l3destmiss	10000	1000
copp-system-l3slowpath	10000	1000
copp-system-l3ttl1	10000	1000
copp-system-default	8000	1000
copp-system-acllog	10000	1000
copp-system-sflow	25000	0

Policy maps are modified in the ***policy-map*** configuration mode. The `policy-map type copp` command enters the ***policy-map*** configuration mode.

Example

This command enters the ***policy-map*** configuration mode for editing **copp-system-policy**.

```
switch(config)#policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)#
```

Dynamic classes are inserted in front of the static classes. Classes automatically contain their eponymous class map; traffic resolution commands are created or edited in the ***policy-map-class*** configuration mode. The class (policy-map (control-plane) Trident) command adds a class to the policy map and places the switch in the ***policy-map-class*** configuration mode, where traffic resolution commands are added to the class.

Example

This command adds the ***CP-CMAP_1*** class to the copp-system-policy policy map and places the switch in the ***policy-map-class*** configuration mode.

```
switch(config-pmap-copp-system-policy)#class CP-CMAP_1
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- bandwidth (policy-map-class (control-plane) Trident) specifies the minimum bandwidth.
- shape (policy-map-class (control-plane) Trident) specifies the maximum bandwidth.

Example

These commands configure a bandwidth range of ***2000*** to ***4000*** packets per seconds (pps) for traffic filtered by the ***CP-CMAP_1*** class map:

```
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)# bandwidth pps 2000
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)# shape pps 4000
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)#
```

Example

The ***policy-map*** and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Example

These commands exits the ***policy-map-class*** configuration mode, display the pending policy-map, then exits the ***policy-map*** configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)# exit
switch(config-pmap-copp-system-policy)# show pending
policy-map type copp copp-system-policy
  class CP-CMAP_1
    shape pps 4000
    bandwidth pps 2000

  class copp-system-bpdu

  class copp-system-lldp

  class copp-system-lacp

  class copp-system-arp

  class copp-system-arpresolver

  class copp-system-default

switch(config-pmap-copp-system-policy)# exit
```

```
switch(config)#
```

Example

To modify traffic resolution commands for a static class, enter the ***policy-map-class*** configuration mode for the class, then enter the desired **bandwidth** and **shape** commands.

Example

These commands enters the ***policy-map-class*** configuration mode for **copp-system-bpdu** class, change the bandwidth range for the class, then save the altered policy map to ***running-config***.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-bpdu
switch(config-pmap-c-copp-system-policy-copp-system-bpdu)# shape pps
 200
switch(config-pmap-c-copp-system-policy-copp-system-bpdu)# bandwidth
 pps 100
switch(config-pmap-c-copp-system-policy-copp-system-bpdu)# exit
switch(config-pmap-copp-system-policy)# show pending
policy-map type copp copp-system-policy
  class CP-CMAP_1
    shape pps 4000
    bandwidth pps 2000

  class copp-system-bpdu
    shape pps 200
    bandwidth pps 100

  class copp-system-lldp

switch(config-pmap-copp-system-policy)# exit
switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

10.2.5.2 Configuring QoS Traffic Policies Trident Platform Switches

QoS traffic policies are implemented by creating class maps and policy maps, then applying the policy maps to Ethernet and port channel interfaces.

Creating Class Maps

QoS traffic policies utilize dynamic class maps that are created and modified in the ***class-map*** configuration mode. The `class-map type qos` command enters the ***class-map*** configuration mode.

Example

This command enters the ***class-map*** configuration mode to create QoS class map named ***Q-CMap_1***.

```
switch(config)# class-map type qos match-any Q-CMap_1
switch(config-cmap-Q-CMap_1)#
```

A class map contains one IPv4 or IPv6 Access Control List (ACL). The `match (class-map (qos) Trident)` command assigns an ACL to the class map. Subsequent **`match`** commands replace the existing **`match`** command. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded.

Example

This command adds the IPv6 ACL named **`ACLv6_1`** to the class map.

```
switch(config-cmap-Q-CMap_1)# match ipv6 access-group ACLv6_1
switch(config-cmap-Q-CMap_1)#
```

The ***class-map*** configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **`show active`** command displays the saved version of class map. The **`show pending`** command displays the unsaved class map.

Example

The **`show active`** command indicates that the configuration mode class map is not stored in ***running-config***. The **`show pending`** command displays the class map to be stored upon exiting the ***class-map*** configuration mode.

```
switch(config-cmap-Q-CMap_1)# show active
switch(config-cmap-Q-CMap_1)# show pending
class-map type qos match-any Q-CMap_1
   match ipv6 access-group ACLv6_1

switch(config-cmap-Q-CMap_1)#
```

The **`exit`** command returns the switch to ***global*** configuration mode and saves pending class map changes. The **`abort`** command returns the switch to ***global*** configuration mode and discards pending class map changes.

Example

This command exits the ***class-map*** configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-CP-CMAP_1)# exit
switch(config)# show class-map type control-plane CP-CMAP_1
  Class-map: CP-CMAP_1 (match-any)
    Match: ip access-group name ACLv4_1
switch(config)#
```

Creating Policy Maps

Policy maps are created and modified in the ***policy-map*** configuration mode. The `policy-map type quality-of-service` command enters the ***policy-map*** configuration mode.

Example

This command enters the ***policy-map*** configuration mode for creating a QoS policy map named ***Q-PMAP_1***.

```
switch(config)# policy-map type quality-of-service Q-PMAP_1
switch(config-pmap-Q-PMAP_1)#
```

Policy maps are edited by adding or removing classes. A class automatically contains its eponymous class map; traffic resolution commands are added or edited in the ***policy-map-class*** configuration mode. The `class (policy-map (qos) Trident)` command adds a class to the configuration mode policy map and places the switch in the ***policy-map-class*** configuration mode, where traffic resolution commands are added to the class.

Example

This command adds the ***Q-CMap_1*** class to the ***Q-PMAP_1*** policy map and places the switch in the ***policy-map-class*** configuration mode.

```
switch(config-pmap-Q-PMAP_1)# class Q-CMap_1
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

The **`set (policy-map-class (qos) Trident)`** command configures traffic resolution methods for data that passes the class map:

- **set cos** sets the layer 2 CoS field.
- **set dscp** sets the DSCP value in the ToS byte.
- **set traffic class** specifies a traffic class queue.

Example

These commands configure the policy map to set ***CoS field 7*** on packets filtered by the class map, then assigns those packets to ***traffic class 4***.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# set cos 7
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# set traffic-class 4
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

The ***policy-map*** and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **`abort`** command. The **`show active`** and **`show pending`** commands display the saved and modified policy map versions, respectively.

Example

These commands exit the ***policy-map-class*** configuration mode, display the pending policy-map, then exits the ***policy-map*** configuration mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)# exit
switch(config-pmap-Q-PMAP_1)# show pending
policy-map type quality-of-service Q-PMAP_1
  class Q-CMap_1
    set cos 7
    set traffic-class 4

  class class-default

switch(config-pmap-Q-PMAP_1)# exit
switch(config)#
```

The last class in all QoS policy maps is ***class-default***. The ***class-default*** class map matches all traffic except IPv4 or IPv6 traffic and provides no traffic resolution commands. The ***class-default*** class map is not editable; traffic resolution commands can be added to the ***class-default*** class.

To modify traffic resolution commands for the ***class-default*** class, enter the ***policy-map-class*** configuration mode for the class, then enter the desired **`set`** commands.

Example

These commands enters the ***policy-map-class*** configuration mode for ***class-default***, configures the stream to enter ***traffic class 2***, and saves the altered policy map to ***running-config***.

```
switch(config)# policy-map type quality-of-service Q-PMap_1
switch(config-pmap-Q-PMap_1)# class class-default
switch(config-pmap-c-Q-PMap_1-class-default)# set traffic-class 2
switch(config-pmap-c-Q-PMap_1-class-default)# exit
switch(config-pmap-Q-PMap_1)# exit
switch(config)# show policy-map type qos Q-PMap_1
Service-policy Q-PMap_1

  Class-map: Q-CMap_1 (match-any)
    Match: ipv6 access-group name ACLv6_1
       set cos 7
       set traffic-class 4

  Class-map: class-default (match-any)
       set traffic-class 2

switch(config)#
```

Applying Policy Maps to an Interface

The `service-policy type qos (Interface mode)` command applies a specified policy map to the configuration mode interface.

Example

These commands apply ***PMAP-1*** policy map to ***interface ethernet 8***.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# show active
```

```
switch(config-if-Et8)# service-policy input PMAP-1
switch(config-if-Et8)# show active
interface Ethernet8
   service-policy type qos input PMAP-1
switch(config-if-Et8)#
```

10.2.5.3 Configuring PBR Policies Trident Platform Switches

Policy-Based Routing (PBR) is implemented by creating class maps and policy maps, then applying the policy maps to Ethernet interfaces, port channel interfaces or Switch Virtual Interfaces (SVIs).

Creating PBR Class Maps

PBR policies utilize class maps that are created and modified in the ***class-map*** configuration mode. The `class-map type pbr` command enters the ***class-map*** configuration mode.

Example

This command enters the ***class-map*** configuration mode to create a PBR class map named ***CMAP1***.

```
switch(config)# class-map type pbr match-any CMAP1
switch(config-cmap-PBR-CMAP1)#
```

A class map contains one or more Access Control Lists (ACLs). The `match (policy-map (pbr))` command assigns an ACL to the class map. Subsequent **`match`** commands add additional ACLs to the class map. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded; if a class map includes ACLs with deny rules, the configuration reverts to its previous state.

Examples

- This command adds the ACL named ***ACL1*** to the class map.

```
switch(config-cmap-PBR-CMAP1)# match ip access-group ACL1
switch(config-cmap-PBR-CMAP1)#
```

The ***class-map*** configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **`show active`** command displays the saved version of class map.

- The **`show active`** command indicates that the configuration mode class map is not stored in ***running-config***.

```
switch(config-cmap-PBR-CMAP1)# show active
switch(config-cmap-PBR-CMAP1)#
```

The **`exit`** command returns the switch to ***global*** configuration mode and saves pending class map changes. The **`abort`** command returns the switch to ***global*** configuration mode and discards pending changes.

Example

This command exits the ***class-map*** configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-PBR-CMAP1)# exit
switch(config)# show class-map type pbr CMAP1
class-map type pbr match-any CMAP1
   10 match ip access-group ACL1
switch(config)#
```

Creating PBR Policy Maps

Policy maps are created and modified in the ***policy-map*** configuration mode. The `policy-map type pbr` command enters policy-map configuration mode.

Example

This command enters the ***policy-map*** configuration mode for creating a PBR policy map named ***PMAP1***.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; next-hop commands are added or edited in the ***policy-map-class*** configuration mode. The `class (policy-map (pbr))` command adds a class to the configuration mode policy map and places the switch in the ***policy-map-class*** configuration mode, where next-hop commands are added to the class.

Example

- This command adds the ***CMAP1*** class to the policy map and places the switch in the ***policy-map-class*** configuration mode.

  ```
  switch(config-pmap-PMAP1)# class CMAP1
  switch(config-pmap-c-PMAP1-CMAP1)#
  ```

- The set nexthop (policy-map-class pbr) command configures the next hop for data that passes the class map.This command configures the policy map to set the next hop to ***10.12.0.5*** on packets filtered by the class map.

  ```
  switch(config-pmap-c-PMAP1-CMAP1)# set nexthop 10.12.0.5
  switch(config-pmap-c-PMAP1-CMAP1)#
  ```

- The ***policy-map*** and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **`exit`** command or discarded with the **`abort`** command. The **`show active`** command displays the currently saved map version.These commands exits the ***policy-map-class*** configuration mode, then exits the ***policy-map*** configuration mode to save the altered policy map to ***running-config***.

  ```
  switch(config-pmap-c-PMAP1-CMAP1)# exit
  switch(config-pmap-PMAP1)# exit
  switch(config)#
  ```

Applying a PBR Policy Map to an Interface

The service-policy type pbr (Interface mode) command applies the specified PBR policy map to the configuration mode interface. Only one PBR service policy is supported per interface.

- These commands apply the ***PMAP1*** PBR policy map to ***interface ethernet 8***.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# service-policy type pbr input PMAP1
switch(config-if-Et8)#
```

Hardware Decapsulation

When hardware decapsulation takes place, PBR policy maps on Trident platform switches match on inner packet headers (i.e., they match based on the attributes of the decapsulated packet).

10.2.6 Traffic Management Configuration Trident II Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane or an interface. Policy maps contain classes, which are composed of class maps and traffic resolution commands. Traffic Management Conceptual Overview describes traffic policies.

Trident platform switches support the following traffic policies:

- Control plane policies manage control plane traffic.
- QoS traffic policies manage traffic on Ethernet and port channel interfaces.

10.2.6.1 Configuring Control Plane Traffic Policies Trident II Platform Switches

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by associating traffic resolution commands with static classes that comprise the control plane policy map.

Static Class Maps

Control plane traffic policies utilize static class maps, which are provided by the switch, are not editable, and cannot be deleted.

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in copp-system-policy default classes: Trident II Platform Switches. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 53: copp-system-policy default classes: Trident II Platform Switches

Class Name	**shape (pps)**	bandwidth (pps)
copp-system-acllog	1000	10000
copp-system-arp	1000	10000
copp-system-arpresolver	1000	10000
copp-system-bfd	5000	10000
copp-system-bgp	5000	5000
copp-system-bpdu	5000	5000
copp-system-default	1000	8000
copp-system-glean	1000	10000
copp-system-igmp	1000	10000
copp-system-ipmcmiss	1000	10000
copp-system-ipmcrsvd	1000	10000
copp-system-l3destmiss	1000	10000
copp-system-l3slowpath	1000	10000
copp-system-l3ttl1	1000	10000
copp-system-lacp	5000	5000
copp-system-lldp	1000	10000
copp-system-mlag	5000	5000
copp-system-selfip	5000	5000
copp-system-selfip-tc6to7	5000	5000
copp-system-sflow	0	25024
copp-system-tc3to5	1000	10000
copp-system-tc6to7	1000	10000
copp-system-urm	1000	10000

Policy maps are modified in the ***policy-map*** configuration mode. The policy-map type copp command enters the ***policy-map*** configuration mode.

Example

This command enters the ***policy-map*** configuration mode for editing **copp-system-policy**.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)#
```

Example

The class (policy-map (control-plane) Trident II) command enters the ***policy-map-class*** configuration mode, where traffic resolution commands are modified for the configuration mode class.

Example

This command enters the ***policy-map-class*** configuration mode for the **copp-system-lacp static** class.

```
switch(config-pmap-copp-system-policy)# class copp-system-lacp
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- bandwidth (policy-map-class (control-plane) Trident II) specifies the minimum bandwidth.
- shape (policy-map-class (control-plane) Trident II) specifies the maximum bandwidth.

Example

These commands configure a bandwidth range of ***2000*** to ***4000*** packets per seconds (pps) for traffic filtered by the **copp-system-lacp** class map:

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)# bandwidth
 pps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lacp)# shape pps
 4000
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#
```

Example

The ***policy-map*** and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Example

These commands exits the ***policy-map-class*** configuration mode, display the pending ***policy-map***, then exit ***policy-map*** configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)# exit
switch(config-pmap-copp-system-policy)# show pending
policy-map type copp copp-system-policy
  class copp-system-bpdu

  class copp-system-lldp

  class copp-system-lacp
    shape pps 4000
```

```
    bandwidth pps 2000

  class copp-system-arp

switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

10.2.7 Traffic Management Configuration Commands

Traffic Policy (Control Plane) Configuration Commands

- bandwidth (policy-map-class (control-plane) Arad)
- bandwidth (policy-map-class (control-plane) FM6000)
- bandwidth (policy-map-class (control-plane) Helix)
- bandwidth (policy-map-class (control-plane) Petra)
- bandwidth (policy-map-class (control-plane) Trident)
- bandwidth (policy-map-class (control-plane) Trident II)
- class-map type copp
- class (policy-map (control-plane) Arad)
- class (policy-map (control-plane) FM6000)
- class (policy-map (control-plane) Helix)
- class (policy-map (control-plane) Petra)
- class (policy-map (control-plane) Trident)
- class (policy-map (control-plane) Trident II)
- match (class-map (control-plane) Helix)
- match (class-map (control-plane) Trident)
- match (class-map (control-plane) Trident II)
- policy-map type copp
- shape (policy-map-class (control-plane) Arad)
- shape (policy-map-class (control-plane) FM6000)
- shape (policy-map-class (control-plane) Helix)
- shape (policy-map-class (control-plane) Petra)
- shape (policy-map-class (control-plane) Trident)
- shape (policy-map-class (control-plane) Trident II)

Traffic Policy (PBR) Configuration Commands

- action set-ttl
- class (policy-map (pbr))
- class-map type pbr
- feature pbr
- match (class-map (pbr))
- match (policy-map (pbr))
- platform arad tcam counters feature
- policy-map type pbr
- resequence (class-map (pbr))

- resequence (policy-map (pbr))
- service-policy type pbr (Interface mode)
- set nexthop (policy-map-class pbr)
- set nexthop-group (policy-map-class(pbr) Arad)

CPU Traffic Policy Command

- feature traffic-policy cpu
- feature traffic-policy port

Traffic Policy (QoS) Configuration Commands

- class-map type qos
- class (policy-map (qos) FM6000)
- class (policy-map (qos) Helix)
- class (policy-map (qos) Trident)
- class (policy-map (qos) Trident II)
- match (class-map (qos) FM6000)
- match (class-map (qos) Helix)
- match (class-map (qos) Trident)
- match (class-map (qos) Trident II)
- policy-map type quality-of-service
- policy-map type quality-of-service policer
- service-policy type qos (Interface mode)
- set (policy-map-class (qos) FM6000)
- set (policy-map-class (qos) Helix)
- set (policy-map-class (qos) Trident)
- set (policy-map-class (qos) Trident II)

Traffic Policy Display and Utility Commands

- clear policy-map counters
- show class-map type control-plane
- show class-map type pbr
- show class-map type qos
- show policy-map type copp
- show policy-map type pbr
- show policy-map type qos
- show policy-map type qos counters
- show policy-map copp
- show policy-map interface type qos
- show policy-map interface type qos counters
- show traffic-policy

10.2.7.1 action set-ttl

The TTL action is effective only when it is configured along with a set nexthop or nexthop-group action. The TCAM profile has the set-ttl-3b or set-ttl action in the pbr ip and pbr ipv6 features, such as in the tc-counters system profile.

Command Mode

For IP

TCAM feature PBR IP configuration mode.

For IPv6

TCAM feature PBR IPv6 configuration mode.

Command Syntax

`action set-time` [set-ttl | set-ttl-3b]

`no action set-time` [set-ttl | set-ttl-3b]

`default action set-time` [set-ttl | set-ttl-3b]

Parameters

- **set-ttl** Set time to live.
- **set-ttl-3b** Set 3-bit time to live.

Example

In the following example, for IP, the action sets the time to live for the next hop.

```
(config)# hardware tcam
(config-tcam)# profile pbr-set-ttl copy default
(config-tcam-profile-pbr-set-ttl)# feature pbr ip
(config-tcam-feature-pbr-ip)# action set-ttl
```

In the following example, for IPv6, the action sets the time to live for the next hop group.

```
config)# hardware tcam
(config-tcam)# profile pbr-set-ttl copy default
(config-tcam-profile-pbr-set-ttl)# feature pbr ip
(config-tcam-feature-pbr-ip)# feature pbr ipv6
(config-tcam-feature-pbr-ipv6)# action set-ttl
```

10.2.7.2 bandwidth (policy-map-class (control-plane) Arad)

The **`bandwidth`** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **`no bandwidth`** and **`default bandwidth`** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **`bandwidth`** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration

accessed through `class (policy-map (control-plane) Arad)`

Command Syntax

`bandwidth kbps` ***kilobits***

`no bandwidth`

`default bandwidth`

Parameters

kilobits Minimum data rate in kilobits per second. Value ranges from ***1*** to ***10000000***.

Related Commands

- class (policy-map (control-plane) Arad) places the switch in the ***policy-map-class*** (control plane) configuration mode.
- shape (policy-map-class (control-plane) Arad) specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Arad platform switches define these default bandwidths for control plane static classes:

- copp-system-bgp 250 copp-system-l3lpmoverflow 250
- copp-system-bpdu 1250 copp-system-l3slowpath 250
- copp-system-default 250 copp-system-l3ttl1 250
- copp-system-ipbroadcast 250 copp-system-lacp 1250
- copp-system-ipmc 250 copp-system-linklocal 250
- copp-system-ipmcmiss 250 copp-system-lldp 250
- copp-system-ipunicast 250 copp-system-mlag 250
- copp-system-l2broadcast 250 copp-system-multicastsnoop 250
- copp-system-l2unicast 250 copp-system-OspfIsis 250
- copp-system-l3destmiss 250 copp-system-sflow 250

Example

These commands configure the minimum bandwidth of ***500*** kbps for data traffic specified by the class map **copp-system-lldp** of the default ***control-plane*** policy map.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# bandwidth kbps 500
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# exit
switch(config-pmap-copp-system-policy)# exit
switch(config)# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy
  Hardware programming status: InProgress

  Class-map: copp-system-lldp (match-any)
       shape : 2500 kbps
       bandwidth : 500 kbps
      Out Packets : 0
      Drop Packets : 0

switch(config)#
```

10.2.7.3 bandwidth (policy-map-class (control-plane) FM6000)

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration

accessed through class (policy-map (control-plane) FM6000)

Command Syntax

bandwidth pps ***packets***

no bandwidth

default bandwidth

Parameters

packets Minimum data rate in packets per second. Value ranges from ***1*** to ***100000***.

Related Commands

- class (policy-map (control-plane) FM6000) places the switch in ***policy-map-class*** (control plane) configuration mode.
- shape (policy-map-class (control-plane) FM6000) specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

FM6000 platform switches define these default bandwidths for control plane static classes:

- copp-system-arp 1000 copp-system-l3slowpath 1000
- copp-system-default 1000 copp-system-pim-ptp 1000
- copp-system-ipmcrsvd 1000 copp-system-ospf-isis 1000
- copp-system-ipmcmiss 1000 copp-system-selfip 5000
- copp-system-igmp 1000 copp-system-selfip-tc6to7 5000
- copp-system-l2rsvd 10000 copp-system-sflow 1000

Example

These commands configure the minimum bandwidth of ***1000*** packets per second for data traffic specified by the class map ***PMAP-1*** in the policy map named **copp-system-policy**.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class PMAP-1
switch(config-pmap-c-copp-system-policy-PMAP-1)# bandwidth pps 1000
switch(config-pmap-c-copp-system-policy-PMAP-1)#
```

10.2.7.4 bandwidth (policy-map-class (control-plane) Helix)

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration

accessed through `class (policy-map (control-plane) Helix)`

Command Syntax

bandwidth pps ***packets***

no bandwidth

default bandwidth

Parameters

packets Minimum data rate in packets per second. Value ranges from ***1*** to ***100000***.

Related Commands

- class (policy-map (control-plane) Helix) places the switch in ***policy-map-class*** (control plane) configuration mode.
- shape (policy-map-class (control-plane) Helix) specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Helix platform switches define these default bandwidths for control plane static classes:

- copp-system-acllog 1000 copp-system-l3ttl1 1000
- copp-system-arp 1000 copp-system-lacp 5000

- copp-system-arpresolver 1000 copp-system-lldp 1000
- copp-system-bfd 5000 copp-system-mlag 5000
- copp-system-bgp 5000 copp-system-OspfIsis 5000
- copp-system-bpdu 5000 copp-system-selfip 5000
- copp-system-default 1000 copp-system-selfip-tc6to7 5000
- copp-system-glean 1000 copp-system-sflow 0
- copp-system-igmp 1000 copp-system-tc3to5 1000
- copp-system-ipmcmiss 1000 copp-system-tc6to7 1000
- copp-system-ipmcrsvd 1000 copp-system-urm 1000
- copp-system-l3destmiss 1000 copp-system-vrrp 1000
- copp-system-l3slowpath 1000

Example

These commands configure the minimum bandwidth of ***500*** packets per second for data traffic specified by the class map **copp-system-lldp**.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# bandwidth pps 500
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# exit
switch(config-pmap-copp-system-policy)# exit
switch(config)# show policy-map interface control-plan copp-system-policy
Service-policy input: copp-system-policy
  Number of units programmed: 4
  Hardware programming status: Successful

  Class-map: copp-system-lldp (match-any)
       shape : 10000 pps
       bandwidth : 500 pps
      Out Packets : 304996
      Drop Packets : 0

switch(config)#
```

10.2.7.5 bandwidth (policy-map-class (control-plane) Petra)

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration

accessed through class (policy-map (control-plane) Petra)

Command Syntax

bandwidth kbps ***kilobits***

no bandwidth

default bandwidth

Parameters

kbits Minimum data rate in kilobits per second. Value ranges from ***1*** to ***10000000***.

Related Commands

- class (policy-map (control-plane) Petra) places the switch in ***policy-map-class*** (control plane) configuration mode.
- shape (policy-map-class (control-plane) Petra) specifies the maximum bandwidth for traffic defined by the associated class map in its ***policy map class*** configuration mode .

Static Classes Default Bandwidth

Petra platform switches define these default bandwidths for control plane static classes:

- copp-system-bpdu 1250 copp-system-l3destmiss 250
- copp-system-default 250 copp-system-l3slowpath 250
- copp-system-igmp 250 copp-system-l3ttl0 250
- copp-system-ipbroadcast 250 copp-system-l3ttl1 250
- copp-system-ipmc 250 copp-system-lacp 1250
- copp-system-ipmcmiss 250 copp-system-lldp 250
- copp-system-ipmcrsvd 250 copp-system-unicast-arp 250
- copp-system-ipunicast 250

Guidelines

Petra does not support all discrete rate values. When a specified discrete value is not supported, the switch converts the rate to the next highest discrete value that it supports. The **`show`** command displays the converted rate and not the user-configured rate.

Example

These commands configure a minimum bandwidth of ***500*** kbps for data traffic specified by the class map ***copp-system-lldp*** of the default ***control-plane*** policy map. Because the switch does not support the discrete value of ***500*** kbps, it converts the bandwidth up to ***651*** kbps.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# bandwidth kbps 500
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# exit
switch(config-pmap-copp-system-policy)# exit
switch(config)# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy
  Hardware programming status: InProgress

  Class-map: copp-system-lldp (match-any)
       shape : 2766 kbps
       bandwidth : 651 kbps
      Out Packets : 0
      Drop Packets : 0

switch(config)#
```

10.2.7.6 bandwidth (policy-map-class (control-plane) Trident II)

The **`bandwidth`** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **`no bandwidth`** and **`default bandwidth`** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **`bandwidth`** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration

accessed through `class (policy-map (control-plane) Trident II)`.

Command Syntax

bandwidth pps ***packets***

no bandwidth

default bandwidth

Parameters

packets Minimum data rate in packets per second. Value ranges from *1* to ***100000***.

Related Commands

- class (policy-map (control-plane) Trident II) places the switch in ***policy-map-class*** (control plane) configuration mode.
- shape (policy-map-class (control-plane) Trident II) specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Trident II platform switches define these default bandwidths for control plane static classes:

- copp-system-acllog 1000 copp-system-l3slowpath 1000
- copp-system-arp 1000 copp-system-l3ttl1 1000
- copp-system-arpresolver 1000 copp-system-lacp 5000
- copp-system-bfd 5000 copp-system-lldp 1000
- copp-system-bgp 5000 copp-system-mlag 5000
- copp-system-bpdu 5000 copp-system-selfip 5000
- copp-system-default 1000 copp-system-selfip-tc6to7 5000
- copp-system-glean 1000 copp-system-sflow 0
- copp-system-igmp 1000 copp-system-tc3to5 1000
- copp-system-ipmcmiss 1000 copp-system-tc6to7 1000
- copp-system-ipmcrsvd 1000 copp-system-urm 1000
- copp-system-l3destmiss 1000

Example

These commands configure the minimum bandwidth of ***500*** packets per second for data traffic specified by the class map **copp-system-lldp**.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# bandwidth pps 500
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# exit
switch(config-pmap-copp-system-policy)# exit
switch(config)# show policy-map interface control-plan copp-system-policy
Service-policy input: copp-system-policy
  Number of units programmed: 4
  Hardware programming status: Successful

  Class-map: copp-system-lldp (match-any)
       shape : 10000 pps
       bandwidth : 500 pps
      Out Packets : 304996
      Drop Packets : 0

switch(config)#
```

10.2.7.7 bandwidth (policy-map-class (control-plane) Trident)

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration

accessed through `class (policy-map (control-plane) Trident)`.

Command Syntax

bandwidth pps ***packets***

no bandwidth

default bandwidth

Parameters

packets Minimum data rate in packets per second. Value ranges from ***1*** to ***100000***.

Related Commands

- class (policy-map (control-plane) Trident) places the switch in ***policy-map-class*** (control plane) configuration mode.
- shape (policy-map-class (control-plane) Trident) specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Trident platform switches define these default bandwidths for control plane static classes:

- copp-system-arp 1000 copp-system-lldp 1000
- copp-system-arpresolver 1000 copp-system-l3destmiss 1000
- copp-system-bpdu 5000 copp-system-l3slowpath 1000
- copp-system-default 1000 copp-system-l3ttl1 1000
- copp-system-glean 1000 copp-system-selfip 5000
- copp-system-igmp 1000 copp-system-selfip-tc6to7 5000
- copp-system-ipmcmiss 1000 copp-system-sflow 0
- copp-system-ipmcrsvd 1000 copp-system-tc6to7 1000
- copp-system-lacp 5000 copp-system-tc3to5 1000

Example

These commands configure the minimum bandwidth of ***1000*** packets per second for data traffic specified by the class map ***PMAP-1*** in the policy map named ***copp-system-policy***.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class PMAP-1
switch(config-pmap-c-copp-system-policy-PMAP-1)# bandwidth pps 1000
switch(config-pmap-c-copp-system-policy-PMAP-1)#
```

10.2.7.8 class (policy-map (control-plane) Arad)

The **class** command places the switch in policy-map-class (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains ***20*** static classes. Each class contains an eponymous class map and may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **bandwidth** command defines the streams minimum transmission rate through the control plane.
- **shape** command defines the streams maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packets content. Dynamic classes are not supported for control plane policing on Arad platform switches.

Each class corresponds to a transmission queue. Queue scheduling is round-robin until ***bandwidth*** rate for a queue is exceeded. Scheduling becomes strict-priority with CPU queue number determining priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **`exit`** command returns the switch to policy-map configuration mode. Saving policy-map-class changes also require an exit from policy-map mode, which saves pending policy-map-class and policy-map changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **`abort`** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **`no class`** and **`default class`** commands remove `policy-map-class` commands for the specified class assignment from the policy map.

Command Mode

Policy-Map (control plane) configuration accessed through `policy-map type copp` command.

Command Syntax

`class` ***class_name***

`no class` ***class_name***

`default class` ***class_name***

Parameters

class_name name of the class.

Static Classes

Arad platform switches provide the following static control plane classes:

- copp-system-bgp copp-system-l2broadcast copp-system-linklocal
- copp-system-bpdu copp-system-l2unicast copp-system-lldp
- copp-system-default copp-system-l3destmiss copp-system-mlag
- copp-system-ipbroadcast copp-system-l3lpmoverflow copp-system-multicastsnoop
- copp-system-ipmc copp-system-l3slowpath copp-system-OspfIsis
- copp-system-ipmcmiss copp-system-l3ttl1 copp-system-sflow
- copp-system-ipunicast copp-system-lacp

Commands Available in Policy-map-class (control plane) Configuration Mode

- bandwidth (policy-map-class (control-plane) Arad)
- shape (policy-map-class (control-plane) Arad)
- **`exit`** saves pending class map changes, then returns the switch to global configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

policy-map type copp places switch in ***policy-map*** (control plane) configuration mode.

Example

These commands enters ***policy-map-class*** configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-lldp***.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

10.2.7.9 class (policy-map (control-plane) FM6000)

The **class** command places the switch in ***policy-map-class*** (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains ***12*** static classes. Each class contains an eponymous class map and may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **bandwidth** command defines the streams minimum transmission rate through the control plane.
- **shape** command defines the streams maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packets content. Dynamic classes are not supported for control plane policing on FM6000 platform switches.

Each class corresponds to a transmission queue. Queue scheduling is round-robin until ***bandwidth*** rate for a queue is exceeded. Scheduling becomes strict-priority with CPU queue number determining priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **exit** command returns the switch to policy-map configuration mode. Saving policy-map-class changes also require an exit from policy-map mode, which saves pending policy-map-class and policy-map changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no class** and **default class** commands remove **policy-map-class** commands for the specified class assignment from the policy map. The class is removed from the policy map if it is a dynamic class.

Command Mode

Policy-Map (control plane) configuration accessed through `policy-map type copp` command.

Command Syntax

class ***class_name***

no class ***class_name***

default class ***class_name***

Parameters

class_name name of the class.

Static Classes

FM6000 platform switches provide the following static control plane classes:

- copp-system-arp copp-system-igmp copp-system-PimPtp
- copp-system-default copp-system-l2rsvd copp-system-selfip
- copp-system-ipmcmiss copp-system-l3slowpath copp-system-selfip-tc6to7
- copp-system-ipmcrsvd copp-system-OspfIsis copp-system-sflow

Commands Available in Policy-map-class (control plane) Configuration Mode

- bandwidth (policy-map-class (control-plane) FM6000)
- shape (policy-map-class (control-plane) FM6000)
- **exit** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **abort** discards pending class map changes, then returns the switch to the ***global*** configuration mode.

Related Commands

policy-map type copp places switch in ***policy-map*** (control plane) configuration mode.

Example

These commands enters ***policy-map-class*** configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-arp***.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-arp
switch(config-pmap-c-copp-system-policy-copp-system-arp)#
```

10.2.7.10 class (policy-map (control-plane) Helix)

The **`class`** command places the switch in ***policy-map-class*** (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The **`control plane`** policy map contains *23* static classes. Each class contains an eponymous class map and may contain **`bandwidth`** and **`shape`** commands.

- The class map identifies a data stream.
- **`bandwidth`** command defines the streams minimum transmission rate through the control plane.
- **`shape`** command defines the streams maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packets content. Dynamic classes are not supported for control plane policing on Helix platform switches.

Each class corresponds to a transmission queue. Queue scheduling is strict-priority; CPU queue number determines priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **`exit`** command returns the switch to ***policy-map*** configuration mode. Saving policy-map-class changes also require an exit from ***policy-map*** mode, which saves the pending ***policy-map-class*** and ***policy-map*** changes to ***running-config*** and returns the switch to global configuration mode. The **`abort`** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **`no class`** and **`default class`** commands remove the ***policy-map-class*** commands for the specified class assignment from the policy map.

Command Mode

Policy-Map (control plane) configuration accessed through `policy-map type copp` command.

Command Syntax

`class` ***class_name***

`no class` ***class_name***

`default class` ***class_name***

Parameters

class_name name of the class.

Static Classes

Helix platform switches provide the following static control plane classes:

- copp-system-acllog copp-system-ipmcmiss copp-system-OspfIsis
- copp-system-arp copp-system-ipmcrsvd copp-system-selfip
- copp-system-arpresolver copp-system-l3destmiss copp-system-selfip-tc6to7
- copp-system-bfd copp-system-l3slowpath copp-system-sflow
- copp-system-bgp copp-system-l3ttl1 copp-system-tc3to5
- copp-system-bpdu copp-system-lacp copp-system-tc6to7
- copp-system-default copp-system-lldp copp-system-urm
- copp-system-glean copp-system-lldp copp-system-vrrp

- copp-system-igmp copp-system-lldp

Commands Available in Policy-map-class (control plane) Configuration Mode

- bandwidth (policy-map-class (control-plane) Helix)
- shape (policy-map-class (control-plane) Helix)
- **exit** saves pending class map changes, then returns the switch to the *global* configuration mode.
- **abort** discards pending class map changes, then returns the switch to the ***global*** configuration mode.

Related Commands

policy-map type copp places switch in ***policy-map*** (control plane) configuration mode.

Example

These commands enters ***policy-map-class*** configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-arp***.

```
switch(config)# policy-map
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

10.2.7.11 class (policy-map (control-plane) Petra)

The **class** command places the switch in policy-map-class (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains 15 static classes. Each class contains an eponymous class map and may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **bandwidth** command defines the streams minimum transmission rate through the control plane.
- **shape** command defines the streams maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packets content. Dynamic classes are not supported for control plane policing on Petra platform switches.

Each class corresponds to a transmission queue. Queue scheduling is round-robin until ***bandwidth*** rate for a queue is exceeded. Scheduling becomes strict-priority with CPU queue number determining priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **exit** command returns the switch to ***policy-map*** configuration mode. Saving the ***policy-map-class*** changes also require an exit from ***policy-map*** mode, which saves the pending ***policy-map-class*** and ***policy-map*** changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no class** and **default class** commands remove the **policy-map-class** commands for the specified class assignment from the policy map.

Command Mode

Policy-Map (control plane) configuration accessed through policy-map type copp command.

Command Syntax

class ***class_name***

no class ***class_name***

default class ***class_name***

Parameters

class_name name of the class.

Static Classes

Petra platform switches provide the following static control plane classes:

- copp-system-bpdu copp-system-ipmcmiss copp-system-l3ttl0
- copp-system-default copp-system-ipmcrsvd copp-system-l3ttl1
- copp-system-igmp copp-system-ipunicast copp-system-lacp
- copp-system-ipbroadcast copp-system-l3destmiss copp-system-lldp
- copp-system-ipmc copp-system-l3slowpath copp-system-unicast-arp

Commands Available in Policy-map-class (control plane) Configuration Mode

- bandwidth (policy-map-class (control-plane) Petra)
- shape (policy-map-class (control-plane) Petra)
- **`exit`** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to the ***global*** configuration mode.

Related Commands

policy-map type copp places switch in ***policy-map*** (control plane) configuration mode.

Example

These commands enters ***policy-map-class*** configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-lldp***.

```
switch(config)# policy-map
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

10.2.7.12 class (policy-map (control-plane) Trident II)

The **`class`** command places the switch in ***policy-map-clas***s (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains ***23*** static classes. Each class contains an eponymous class map and may contain **`bandwidth`** and **`shape`** commands.

- The class map identifies a data stream.
- **`bandwidth`** command defines the streams minimum transmission rate through the control plane.
- **`shape`** command defines the streams maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packets content. Dynamic classes are not supported for control plane policing on Trident II platform switches.

Each class corresponds to a transmission queue. Queue scheduling is strict-priority; CPU queue number determines priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **`exit`** command returns the switch to the ***policy-map*** configuration mode. Saving the ***policy-map-class*** changes also require an exit from the ***policy-map*** mode, which saves the pending ***policy-map-class*** and ***policy-map*** changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **`abort`** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **`no class`** and **`default class`** commands remove the ***policy-map-class*** commands for the specified class assignment from the policy map.

Command Mode

Policy-Map (control plane) configuration accessed through `policy-map type copp` command.

Command Syntax

`class` ***class_name***

`no class` ***class_name***

`default class`***class_name***

Parameters

class_name name of the class.

Static Classes

Trident II platform switches provide the following static control plane classes:

- copp-system-acllog copp-system-igmp copp-system-mlag
- copp-system-arp copp-system-ipmcmiss copp-system-selfip
- copp-system-arpresolver copp-system-ipmcrsvd copp-system-selfip-tc6to7
- copp-system-bfd copp-system-l3destmiss copp-system-sflow
- copp-system-bgp copp-system-l3slowpath copp-system-tc3to5
- copp-system-bpdu copp-system-l3ttl1 copp-system-tc6to7
- copp-system-default copp-system-lacp copp-system-urm
- copp-system-glean copp-system-lldp

Commands Available in Policy-map-class (control plane) Configuration Mode

- bandwidth (policy-map-class (control-plane) Trident II)
- shape (policy-map-class (control-plane) Trident II)
- **`exit`** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to the ***global*** configuration mode.

Related Commands

policy-map type copp places switch in ***policy-map (control plane)*** configuration mode.

Example

These commands enters the ***policy-map-class*** configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-arp***.

```
switch(config)# policy-map
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

10.2.7.13 class (policy-map (control-plane) Trident)

The **`class`** command places the switch in ***policy-map-class*** (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. The command adds the specified class to the policy map if it was not previously included. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains 18 static classes and up to *30* dynamic classes. Dynamic classes contain an eponymous class map. All classes may contain **`bandwidth`** and **`shape`** commands.

- The class map identifies a data stream.
- **`bandwidth`** command defines the streams minimum transmission rate through the control plane.
- **`shape`** command defines the streams maximum transmission rate through the control plane.

Dynamic class maps identify a data stream with an ACL assigned by `match (class-map (control-plane) Trident)`. Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packets content.

Static classes are provided with the switch and cannot be removed from the policy map or modified by the **`class`** command. Dynamic classes are user defined and added to the policy map by this command. Dynamic classes are always placed in front of the static classes. Bandwidth and shape parameters are editable for all classes.

Each class corresponds to a transmission queue. Queue scheduling is round-robin until ***bandwidth*** rate for a queue is exceeded. Scheduling becomes strict-priority with CPU queue number determining priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **`exit`** command returns the switch to policy-map configuration mode. Saving the ***policy-map-class*** changes also require an exit from ***policy-map*** mode, which saves the pending ***policy-map-class*** and ***policy-map*** changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **`abort`** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **`no class`** and **`default class`** commands remove the ***policy-map-class*** commands for the specified class assignment from the policy map. The class is removed from the policy map if it is a dynamic class.

Command Mode

Policy-Map (control plane) configuration accessed through `policy-map type copp` command.

Command Syntax

`class` ***class_name*** [**PLACEMENT**]

`no class` ***class_name*** [**PLACEMENT**]

`default class` ***class_name*** [**PLACEMENT**]

Parameters

- ***class_name*** name of the class.
- **PLACEMENT** Specifies the classs map placement. Configurable only for dynamic classes.
  - ***no parameter*** New classes are placed between the dynamic and static classes. Previously defined classes retain their current policy map placement.
  - **insert-before** ***dynamic_class*** Class is inserted in front of the specified dynamic class.

Static Classes

Trident switches provide the following static control plane classes:

- copp-system-acllog copp-system-ipmcmiss copp-system-lldp
- copp-system-arp copp-system-ipmcrsvd copp-system-selfip
- copp-system-arpresolver copp-system-l3destmiss copp-system-selfip-tc6to7
- copp-system-bpdu copp-system-l3slowpath copp-system-sflow
- copp-system-glean copp-system-l3ttl1 copp-system-tc3to5
- copp-system-igmp copp-system-lacp copp-system-tc6to7

Commands Available in Policy-map-class (control plane) Configuration Mode

- bandwidth (policy-map-class (control-plane) Trident)
- shape (policy-map-class (control-plane) Trident)
- **`exit`** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to the ***global*** configuration mode.

Related Commands

- class-map type copp places switch in the ***class-map*** (control-plane) configuration mode.
- policy-map type copp places switch in the ***policy-map*** (control plane) configuration mode.

Example

These commands add ***CM-1*** class to the ***copp-system-policy*** policy map.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class CM-1
switch(config-pmap-c-copp-system-policy-CM-1)#
```

10.2.7.14 class (policy-map (pbr)

The **`class (policy-map (pbr)`** command places the switch in ***policy-map-class (pbr)*** configuration mode, which is a group change mode that modifies the specified class of the configuration mode Policy-Based Routing (PBR) policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the mode is exited, and can be canceled by using the **`abort`** command.

A PBR policy map is an ordered list of classes. Each class contains an eponymous class map and can contain set commands to specify next hop. Classes without set commands translate to no action being performed on that class of packets.

- The class map identifies a data stream through ACLs. Class maps are configured in the ***class-map*** (pbr) configuration mode.
- **`Set`** commands can be used to specify the next hop for a given class. **`Set`** commands are configured in ***policy-map-class*** (pbr) configuration mode.

PBR policy maps can also contain one or more raw match statements which filter incoming traffic without using ACLs. Data packets are managed by commands of the first class or raw match statement matching the packets contents.

The **`exit`** command returns the switch to the ***policy-map*** (pbr) configuration mode. However, saving the policy-map-class changes also requires an exit from ***policy-map*** (pbr) configuration mode. This saves all the pending policy map and policy-map-class changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **`abort`** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **`no class`** and **`default class`** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **`class`** configuration from ***running-config***.

Command Mode

Policy-Map (pbr) Configuration accessed through `policy-map type pbr`.

Command Syntax

[***sequence_number***] **`class`** ***class_name***

`no` [***sequence_number***] **`class`** ***class_name***

`default` [***sequence_number***] **`class`** ***class_name***

`no` [***sequence_number***]

`default` [***sequence_number***]

Parameters

- ***sequence_number*** Sequence number (***1*** to ***4294967295***) assigned to the rule. If no number is entered, the number is derived by adding ***10*** to the number of the policy maps last numbered line. To increase the distance between existing entries, use the **`resequence`** command.
- ***class_name*** name of the class.

Commands Available in Policy-map-class (pbr) Configuration Mode

- set nexthop (policy-map-class pbr) sets next hop for the class.

- **exit** saves pending class changes and returns switch to ***policy-map (pbr)*** configuration mode.
- **abort** discards pending class changes and returns switch to ***policy-map (pbr)*** configuration mode.

Related Commands

- class-map type pbr places switch in the ***class-map*** (pbr) configuration mode.
- policy-map type pbr places switch in the ***policy-map (pbr)*** configuration mode.

Example

These commands add the ***CMAP1*** class map to the ***PMAP1*** policy map, then place the switch in ***policy-map-class*** configuration mode where the next hops can be assigned to the class. Changes will not take effect until both modes are exited.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)# class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

10.2.7.15 class (policy-map (qos) FM6000)

The **class** command places the switch in ***policy-map-class*** (qos) configuration mode, which is a group change mode that modifies the specified class of the configuration mode policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. Each class contains an eponymous class map and at least one set command:

- The class map identifies a data stream through an ACL. Class maps are configured in the ***class-map*** (qos) configuration mode.
- **Set** commands either modify a packets content (CoS or DSCP fields) or assigns it to a traffic class queue. **Set** commands are configured in the ***policy-map-class*** (qos) configuration mode.

  Data packets are managed by commands of the first class whose map matches the packets content.

The **exit** command returns the switch to the ***policy-map*** configuration mode. However, saving policy-map-class changes also require an exit from the ***policy-map*** mode. This saves all pending policy map and policy-map-class changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (qos) Configuration accessed through `policy-map type quality-of-service`.

Command Syntax

class ***class_name*** [**PLACEMENT**]

no class ***class_name*** [**PLACEMENT**]

default class ***class_name*** [**PLACEMENT**]

Parameters

- ***class_name*** name of the class.
- **PLACEMENT** Specifies the map placement within the list of class maps.
  - ***no parameter*** Class is placed at the top of the list.
  - **insert-before** ***existing_class*** Class is inserted in front of the specified class.

Commands Available in Policy-map-class (qos) Configuration Mode

- set (policy-map-class (qos) FM6000)
- **exit** saves pending class changes and returns switch to ***policy-map (qos)*** configuration mode.
- **abort** discards pending class changes and returns switch to ***policy-map (qos)*** configuration mode.

Related Commands

- class-map type qos places switch in the ***class-map*** (QoS) configuration mode.
- policy-map type quality-of-service places switch in the ***policy-map*** (QoS) configuration mode

Example

These commands add the ***CMAP_1*** class map to the ***PMAP_1*** policy map, then places the switch in the ***policy-map-class*** configuration mode.

```
switch(config)# policy-map type quality-of-service PMAP-1
switch(config-pmap-PMAP-1)# class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

10.2.7.16 class (policy-map (qos) Helix)

The **class** command places the switch in the ***policy-map-class*** (QoS) configuration mode, which is a group change mode that modifies the specified class of the configuration mode policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. Each class contains an eponymous class map and at least one set command:

- The class map identifies a data stream through an ACL. Class maps are configured in the ***class-map*** (qos) configuration mode.
- **Set** commands either modify a packets content (CoS or DSCP fields) or assigns it to a traffic class queue. **Set** commands are configured in the ***policy-map-class*** (qos) configuration mode.

  Data packets are managed by commands of the first class whose map matches the packets content.

The **exit** command returns the switch to the ***policy-map*** configuration mode. However, saving policy-map-class changes also require an exit from the ***policy-map*** mode. This saves all the pending policy map and policy-map-class changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (qos) Configuration accessed through `policy-map type quality-of-service` command.

Command Syntax

class ***class_name*** [**PLACEMENT**]

no class ***class_name*** [**PLACEMENT**]

default class ***class_name*** [**PLACEMENT**]

Parameters

- ***class_name*** name of the class.
- **PLACEMENT** Specifies the map placement within the list of class maps.
  - ***no parameter*** Class is placed at the top of the list.
  - **insert-before** ***existing_class*** Class is inserted in front of the specified class.

Commands Available in Policy-map-class (QoS) Configuration Mode

- set (policy-map-class (qos) Helix)
- **exit** saves pending class changes and returns switch to ***policy-map (qos)*** configuration mode.
- **abort** discards pending class changes and returns switch to ***policy-map (qos)*** configuration mode.

Related Commands

- class-map type qos places switch in the ***class-map*** (qos) configuration mode.
- policy-map type quality-of-service places switch in the ***policy-map*** (QoS) configuration mode.

Example

These commands add the ***CMAP_1*** class map to the ***PMAP_1*** policy map, then places the switch in ***policy-map-class*** configuration mode.

```
switch(config)# policy-map type quality-of-service PMAP-1
switch(config-pmap-PMAP-1)# class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

10.2.7.17 class (policy-map (qos) Trident II)

The **class** command places the switch in the ***policy-map-class*** (QoS) configuration mode, which is a group change mode that modifies the specified class of the configuration mode policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. Each class contains an eponymous class map and at least one set command:

- The class map identifies a data stream through an ACL. Class maps are configured in ***class-map (qos)*** configuration mode.
- **Set** commands either modify a packets content (CoS or DSCP fields) or assigns it to a traffic class queue. **Set** commands are configured in ***policy-map-class (qos)*** configuration mode.

Data packets are managed by commands of the first class whose map matches the packets content.

The **exit** command returns the switch to the ***policy-map*** configuration mode. However, saving the policy-map-class changes also require an exit from the ***policy-map*** mode. This saves all the pending policy map and policy-map-class changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (qos) Configuration accessed through `policy-map type quality-of-service` command.

Command Syntax

class ***class_name*** [**PLACEMENT**]

no class ***class_name*** [**PLACEMENT**]

default class ***class_name*** [**PLACEMENT**]

Parameters

- ***class_name*** name of the class.
- **PLACEMENT** Specifies the map placement within the list of class maps.
  - ***no parameter*** Class is placed at the top of the list.
  - **insert-before** ***existing_class*** Class is inserted in front of the specified class.

Commands Available in Policy-map-class (qos) Configuration Mode

- set (policy-map-class (qos) Trident II)
- **exit** saves pending class changes and returns switch to ***policy-map (qos)*** configuration mode.
- **abort** discards pending class changes and returns switch to ***policy-map (qos)*** configuration mode.

Related Commands

- class-map type qos places switch in ***class-map (qos)*** configuration mode.
- policy-map type quality-of-service places switch in ***policy-map (qos)*** configuration mode.

Example

These commands add the ***CMAP_1*** class map to the ***PMAP_1*** policy map, then places the switch in ***policy-map-class*** configuration mode.

```
switch(config)# policy-map type quality-of-service PMAP-1
switch(config-pmap-PMAP-1)# class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

10.2.7.18 class (policy-map (qos) Trident)

The **class** command places the switch in ***policy-map-class (qos)*** configuration mode, which is a group change mode that modifies the specified class of the configuration mode policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. Each class contains an eponymous class map and at least one set command:

- The class map identifies a data stream through an ACL. Class maps are configured in ***class-map (qos)*** configuration mode.
- **Set** commands either modify a packets content (CoS or DSCP fields) or assigns it to a traffic class queue. **Set** commands are configured in ***policy-map-class (qos)*** configuration mode.

  Data packets are managed by commands of the first class whose map matches the packets content.

The **exit** command returns the switch to ***policy-map*** configuration mode. However, saving policy-map-class changes also require an exit from ***policy-map*** mode. This saves all the pending policy map and policy-map-class changes to ***running-config*** and returns the switch to the ***global*** configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (qos) Configuration accessed through policy-map type quality-of-service command.

Command Syntax

class ***class_name*** [**PLACEMENT**]

no class ***class_name*** [**PLACEMENT**]

default class ***class_name*** [**PLACEMENT**]

Parameters

- ***class_name*** name of the class.
- **PLACEMENT** Specifies the map placement within the list of class maps.
  - ***no parameter*** Class is placed at the top of the list.
  - **insert-before** ***existing_class*** Class is inserted in front of the specified class.

Commands Available in Policy-map-class (qos) Configuration Mode

- set (policy-map-class (qos) Trident)
- **exit** saves pending class changes and returns switch to ***policy-map (qos)*** configuration mode.
- **abort** discards pending class changes and returns switch to ***policy-map (qos)*** configuration mode.

Related Commands

- class-map type qos places switch in ***class-map (qos)*** configuration mode.
- policy-map type quality-of-service places switch in ***policy-map (qos)*** configuration mode.

Example

These commands add the ***CMAP_1*** class map to the ***PMAP_1*** policy map, then places the switch in ***policy-map-class*** configuration mode.

```
switch(config)# policy-map type quality-of-service PMAP-1
switch(config-pmap-PMAP-1)# class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

10.2.7.19 class-map type copp

The **class-map type copp** command places the switch in ***Class-Map*** (control plane) configuration mode, which is a group change mode that modifies a control-plane dynamic class map. A dynamic class map is a data structure that uses Access Control Lists (ACLs) to define a data stream by specifying characteristics of data packets that comprise that stream. Control-plane policy maps use class maps to specify which control plane traffic is controlled by policy map criteria.

The **exit** command saves pending class map changes to ***running-config*** and returns the switch to the ***global*** configuration mode. Class map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes and returns the switch to the ***global*** configuration mode.

The **no class-map type copp** and **default class-map type copp** commands delete the specified class map by removing the corresponding **class-map type copp** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

class-map type copp match-any ***class_name***

no class-map type copp [match-any] ***class_name***

default class-map type copp [match-any] ***class_name***

Parameters

class_name Name of class map.

Commands Available in Class-Map (Control Plane) Configuration Mode

match (class-map (control-plane) Trident)

Related Commands

- policy-map type copp
- class (policy-map (control-plane) Trident)
- class-map type qos

Example

This command creates the control plane class map named ***CP-MAP-1*** and places the switch in ***class-map*** configuration mode.

```
switch(config)# class-map type copp match-any CP-CMAP-1
switch(config-cmap-CP-CMAP-1)#
```

10.2.7.20 class-map type pbr

The **`class-map type pbr`** command places the switch in the ***class-map*** (pbr) configuration mode for the specified class map, and creates the class map if one does not already exist. The ***class-map*** (PBR) configuration mode is a group change mode that modifies a class map for Policy-Based Routing (PBR). PBR class maps contain one or more **match** statements which filter incoming traffic using ACLs. PBRs can then use these class maps to set next-hop IP addresses for the traffic that matches them. (Classes without set commands translate to no action being performed on that class of packets.)

The **`exit`** command saves pending class map changes to ***running-config***, then returns the switch to the ***global*** configuration mode. Class map changes are also saved by directly entering a different configuration mode. The **`abort`** command discards pending changes and returns the switch to the ***global*** configuration mode.

The **`no class-map type pbr`** and **`default class-map type pbr`** commands delete the specified class map by removing the corresponding **`class-map type pbr`** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

`class-map type pbr match-any` ***map_name***

`no class-map type pbr match-any` ***map_name***

`default class-map type pbr match-any` ***map_name***

Parameters

map_name Name of class map.

Commands Available in Class-Map (PBR) configuration mode

- match (class-map (pbr))
- resequence (class-map (pbr))

Related Commands

- policy-map type pbr
- class (policy-map (pbr))

Example

This command creates the PBR class map named ***MAP1*** and places the switch in ***class-map (pbr)*** configuration mode where match criteria can be configured for the class.

```
switch(config)# class-map type pbrmatch-any MAP1
switch(config-cmap-MAP1)#
```

10.2.7.21 class-map type qos

The **`class-map type qos`** command places the switch in the ***class-map*** (QoS) configuration mode, which is a group change mode that modifies a QoS dynamic class map. A dynamic class map is a data structure that uses Access Control Lists (ACLs) to define a data stream by specifying characteristics of data packets

that comprise that stream. QoS policy maps use class maps to specify the traffic (to which the policy map is assigned) that is transformed by policy map criteria.

The **exit** command saves pending class map changes to ***running-config***, then returns the switch to the ***global*** configuration mode. Class map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes and returns the switch to the ***global*** configuration mode.

The **no class-map type qos** and **default class-map type qos** commands delete the specified class map by removing the corresponding **class-map type qos** command and its associated configuration. The **class-map** and **class-map type qos** commands are equivalent.

Command Mode

Global Configuration

Command Syntax

class-map [type qos] match-any ***class_name***

no class-map [type qos] match-any ***class_name***

default class-map [type qos] match-any ***class_name***

Parameters

class_name Name of class map.

Commands Available in Class-Map (QoS) Configuration Mode

- match (class-map (qos) FM6000)
- match (class-map (qos) Trident)

Conditions

class-map ***map_name*** and **class-map type qos** ***map_name*** are identical commands.

Related Commands

- policy-map type quality-of-service
- class (policy-map (qos) FM6000)
- class (policy-map (qos) Trident)

Example

This command creates the QoS class map named ***MAP-1*** and places the switch in ***class-map*** configuration mode.

```
switch(config)# class-map type qos match-any MAP-1
switch(config-cmap-MAP-1)#
```

10.2.7.22 clear policy-map counters

The **clear policy-map** command resets the specified policy map counters to zero. Policy map counters record the quantity of packets that are filtered by the ACLs that comprise a specified policy map.

Command Mode

Privileged EXEC

Command Syntax

clear policy-map **INTERFACE_NAME** **counters** **MAP_NAME**

Parameters

- **INTERFACE_NAME** Interface for which command clears table counters. Options include:
  - **interface control-plane** Control plane.

- **MAP_NAME** Policy map for which command clears counters. Options include:
  - ***copp-system-policy*** Name of only policy map supported for the control plane.

10.2.7.23 feature pbr

Policy-Based Routing (PBR) is a feature that is applied on IPv4 or IPv6 routable ports, to preferentially route packets. Forwarding is based on a policy that is enforced at the ingress of the applied interface and overrides normal routing decisions. In addition to matches on regular ACLs, PBR policy-maps can also include “raw match” statements that look like a single entry of an ACL as a convenience for users.

Configuration Mode

For IP:

TCAM PBR profile set TTL configuration mode.

For IPv6:

TCAM feature PBR IP configuration mode.

Command Syntax

For IP:

`feature pbr ip` [copy]

`no feature pbr ip` [copy]

`default featue pbr ip` [copy]

For IPv6:

`feature pbr ipv6`[copy | bank]

`no feature pbr ipv6` [copy | bank]

`default featue pbr ipv6` [copy | bank]

Parameters

For IP:

copy Copy a feature from a TCAM profile.

For IPv6:

- **copy** Copy a feature from a TCAM profile.
- **bank** TCAM banks to reserve.

Example

In the following example, the PBR is configured on an IP routable port.

```
(config)# hardware tcam
(config-tcam)# profile pbr-set-ttl copy default
(config-tcam-profile-pbr-set-ttl)# feature pbr ip
```

In the following example, the PBR is configured on an IPv6 routable port.

```
(config)# hardware tcam
(config-tcam)# profile pbr-set-ttl copy default
(config-tcam-profile-pbr-set-ttl)# feature pbr ip
(config-tcam-feature-pbr-ip)# feature pbr ipv6
```

10.2.7.24 feature traffic-policy cpu

The **feature traffic-policy cpu** command configures the CPU traffic policy features for the IPv4 and IPv6 traffic in user-defined TCAM profile.

The **no feature traffic-policy cpu** and **default feature traffic-policy cpu** commands remove the CPU policy configurations from ***running-config***.

Command Mode

Hardware TCAM

Command Syntax

feature traffic-policy cpu [ipv4 | ipv6]

no feature traffic-policy cpu [ipv4 | ipv6]

default feature traffic-policy cpu [ipv4 | ipv6]

Parameters

- **ipv4** CPU traffic policy for IPv4 traffic.
- **ipv6** CPU traffic policy for IPv6 traffic.

Example

These commands places the switch in the hardware TCAM profile mode and configures the CPU traffic policy features for IPv4 traffic in the TCAM profile test.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile test
switch(config-hw-tcam-profile-test)# feature traffic-policy cpu ipv4
```

10.2.7.25 feature traffic-policy port

The **feature traffic-policy port** command configures the port-related traffic policy features for the IPv4 and IPv6 traffic in user-defined TCAM profile.

The **no feature traffic-policy port** and **default feature traffic-policy port** commands remove the CPU policy configurations from ***running-config***.

Command Mode

Hardware TCAM

Command Syntax

feature traffic-policy port [ipv4 | ipv6]

no feature traffic-policy port [ipv4 | ipv6]

default feature traffic-policy port [ipv4 | ipv6]

Parameters

- **ipv4** port traffic policy for IPv4 traffic.
- **ipv6** port traffic policy for IPv6 traffic.

Example

These commands places the switch in the hardware TCAM profile mode and configures the port traffic policy features for IPv4 traffic in the TCAM profile test.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile test
switch(config-hw-tcam-profile-test)# feature traffic-policy port ipv4
```

10.2.7.26 match (class-map (control-plane) Helix)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4 and IPv4 standard ACLs.

A class map is assigned to a policy map by the `class (policy-map (control-plane) Helix)` command.

The ***class map*** (control plane) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (control plane) configuration accessed through `class-map type copp` command.

Command Syntax

match ip access-group ***list_name***

no match ip access-group ***list_name***

default match ip access-group ***list_name***

Parameters

list_name name of ACL assigned to class map.

Related Commands

- class-map type copp places the switch in the ***class-map*** configuration mode.
- **exit** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **abort** discards pending class map changes, then returns the switch to the ***global*** configuration mode.
- class (policy-map (control-plane) Helix) assigns a ***class map*** to a ***policy map***.

Guidelines

Static class maps cannot be modified by this command.

Match statements are saved to ***running-config*** only upon exiting ***class-map (control plane)*** configuration mode.

Example

These commands add the IP ACL ***list_1*** to the ***map_1*** class map, then saves the command by exiting ***class-map*** mode.

```
switch(config)# class-map type copp map_1
switch(config-cmap-map_1)# match ip access-group list_1
switch(config-cmap-map_1)# exit
switch(config)#
```

10.2.7.27 match (class-map (control-plane) Trident II)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4 and IPv4 standard ACLs.

A class map is assigned to a policy map by the `class (policy-map (control-plane) Trident II)` command.

The ***class map*** (control plane) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (control plane) configuration accessed through `class-map type copp` command.

Command Syntax

list_name

list_name

list_name

Parameters

list_name name of ACL assigned to class map.

Related Commands

- class-map type copp places the switch in the ***class-map*** configuration mode.
- **exit** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **abort** discards pending class map changes, then returns the switch to the ***global*** configuration mode.
- class (policy-map (control-plane) Trident II) assigns a class map to a ***policy map***.

Guidelines

Static class maps cannot be modified by this command.

Match statements are saved to ***running-config*** only upon exiting ***class-map (control plane)*** configuration mode.

Example

These commands add the IP ACL ***list_1*** to the ***map_1*** class map, then saves the command by exiting ***class-map*** mode.

```
switch(config)# class-map type copp map_1
switch(config-cmap-map_1)# match ip access-group list_1
switch(config-cmap-map_1)# exit
switch(config)#
```

10.2.7.28 match (class-map (control-plane) Trident)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4, IPv6, IPv4 standard, and IPv6 standard ACLs.

A class map is assigned to a policy map by the `class (policy-map (control-plane) Trident)` command.

Class map (control plane) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (control plane) configuration accessed through `class-map type copp` command

Command Syntax

match IP_VERSION **access-group** ***list_name***

no match IP_VERSION **access-group** ***list_name***

`default match` IP_VERSION `access-group` *list_name*

Parameters

- **IP_VERSION** IP version of the specified ACL. Options include:
  - **ipv4** IPv4.
  - **ipv6** IPv6.
- ***list_name*** name of ACL assigned to class map.

Related Commands

- class-map type copp places the switch in ***class-map*** configuration mode.
- **exit** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **abort** discards pending class map changes, then returns the switch to the ***global*** configuration mode.
- class (policy-map (control-plane) Trident) assigns a class map to a policy map.

Guidelines

Static class maps cannot be modified by this command.

`Match` statements are saved to ***running-config*** only upon exiting ***class-map (control plane)*** configuration mode.

Example

These commands add the IPv4 ACL names ***list_1*** to the ***map_1*** class map, then saves the command by exiting ***class-map*** mode.

```
switch(config)# class-map type copp map_1
switch(config-cmap-map_1)# match ip access-group list_1
switch(config-cmap-map_1)# exit
switch(config)#
```

10.2.7.29 match (class-map (pbr))

The **`match`** command assigns ACLs to the configuration mode Policy-Based Routing (PBR) class map. The command accepts IPv4, IPv4 standard, IPv6 and IPv6 standard ACLs.

Class map (pbr) configuration mode is a group change mode. **`Match`** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **`no match`** and **`default match`** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Note: PBR ACLs use only permit rules to filter data; if there are deny rules in an ACL used by PBR, the configuration will be reverted.

Command Mode

Class-map (pbr) configuration accessed through `class-map type pbr` command.

Command Syntax

[*sequence_number*] `match` [ip | ipv6] `access-group` *list_name*

`no` [*sequence_number*] `match` [ip | ipv6] `access-group` *list_name*

`default` [*sequence_number*] [ip | ipv6] `access-group` *list_name*

`no` [*sequence_number*]

`default` [*sequence_number*]

Parameters

- ***sequence_number*** Sequence number (***1*** to ***4294967295***) assigned to the rule. If no number is entered, the number is derived by adding ***10*** to the number of the class maps last numbered line. To increase the distance between existing entries, use the **`resequence`** command.
- ***list_name*** name of ACL assigned to class map.

Related Commands

- class-map type pbr places the switch in the ***class-map*** configuration mode.
- **`exit`** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to the ***global*** configuration mode.
- class (policy-map (pbr)) assigns a class map to a policy map.

Example

These commands add the IPv4 ACL named ***list1*** to the ***map1*** class map, then save the change by exiting ***class-map*** mode.

```
switch(config)# class-map type pbr map1
switch(config-cmap-map1)# match ip access-group list1
switch(config-cmap-map1)# exit
switch(config)#
```

10.2.7.30 match (class-map (qos) FM6000)

The **`match`** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4 and IPv4 standard ACLs.

The ***class map (qos)*** configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **`no match`** and **`default match`** commands remove the **`match`** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-map (qos) configuration accessed through `class-map type qos` command.

Command Syntax

`match` **IP_VERSION** **`access-group`** ***list_name***

`no match` **IP_VERSION** **`access-group`** ***list_name***

`default match` **IP_VERSION** **`access-group`** ***list_name***

Parameters

- **IP_VERSION** IP version of the specified ACL. Options include:
  - **ipv4** IPv4.
- ***list_name*** name of ACL assigned to class map.

Related Commands

- class-map type qos places the switch in the ***class-map*** configuration mode.
- **`exit`** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to the ***global*** configuration mode.
- class (policy-map (qos) FM6000) assigns a ***class map*** to a ***policy map***.

Example

These commands add the IPv4 ACL named ***list_1*** to the ***map_1*** class map, then saves the command by exiting ***class-map*** mode.

```
switch(config)# class-map type qos map_1
switch(config-cmap-map_1)# match ip access-group list_1
switch(config-cmap-map_1)# exit
switch(config)#
```

10.2.7.31 match (class-map (qos) Helix)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4, IPv4 standard, IPv6, and IPv6 standard ACLs.

the ***class map (QoS)*** configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (QoS) configuration accessed through `class-map type qos` command.

Command Syntax

match IP_VERSION **access-group** ***list_name***

no match IP_VERSION **access-group** ***list_name***

default match IP_VERSION **access-group** ***list_name***

Parameters

- **IP_VERSION** IP version of the specified ACL. Options include:
  - **ipv4** IPv4.
  - **ipv6** IPv6.
- ***list_name*** name of ACL assigned to class map.

Related Commands

- class-map type qos places the switch in the ***class-map*** configuration mode.
- **exit** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **abort** discards pending class map changes, then returns the switch to the ***global*** configuration mode.
- class (policy-map (qos) Helix) assigns a class map to a policy map.

Example

These commands add the IPv4 ACL named ***list_1*** to the ***map_1*** class map, then saves the command by exiting ***class-map*** mode.

```
switch(config)# class-map type qos map_1
switch(config-cmap-map_1)# match ip access-group list_1
switch(config-cmap-map_1)# exit
switch(config)#
```

10.2.7.32 match (class-map (qos) Trident II)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4, IPv4 standard, IPv6, and IPv6 standard ACLs.

The ***class map (QoS)*** configuration mode is a group change mode. **`Match`** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **`no match`** and **`default match`** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

The ***class-map (qos)*** configuration accessed through `class-map type qos` command.

Command Syntax

IP_VERSION *list_name*

IP_VERSION *list_name*

IP_VERSION *list_name*

Parameters

- **IP_VERSION** IP version of the specified ACL. Options include:
  - **ipv4** IPv4.
  - **ipv6** IPv6.
- ***list_name*** name of ACL assigned to class map.

Related Commands

- class-map type qos places the switch in the ***class-map*** configuration mode.
- **`exit`** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- `abort` discards pending class map changes, then returns the switch to the ***global*** configuration mode.
- class (policy-map (qos) Trident) assigns a class map to a policy map.

Example

These commands add the IPv4 ACL named ***list_1*** to the ***map_1*** class map, then saves the command by exiting ***class-map*** mode.

```
switch(config)# class-map type qos map_1
switch(config-cmap-map_1)# match ip access-group list_1
switch(config-cmap-map_1)# exit
switch(config)#
```

10.2.7.33 match (class-map (qos) Trident)

The **`match`** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4, IPv4 standard, IPv6, and IPv6 standard ACLs.

Class map (QoS) configuration mode is a group change mode. **`Match`** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **`no match`** and **`default match`** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (qos) configuration accessed through `class-map type qos` command.

Command Syntax

`match` **IP_VERSION** **`access-group`** ***list_name***

`no match` **IP_VERSION** **`access-group`** ***list_name***

`default match` **IP_VERSION** **`access-group`** ***list_name***

Parameters

- **IP_VERSION** IP version of the specified ACL. Options include:
  - **ipv4** IPv4.
  - **ipv6** IPv6.
- ***list_name*** name of ACL assigned to class map.

Related Commands

- class-map type qos places the switch in the ***class-map*** configuration mode.
- **`exit`** saves pending class map changes, then returns the switch to the ***global*** configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to the ***global*** configuration mode.
- class (policy-map (qos) Trident) assigns a ***class map*** to a ***policy map***.

Example

These commands add the IPv4 ACL named ***list_1*** to the ***map_1*** class map, then saves the command by exiting ***class-map*** mode.

```
switch(config)# class-map type qos map_1
switch(config-cmap-map_1)# match ip access-group list_1
switch(config-cmap-map_1)# exit
switch(config)#
```

10.2.7.34 match (policy-map (pbr))

The **`match`** command creates a policy map clause entry that specifies one filtering condition. When a packet matches the filtering criteria, its next hop is set as specified. When a packets properties do not equal the statement parameters, the packet is evaluated against the next clause or class map in the policy map, as determined by sequence number. If all clauses fail to set a next hop for the packet, the packet is routed according to the FIB.

The **`no match`** and **`default match`** commands remove the **match** statement from the configuration mode policy map by deleting the corresponding command from ***running-config***.

Command Mode

Policy-Map (pbr) Configuration accessed through `policy-map type pbr` command.

Command Syntax

[***sequence_number***] **`match ip`** **SOURCE_ADDR DEST_ADDR** [**`set nexthop [recursive]`** ***NH-addr_1*** [***NH-addr_2***] ***...*** [***NH-addr_n***]]

`no match ip` **SOURCE_ADDR DEST_ADDR** [**`set nexthop [recursive]`** ***NH-addr_1*** [***NH-addr_2***] ***...*** [***NH-addr_n***]]

`default match match ip` **SOURCE_ADDR DEST_ADDR** [**`set nexthop [recursive]`** ***NH-addr_1*** [***NH-addr_2***] ***...*** [***NH-addr_n***]]

`no` **SEQ_NUM**

`default` **SEQ_NUM**

Parameters

- ***sequence_number*** Sequence number assigned to the rule. If no number is entered, the number is derived by adding ***10*** to the number of the policy maps last numbered line. To increase the distance between existing entries, use the **`resequence`** command.
- **SOURCE_ADDR** and **DEST_ADDR** source and destination address filters. Options include:
  - ***network_addr*** subnet address (CIDR or address-mask).
  - **any** packets from or to all addresses are matched.

- **host *ip_addr*** IP address (dotted decimal notation).

  Source and destination subnet addresses support discontiguous masks.
- **recursive** enables recursive next hop resolution.
- ***NH_addr*** IP address of next hop. If multiple addresses are entered, they are treated as an ECMP group.

Related Commands

- policy-map type pbr enters the policy-map (PBR) configuration mode.
- show policy-map type pbr displays the PBR policy maps.

Example

These commands create a match rule in policy map ***PMAP1*** which sets the next hop to ***192.168.3.5*** for packets received from ***172.16.0.0/12*** regardless of their destination, then exit the mode to save the changes.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)# match ip 172.16.0.0/12 any set nexthop 192.163.3.5
switch(config-pmap-PMAP1)# exit
switch(config)#
```

10.2.7.35 platform arad tcam counters feature

The **`platform arad tcam counters feature`** command enables incrementing PBR hardware counters corresponding to ACL. If counters for PBR are enabled, then counters for ACL will be automatically disabled in all cases. If counters for ACL are enabled, then counters for PBR will be automatically disabled in all cases.

The **`no platform arad tcam counters feature`** command disables PBR/ACL counters selection. The **`default platform arad tcam counters feature`** commands resets the default behavior.

Command Mode

Global Configuration

Command Syntax

`platform arad tcam counters feature` [***OPTIONS***]

`no platform arad tcam counters feature` [***OPTIONS***]

`default platform arad tcam counters feature` [***OPTIONS***]

Parameters

OPTIONS Assign the TCAM counters feature. Options include:

- **pbr** assign the TCAM counters feature PBR hardware counters.
- **acl** assign the TCAM counters feature ACL hardware counters.

Example

- This command enables incrementing ACL hardware counters selection.

```
switch(config)# platform arad tcam counters feature acl
switch(config)#
```

- This command disables incrementing ACL hardware counters selection.

```
switch(config)# no platform arad tcam counters feature acl
switch(config)#
```

10.2.7.36 policy-map type copp

The **policy-map type copp** command places the switch in the ***policy-map*** (control plane) configuration mode, which is a group change mode that modifies a ***control-plane*** policy map. A policy map is a data structure that consists of class maps that identify a specific data stream and specify bandwidth and shaping parameters that controls its transmission. Control plane policy maps are applied to the control plane to manage traffic.

The ***copp-system-policy*** policy map is supplied with the switch and is always applied to the control plane. The ***copp-system-policy*** is the only valid control plane policy map.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to the ***global*** configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no policy-map type copp** and **default policy-map type copp** commands delete the specified policy map by removing the corresponding **policy-map type copp** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

policy-map type copp copp-system-policy

no policy-map type copp copp-system-policy

default policy-map type copp copp-system-policy

The ***copp-system-policy*** is supplied with the switch and is the only valid control plane policy map.

Commands Available in Policy-Map Configuration Mode

- class (policy-map (control-plane) FM6000)
- class (policy-map (control-plane) Trident)

Related Commands

class-map type copp enters the ***control-plane class-map*** configuration mode for modifying a control-plane dynamic class map.

Only Helix and Trident platform switches support dynamic classes for control plane policing.

Example

This command places the switch in the ***policy-map*** configuration mode to edit the ***copp-system-policy*** policy map.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)#
```

10.2.7.37 policy-map type pbr

The **policy-map type pbr** command places the switch in ***policy-map (pbr)*** configuration mode, which is a group change mode that modifies a Policy-Based Routing (PBR) policy map. The command also creates the specified policy map if it does not already exist. A PBR policy map is a data structure that consists of class maps that identify specific packets and the next hops for those packets. Policy maps are applied to Ethernet or port channel interfaces to manage traffic.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to the ***global*** configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no policy-map type pbr** and **default policy-map type pbr** commands delete the specified policy map by removing the corresponding **policy-map type pbr** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

policy-map type pbr ***map_name***

no policy-map type pbr ***map_name***

default policy-map type pbr ***map_name***

Parameters

map_name Name of policy map.

Commands Available in Policy-Map Configuration Mode

- class (policy-map (pbr))
- match (policy-map (pbr))

Related Commands

- class-map type pbr
- service-policy type pbr (Interface mode)

Example

This command creates the PBR policy map named ***PMAP1*** and places the switch in ***policy-map*** configuration mode.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#
```

10.2.7.38 policy-map type quality-of-service

The **policy-map type quality-of-service** command places the switch in the ***policy-map (QoS)*** configuration mode, which is a group change mode that modifies a QoS policy map. A policy map is a data structure that consists of class maps that identify a specific data stream and shaping parameters that controls its transmission. Policy maps are applied to Ethernet or port channel interfaces to manage traffic.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to the ***global*** configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to the ***global*** configuration mode.

The **no policy-map type quality-of-service** and **default policy-map type quality-of-service** commands delete the specified policy map by removing the corresponding **policy-map type quality-of-service** command and its associated configuration. The **policy-map** and **policy-map type quality-of-service** commands are equivalent.

Command Mode

Global Configuration

Command Syntax

policy-map type quality-of-service ***map_name***

no policy-map type quality-of-service ***map_name***

default policy-map type quality-of-service ***map_name***

Parameters

map_name Name of policy map.

Commands Available in Policy-Map Configuration Mode

- class (policy-map (qos) FM6000)
- class (policy-map (qos) Trident)

Conditions

`policy-map` ***map_name*** and **`policy-map type quality-of-service`** ***map_name*** are identical commands.

Related Commands

- class-map type qos
- service-policy type qos (Interface mode)

Example

This command creates the QoS policy map named ***PMAP-1*** and places the switch in the ***policy-map*** configuration mode.

```
switch(config)# policy-map PMAP-1
switch(config-pmap-PMAP-1)#
```

10.2.7.39 resequence (class-map (pbr))

The **`resequence`** command assigns sequence numbers to rules in the configuration mode class map. Command parameters specify the number of the first rule and the numeric interval between consecutive rules. Once changed, rule numbers persist unless changed again using the **`resequence`** command, but the interval used for numbering new rules reverts to ***10*** on the exiting ***class-map (pbr)*** configuration mode.

Maximum rule sequence number is ***4294967295***.

Command Mode

Class-Map (PBR) Configuration accessed through `class-map type pbr` command.

Command Syntax

`resequence` [***start_num*** [***inc_num***]]

Parameters

- ***start_num*** sequence number assigned to the first rule. Default is ***10***.
- ***inc_num*** numeric interval between consecutive rules. Default is ***10***.

Example

The **`resequence`** command renumbers the rules in ***CMAP1***, starting the first command at number ***100*** and incrementing subsequent lines by ***20***.

```
switch(config)# class-map type pbr match-any CMAP1
switch(config-cmap-CMAP1)# show active
class-map type pbr match-any CMAP1
10 match ip access-group group1
20 match ip access-group group2
30 match ip access-group group3
switch(config-cmap-CMAP1)# resequence 100 20
switch(config-cmap-CMAP1)# exit
switch(config)# class-map type pbr match-any CMAP1
switch(config-cmap-CMAP1)# show active
class-map type pbr match-any CMAP1
100 match ip access-group group1
120 match ip access-group group2
```

```
140 match ip access-group group3
```

10.2.7.40 resequence (policy-map (pbr))

The **resequence** command assigns sequence numbers to rules in the configuration mode policy map. Command parameters specify the number of the first rule and the numeric interval between consecutive rules. Once changed, rule numbers persist unless changed again using the **resequence** command, but the interval used for numbering new rules reverts to ***10*** on the exiting ***policy-map (pbr)*** configuration mode.

Maximum rule sequence number is ***4294967295***.

Command Mode

Policy-Map (PBR) Configuration accessed through `policy-map type pbr` command

Command Syntax

resequence [***start_num*** [***inc_num***]]

Parameters

- ***start_num*** sequence number assigned to the first rule. Default is ***10***.
- ***inc_num*** numeric interval between consecutive rules. Default is ***10***.

Example

The **resequence** command renumbers the rules in ***PMAP1***, starting the first command at number ***100*** and incrementing subsequent lines by ***20***.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)# show active
policy-map type pbr PMAP1
10 class CMAP1
set nexthop 172.16.1.1
20 class CMAP2
set nexthop 172.16.2.2
30 class CMAP3
set nexthop 172.16.3.3
switch(config-pmap-PMAP1)# resequence 100 20
switch(config-pmap-PMAP1)# exit
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)# show active
class-map type pbr PMAP1
100 class CMAP1
set nexthop 172.16.1.1
120 class CMAP2
set nexthop 172.16.2.2
140 class CMAP3
set nexthop 172.16.3.3
switch(config-pmap-PMAP1)#
```

10.2.7.41 service-policy type pbr (Interface mode)

The **service-policy pbr** command applies the specified Policy-Based Routing (PBR) policy map to the configuration mode interface. A PBR policy map is a data structure that consists of class maps that identify specific packets and the next hops for those packets. Policy maps are applied to Ethernet or port channel interfaces to manage traffic. Only one service policy is supported per interface.

The **no service-policy pbr** and **default service-policy pbr** commands remove the service policy assignment from the configuration mode interface by deleting the corresponding **service-policy pbr** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`service-policy type pbr` **TRAFFIC_DIRECTION** ***map_name***

`no service-policy type pbr` **TRAFFIC_DIRECTION** ***map_name***

`default service-policy type pbr` **TRAFFIC_DIRECTION** ***map_name***

Parameters

- **TRAFFIC_DIRECTION** IP address or peer group name. Values include:
  - **input** Policy map applies to inbound packet streams.
- ***map_name*** Name of policy map.

Guidelines

A policy map that is attached to a port channel interface takes precedence for member interfaces of the port channel over their individual interface Ethernet configuration. Members that are removed from a port channel revert to the policy map implementation specified by its interface Ethernet configuration.

Related Commands

policy-map type pbr

Example

This command applies the PBR policy map ***PMAP1*** to ***interface Ethernet 8***.

```
switch# config
switch(config)# interface ethernet 8
switch(config-if-Et8)# service-policy type pbr input PMAP1
switch(config-if-Et8)#
```

10.2.7.42 service-policy type qos (Interface mode)

The **`service-policy`** command applies a specified policy map to the configuration mode interface. A policy map is a data structure that identifies data traffic through class maps, then specifies actions to classify the traffic (by setting the traffic class), mark the traffic (by setting the cos and dscp values), and police the traffic (by setting the police rate) through data packet field modifications.

The **`no service-policy`** and **`default service-policy`** commands remove the service policy assignment from the configuration mode interface by deleting the corresponding **`service-policy`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`service-policy [type qos]` **TRAFFIC_DIRECTION** ***map_name***

`no service-policy [type qos]` **TRAFFIC_DIRECTION** ***map_name***

`default service-policy [type qos]` **TRAFFIC_DIRECTION** ***map_name***

Parameters

- **`type qos`** Parameter has no functional effect.
- **TRAFFIC_DIRECTION** Direction of data stream to which command applies. Options include:

- **input** Policy map applies to inbound packet streams.
- ***map_name*** Name of policy map.

Guidelines

A policy map that is attached to a port channel interface takes precedence for member interfaces of the port channel over their individual interface Ethernet configuration. Members that are removed from a port channel revert to the policy map implementation specified by its interface Ethernet configuration.

DCS-7500E and DCS-7280E limitations:

- A maximum of ***31*** QoS service policies per chip may be applied on L3 interfaces.
- Applying different QoS service policies to an SVI and its member interfaces causes unpredictable behavior.
- When an SVI on which QoS service policies are applied experiences partial failure due to limited hardware resources, a forwarding agent restart causes unpredictable behavior.
- Policy-map programming may fail when QoS service policies are applied on two SVIs if an event causes a member interface to switch membership from one to the other. To change the VLAN membership of an interface in this case, remove the interface from one VLAN before adding it to the other.
- Outgoing COS rewrite is not supported.
- QoS policy-map counters are not supported.

DCS-7010, DCS-7050, DCS-7050X, DCS-7250X, and DCS-7300X limitations:

- When the same policy map is applied to multiple SVIs, TCAM resources are not shared.
- A policy map applied to an SVI results in TCAM allocation on all chips whether SVI members are present or not.
- Applying different QoS service policies to an SVI and its member interfaces causes unpredictable behavior.

Related Commands

policy-map type quality-of-service

Example

This command applies the ***PMAP-1*** policy map to ***interface ethernet 8***.

```
switch# config
switch(config)# interface ethernet 8
switch(config-if-Et8)# show active
switch(config-if-Et8)# service-policy input PMAP-1
switch(config-if-Et8)# show active
interface Ethernet8
   service-policy type qos input PMAP-1
switch(config-if-Et8)#
```

10.2.7.43 set (policy-map-class (qos) FM6000)

The **set** command specifies traffic resolution methods for traffic defined by its associated class map in its configuration mode policy map class. Three set statements are available for each class:

- **cos** Sets the Layer 2 class of service field.
- **dscp** Sets the differentiated services code point value in the type of service (ToS) byte.
- **traffic-class** Sets the traffic class queue for data packets.

Each type of set command can be assigned to a class, allowing for the simultaneous modification of both (cos, dscp) fields and assignment to a traffic class.

The **no set** and **default set** commands remove the specified data action from the class map by deleting the associated **set** command from ***running-config***.

Command Mode

Policy-map-class (qos) configuration

accessed through `class (policy-map (qos) FM6000)` command.

Command Syntax

`set` **QOS_TYPE** *value*

`no set` **QOS_TYPE**

`default set` **QOS_TYPE**

Parameters

- **QOS_TYPE** Specifies the data stream resolution method. Valid options include:
  - **cos** Layer 2 class of service field of outbound packet is modified.
  - **dscp** Differentiated services code point value in the ToS byte is modified.
  - **traffic-class** Data stream is assigned to a traffic class queue.
- ***value*** Specifies the data field value or traffic class queue. Valid data range depends on **QOS_TYPE**.
  - **QOS_TYPE** is **cos** Value ranges from ***0*** to ***7***.
  - **QOS_TYPE** is **dscp** Value ranges from ***0*** to ***63***.
  - **QOS_TYPE** is **traffic-class** Value ranges from ***0*** to ***7***.

Related Commands

- policy-map type quality-of-service
- class (policy-map (qos) FM6000)

Example

These commands configure the policy map to set ***CoS field 7*** to data traffic specified by the class map ***CMAP-1***, then assigns that data to traffic class ***queue 4***.

```
switch(config)# policy-map type quality-of-service PMAP-1
switch(config-pmap-PMAP-1)# class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)# set cos 7
switch(config-pmap-c-PMAP-1-CMAP-1)# set traffic-class 4
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

10.2.7.44 set (policy-map-class (qos) Helix)

The **`set`** command specifies traffic resolution methods for traffic defined by its associated class map in its configuration mode policy map class. Three set statements are available for each class:

- **cos** Sets the Layer 2 class of service field.
- **dscp** Sets the differentiated services code point value in the type of service (ToS) byte.
- **traffic-class** Sets the traffic class queue for data packets.

Each type of set command can be assigned to a class, allowing for the simultaneous modification of both (**cos**, **dscp**) fields and assignment to a traffic class.

The **`no set`** and **`default set`** commands remove the specified data action from the class map by deleting the associated **`set`** command from ***running-config***.

Command Mode

Policy-map-class (qos) configuration accessed through `class (policy-map (qos) Helix)` command.

Command Syntax

`set` **QOS_TYPE** ***value***

`no set` QOS_TYPE

`default set` QOS_TYPE

Parameters

- **QOS_TYPE** Specifies the data stream resolution method. Valid options include:
  - **cos** Layer 2 class of service field of outbound packet is modified.
  - **dscp** Differentiated services code point value in the ToS byte is modified.
  - **traffic-class** Data stream is assigned to a traffic class queue.
- ***value*** Specifies the data field value or traffic class queue. Valid data range depends on QOS type.
  - **QOS_TYPE** is **cos** Value ranges from ***0*** to ***7***.
  - **QOS_TYPE** is **dscp** Value ranges from ***0*** to ***63***.
  - **QOS_TYPE** is **traffic-class** Value ranges from ***0*** to ***7***.

Related Commands

- policy-map type quality-of-service
- class (policy-map (qos) Helix)

Example

These commands configure the policy map to set ***CoS field 7*** to data traffic specified by the class map ***CMAP-1***, then assigns that data to ***traffic class queue 4***.

```
switch(config)# policy-map type quality-of-service PMAP-1
switch(config-pmap-PMAP-1)# class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)# set cos 7
switch(config-pmap-c-PMAP-1-CMAP-1)# set traffic-class 4
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

10.2.7.45 set (policy-map-class (qos) Trident II)

The **`set`** command specifies traffic resolution methods for traffic defined by its associated class map in its configuration mode policy map class. Three set statements are available for each class:

- **cos** Sets the Layer 2 class of service field.
- **dscp** Sets the differentiated services code point value in the type of service (ToS) byte.
- **traffic-class** Sets the traffic class queue for data packets.

Each type of set command can be assigned to a class, allowing for the simultaneous modification of both (cos, dscp) fields and assignment to a traffic class.

The **`no set`** and **`default set`** commands remove the specified data action from the class map by deleting the associated **`set`** command from ***running-config***.

Command Mode

Policy-map-class (qos) configuration accessed through `class (policy-map (qos) Trident)` command.

Command Syntax

`set` QOS_TYPE *value*

`no set` QOS_TYPE

`default set` QOS_TYPE

Parameters

- **QOS_TYPE** Specifies the data stream resolution method. Valid options include:
  - **cos** Layer 2 class of service field of outbound packet is modified.

  - **dscp** Differentiated services code point value in the ToS byte is modified.
  - **traffic-class** Data stream is assigned to a traffic class queue.
- ***value*** Specifies the data field value or traffic class queue. Valid data range depends on QOS type.
  - **QOS_TYPE** is **cos** Value ranges from ***0*** to ***7***.
  - **QOS_TYPE** is **dscp** Value ranges from ***0*** to ***63***.
  - **QOS_TYPE** is **traffic-class** Value ranges from ***0*** to ***7***.

Related Commands

- policy-map type quality-of-service
- class (policy-map (qos) Trident)

Example

These commands configure the policy map to set ***CoS field 7*** to data traffic specified by the class map ***CMAP-1***, then assigns that data to ***traffic class queue 4***.

```
switch(config)# policy-map type quality-of-service PMAP-1
switch(config-pmap-PMAP-1)# class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)# set cos 7
switch(config-pmap-c-PMAP-1-CMAP-1)# set traffic-class 4
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

10.2.7.46 set (policy-map-class (qos) Trident)

The **set** command specifies traffic resolution methods for traffic defined by its associated class map in its configuration mode policy map class. Three set statements are available for each class:

- **cos** Sets the Layer 2 class of service field.
- **dscp** Sets the differentiated services code point value in the type of service (ToS) byte.
- **traffic-class** Sets the traffic class queue for data packets.

Each type of set command can be assigned to a class, allowing for the simultaneous modification of both (cos, dscp) fields and assignment to a traffic class.

The **no set** and **default set** commands remove the specified data action from the class map by deleting the associated **set** command from ***running-config***.

Command Mode

Policy-map-class (qos) configuration accessed through `class (policy-map (qos) Trident)` command.

Command Syntax

set QOS_TYPE ***value***

no set QOS_TYPE

default set QOS_TYPE

Parameters

- **QOS_TYPE** Specifies the data stream resolution method. Valid options include:
  - **cos** Layer 2 class of service field of outbound packet is modified.
  - **dscp** Differentiated services code point value in the ToS byte is modified.
  - **traffic-class** Data stream is assigned to a traffic class queue.
- ***value*** Specifies the data field value or traffic class queue. Valid data range depends on QOS type.
  - **QOS_TYPE** is **cos** Value ranges from ***0*** to ***7***.
  - **QOS_TYPE** is **dscp** Value ranges from ***0*** to ***63***.

- **QOS_TYPE** is **traffic-class** Value ranges from ***0*** to ***7***.

Related Commands

- policy-map type quality-of-service
- class (policy-map (qos) Trident)

Example

These commands configure the policy map to set ***CoS field 7*** to data traffic specified by the ***class map CMAP-1***, then assigns that data to ***traffic class queue 4***.

```
switch(config)# policy-map type quality-of-service PMAP-1
switch(config-pmap-PMAP-1)# class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)# set cos 7
switch(config-pmap-c-PMAP-1-CMAP-1)# set traffic-class 4
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

10.2.7.47 set nexthop (policy-map-class pbr)

The **`set nexthop`** command specifies the next hop for traffic defined by its associated class map in its configuration mode policy map class.

The **`no set nexthop`** and **`default set nexthop`** commands remove the specified action from the class map by deleting the associated **`set nexthop`** command from ***running-config***.

Command Mode

Policy-map-class (pbr) configuration accessed through `class (policy-map (pbr))` command.

Command Syntax

`set nexthop` [recursive] ***NH-addr_1*** [***NH-addr_2***] ***...*** [***NH-addr_n***]

`no set nexthop` [recursive]

`default set nexthop` [recursive]

Parameters

- **recursive** enables recursive next hop resolution.
- ***NH_addr*** IP address of next hop. If multiple addresses are entered, they are treated as an ECMP group.

Related Commands

- policy-map type pbr
- class (policy-map (pbr))

Example

These ***192.168.5.3*** commands configure the policy map ***PMAP1*** to set the next hop to for traffic defined by class map ***CMAP1***.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)# class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)# set nexthop 192.168.5.3
switch(config-pmap-c-PMAP1-CMAP1)#
```

10.2.7.48 set nexthop-group (policy-map-class(pbr) Arad)

The **`set nexthop-group`** command specifies a nexthop group as the next hop for traffic defined by its associated class map in its configuration mode policy map class.

The **no set nexthop-group** and **default set nexthop-group** commands remove the specified action from the class map by deleting the associated **set nexthop-group** command from ***running-config***.

Command Mode

Policy-map-class (pbr) configuration accessed through `class (policy-map (pbr))` command.

Command Syntax

set nexthop-group ***group_name***

no set nexthop-group ***group_name***

default set nexthop-group ***group_name***

Parameters

group_name name of ECMP group to use as next hop.

Related Commands

- policy-map type pbr
- class (policy-map (pbr))

Example

These commands configure the policy map ***PMAP1*** to set the next hop to a nexthop group named ***GROUP1*** for traffic defined by class map ***CMAP1***.

```
switch(config)# policy-map type pbr PMAP1
switch(config-pmap-PMAP1)# class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)# set nexthop-group GROUP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

10.2.7.49 shape (policy-map-class (control-plane) Arad)

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration accessed through `class (policy-map (control-plane) Arad)`

Command Syntax

Parameters

kilobits Maximum data rate in kilobits per second. Value ranges from 1 to 10000000.

Related Commands

- class (policy-map (control-plane) Arad) places the switch in the ***policy-map-class (control plane)*** configuration mode.
- bandwidth (policy-map-class (control-plane) Arad) specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Static Classes Default Shape

Arad platform switches define these default shapes for static classes:

- copp-system-bgp 2500 copp-system-l3lpmoverflow 2500

- copp-system-bpdu 2500 copp-system-l3slowpath 2500
- copp-system-default 2500 copp-system-l3ttl1 2500
- copp-system-ipbroadcast 2500 copp-system-lacp 2500
- copp-system-ipmc 2500 copp-system-linklocal 2500
- copp-system-ipmcmiss 2500 copp-system-lldp 2500
- copp-system-ipunicast NO LIMIT copp-system-mlag 2500
- copp-system-l2broadcast 2500 copp-system-multicastsnoop 2500
- copp-system-l2unicast NO LIMIT copp-system-OspfIsis 2500
- copp-system-l3destmiss 2500 copp-system-sflow 2500

Example

These commands configure the maximum bandwidth of ***2000*** kbps for data traffic specified by the class map ***copp-system-lldp*** of the default ***control-plane policy map***.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# shape kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# exit
switch(config-pmap-copp-system-policy)# exit
switch(config)# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy

  Class-map: copp-system-lldp (match-any)
       shape : 2000 kbps
       bandwidth : 250 kbps
      Out Packets : 0
      Drop Packets : 0

switch(config)#
```

10.2.7.50 shape (policy-map-class (control-plane) FM6000)

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration accessed through class (policy-map (control-plane) FM6000).

Command Syntax

shape pps ***packets***

no shape

default shape

Parameters

packets Maximum data rate in packets per second. Value ranges from 1 to 100000.

Related Commands

- class (policy-map (control-plane) FM6000) places the switch in the ***policy-map-class (control plane)*** configuration mode.
- bandwidth (policy-map-class (control-plane) FM6000) specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Static Classes Default Shape

FM6000 platform switches define these default shapes for static classes:

- copp-system-arp 10000 copp-system-l3slowpath 10000
- copp-system-default 8000 copp-system-pim-ptp 10000
- copp-system-ipmcrsvd 10000 copp-system-ospf-isis 10000
- copp-system-ipmcmiss 10000 copp-system-selfip 5000
- copp-system-igmp 10000 copp-system-selfip-tc6to7 5000
- copp-system-l2rsvd 10000 copp-system-sflow 25000

Example

These commands configure a maximum bandwidth of ***5000*** packets per second for data traffic specified by the class map ***PMAP-1*** in the policy map named ***copp-system-policy***.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class PMAP-1
switch(config-pmap-c-copp-system-policy-PMAP-1)# shape pps 5000
switch(config-pmap-c-copp-system-policy-PMAP-1)#
```

10.2.7.51 shape (policy-map-class (control-plane) Helix)

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration accessed through class (policy-map (control-plane) Helix).

Command Syntax

shape pps ***packets***

no shape

default shape

Parameters

packets Maximum data rate in packets per second. Value ranges from ***1*** to ***100000***.

Static Classes Default Shape

Trident platform switches define these default shapes for static classes:

- copp-system-acllog 10000 copp-system-l3ttl1 10000
- copp-system-arp 10000 copp-system-lacp 5000
- copp-system-arpresolver 10000 copp-system-lldp 10000
- copp-system-bfd 10000 copp-system-mlag 5000
- copp-system-bgp 5000 copp-system-OspfIsis 10000
- copp-system-bpdu 5000 copp-system-selfip 5000
- copp-system-default 8000 copp-system-selfip-tc6to7 5000
- copp-system-glean 10000 copp-system-sflow 25024
- copp-system-igmp 10000 copp-system-tc3to5 10000
- copp-system-ipmcmiss 10000 copp-system-tc6to7 10000
- copp-system-ipmcrsvd 10000 copp-system-urm 10000
- copp-system-l3destmiss 10000 copp-system-vrrp 5000

- copp-system-l3slowpath 10000

Related Commands

- class (policy-map (control-plane) Helix) places the switch in the ***policy-map-class (control plane)*** configuration mode.
- bandwidth (policy-map-class (control-plane) Helix) specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Example

These commands configure a maximum bandwidth of ***5000*** packets per second for data traffic specified by the ***copp-system-lldp*** of the default control-plane policy map.

```
switch(config)# policy-map type control-plan copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# shape pps 5000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# exit
switch(config-pmap-copp-system-policy)# exit
switch(config)# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy

  Class-map: copp-system-lldp (match-any)
       shape : 5000 pps
       bandwidth : 500 pps
      Out Packets : 305961
      Drop Packets : 0

switch(config)#
```

10.2.7.52 shape (policy-map-class (control-plane) Petra)

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration accessed through class (policy-map (control-plane) Petra)

Command Syntax

shape kbps ***kilobits***

no shape

default shape

Parameters

kilobits Maximum data rate in kilobits per second. Value ranges from ***1*** to ***10000000***.

Related Commands

- class (policy-map (control-plane) Petra) places the switch in policy-map-class (control plane) configuration mode.
- bandwidth (policy-map-class (control-plane) Petra) specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Static Classes Default Shape

Petra platform switches define these default shapes for static classes:

- copp-system-bpdu 2500 copp-system-l3destmiss 2500
- copp-system-default 2500 copp-system-l3slowpath 2500
- copp-system-igmp 2500 copp-system-l3ttl0 2500
- copp-system-ipbroadcast 2500 copp-system-l3ttl1 2500
- copp-system-ipmc 2500 copp-system-lacp 2500
- copp-system-ipmcmiss 2500 copp-system-lldp 2500
- copp-system-ipmcrsvd 2500 copp-system-unicast-arp 2500
- copp-system-ipunicast No Limit

Guidelines

Petra does not support all discrete rate values. When a specified discrete value is not supported, the switch converts the rate to the next highest discrete value that it supports. The **show** command displays the converted rate and not the user-configured rate.

Example

These commands configure the maximum bandwidth of ***2000*** kbps for data traffic specified by the class map ***copp-system-lldp*** of the ***default control-plane*** policy map. Because the switch does not support the discrete value of ***2000*** kbps, it converts the bandwidth up to ***2115*** kbps.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# shape kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# exit
switch(config-pmap-copp-system-policy)# exit
switch(config)# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy

  Class-map: copp-system-lldp (match-any)
       shape : 2115 kbps
       bandwidth : 325 kbps
      Out Packets : 0
      Drop Packets : 0

switch(config)#
```

10.2.7.53 shape (policy-map-class (control-plane) Trident II)

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration accessed through `class (policy-map (control-plane) Trident II)`.

Command Syntax

shape pps ***packets***

no shape

default shape

Parameters

packets Maximum data rate in packets per second. Value ranges from ***1*** to ***100000***.

Static Classes Default Shape

Trident II platform switches define these default shapes for static classes:

- copp-system-acllog 10000 copp-system-l3slowpath 10000
- copp-system-arp 10000 copp-system-l3ttl1 10000
- copp-system-arpresolver 10000 copp-system-lacp 5000
- copp-system-bfd 10000 copp-system-lldp 10000
- copp-system-bgp 5000 copp-system-mlag 5000
- copp-system-bpdu 5000 copp-system-selfip 5000
- copp-system-default 8000 copp-system-selfip-tc6to7 5000
- copp-system-glean 10000 copp-system-sflow 25024
- copp-system-igmp 10000 copp-system-tc3to5 10000
- copp-system-ipmcmiss 10000 copp-system-tc6to7 10000
- copp-system-ipmcrsvd 10000 copp-system-urm 10000

Related Commands

- class (policy-map (control-plane) Trident II) places the switch in ***policy-map-class (control plane)*** configuration mode.
- bandwidth (policy-map-class (control-plane) Trident II) specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Example

These commands configure a maximum bandwidth of ***5000*** packets per second for data traffic specified by the ***copp-system-lldp*** of the ***default control-plane policy*** map.

```
switch(config)# policy-map type control-plan copp-system-policy
switch(config-pmap-copp-system-policy)# class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# shape pps 5000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)# exit
switch(config-pmap-copp-system-policy)# exit
switch(config)# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy

  Class-map: copp-system-lldp (match-any)
       shape : 5000 pps
       bandwidth : 500 pps
      Out Packets : 305961
      Drop Packets : 0

switch(config)#
```

10.2.7.54 shape (policy-map-class (control-plane) Trident)

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration accessed through class (policy-map (control-plane) Trident).

Command Syntax

shape pps ***packets***

no shape

default shape

Parameters

packets Maximum data rate in packets per second. Value ranges from 1 to 100000.

Static Classes Default Shape

Trident platform switches define these default shapes for static classes:

- copp-system-arp 10000 copp-system-lldp 10000
- copp-system-arpresolver 10000 copp-system-l3destmiss 10000
- copp-system-bpdu 5000 copp-system-l3slowpath 10000
- copp-system-default 8000 copp-system-l3ttl1 10000
- copp-system-glean 10000 copp-system-selfip 5000
- copp-system-igmp 10000 copp-system-selfip-tc6to7 5000
- copp-system-ipmcmiss 10000 copp-system-sflow 25000
- copp-system-ipmcrsvd 10000 copp-system-tc3to5 10000
- copp-system-lacp 5000 copp-system-tc6to7 10000

Related Commands

- class (policy-map (control-plane) Trident) places the switch in the ***policy-map-class (control plane)*** configuration mode.
- bandwidth (policy-map-class (control-plane) Trident) specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Example

These commands configure a maximum bandwidth of ***5000*** packets per second for data traffic specified by the class map ***PMAP-1*** in the policy map named ***copp-system-policy***.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-copp-system-policy)# class PMAP-1
switch(config-pmap-c-copp-system-policy-PMAP-1)# shape pps 5000
switch(config-pmap-c-copp-system-policy-PMAP-1)
```

10.2.7.55 show class-map type control-plane

The **`show class-map`** command displays contents of available control-plane class maps. **Control-plane** class maps can be added to the ***copp-system-policy*** policy map. ***Control-plane*** class maps can be static class maps defined by the system or dynamic maps created in ***class-map*** configuration mode.

Dynamic class maps are composed of statements that match IPv4 access control lists. Static class maps are defined by the switch and cannot be altered.

Command Mode

EXEC

Command Syntax

`show class-map type control-plane` [**MAP_NAME**]

Parameters

MAP_NAME Name of class map displayed by the command. Options include:

- ***no parameter*** Command displays all control plane class maps.
- ***name_text*** Command displays specified control-plane class maps.

Related Command

- **`show class-map`** command displays QoS class maps.
- show class-map type qos displays control plane class maps.

Example

This command displays the available control plane class maps.

```
switch# show class-map type control-plane
  Class-map: CM-CP1 (match-any)
    Match: ip access-group name LIST-CP1
  Class-map: copp-system-acllog (match-any)
  Class-map: copp-system-arp (match-any)
  Class-map: copp-system-arpresolver (match-any)
  Class-map: copp-system-bpdu (match-any)
  Class-map: copp-system-glean (match-any)
  Class-map: copp-system-igmp (match-any)
  Class-map: copp-system-ipmcmiss (match-any)
  Class-map: copp-system-ipmcrsvd (match-any)
  Class-map: copp-system-l3destmiss (match-any)
  Class-map: copp-system-l3slowpath (match-any)
  Class-map: copp-system-l3ttl1 (match-any)
  Class-map: copp-system-lacp (match-any)
  Class-map: copp-system-lldp (match-any)
  Class-map: copp-system-selfip (match-any)
  Class-map: copp-system-selfip-tc6to7 (match-any)
  Class-map: copp-system-sflow (match-any)
  Class-map: copp-system-tc3to5 (match-any)
  Class-map: copp-system-tc6to7 (match-any)
switch>
```

10.2.7.56 show class-map type pbr

The **`show class-map`** command displays contents of all available Policy-Based Routing (PBR) class maps, or of a specified PBR class map. PBR class maps are used by PBR policy maps. PBR class maps are dynamic maps that are created in class-map-configuration mode. Dynamic class maps are composed of statements that match IPv4 or IPv6 access control lists.

Command Mode

EXEC

Command Syntax

`show class-map type pbr` [***map_name***]

Parameters

map_name Name of class map displayed by the command. If no parameter is entered, command show all available PBR class maps.

Related Command

show policy-map type pbr displays PBR policy maps.

Example

This command displays the contents of the PBR class map ***CMAP1***.

```
switch# show class-map type pbr CMAP1
  Class-map: CMAP1 (match-any)
    Match: 10 ip access-group PBRgroup1
    Match: 20 ip access-group PBRgroup2
    Match: 30 ip access-group PBRgroup3 switch>
```

10.2.7.57 show class-map type qos

The **show class-map**command displays contents of all available QoS class maps. QoS class maps are used by QoS policy maps. QoS class maps are dynamic maps that are created in ***class-map*** configuration mode. Dynamic class maps are composed of statements that match IPv4 or IPv6 access control lists.

Command Mode

EXEC

Command Syntax

show class-map [**type qos**][**MAP_NAME**]

Parameters

MAP_NAME Name of class map displayed by the command.

- ***no parameter*** Command displays all QoS class maps.
- ***name_text*** Command displays specified QoS class maps.

show class-map and **show class-map type qos** are identical commands.

Related Command

show class-map type control-plane displays control plane class maps.

Example

This command displays the available QoS class maps.

```
switch# show class-map type qos
  Class-map: CM-Q1 (match-any)
    Match: ipv6 access-group name LIST-1
  Class-map: CM-Q2 (match-any)
    Match: ip access-group name LIST-2 switch>
```

10.2.7.58 show policy-map copp

The **show policy-map copp** command displays contents of the control-plane policy map. Control-plane policy maps are applied to the control plane, and copp-system-policy is the only supported policy map.

Command Mode

EXEC

Command Syntax

show policy-map copp copp-system-policy

Example

This command displays the contents and throughput of the policy map applied to the control plane.

```
switch# show policy-map copp copp-system-policy
Service-policy input: copp-system-policy
  Number of units programmed: 1
  Hardware programming status: Successful

  Class-map: copp-system-bpdu (match-any)
       shape : 5000 pps
       bandwidth : 5000 pps
      Out Packets : 2
      Drop Packets : 0

  Class-map: copp-system-lacp (match-any)
       shape : 5000 pps
```

```
        bandwidth : 5000 pps
       Out Packets : 0
       Drop Packets : 0 switch>
```

10.2.7.59 show policy-map interface type qos counters

The **show policy-map interface** command displays the quantity of packets that are filtered by ACLs applied to a interface.

Command Mode

EXEC

Command Syntax

show policy-map [**INTERFACE_NAME**][**type qos**][**TRAFFIC**] **counters**

Parameters

- **INTERFACE_NAME** Filters policy map list by interfaces. Options include:
  - ***no parameter*** Displays data for all configured interfaces.
  - **interface ethernet *e_range*** Ethernet ports for which command displays policy maps.
  - **interface port-channel *p_range*** Port channels for which command displays policy maps.
- **TRAFFIC** Filters policy maps by the traffic they manage. Options include:
  - ***no parameter*** Policy maps that manage interfaces ingress traffic (same as **input** option).
  - **input** Policy maps that manage interfaces ingress traffic.

Example

This command displays the policy maps applied to interfaces Ethernet ***7*** and ***8***.

```
switch# show policy-map interface ethernet 7-8
Service-policy input: PMAP-1
  Hardware programming status: Successful

  Class-map: cmap-1 (match-any)
    Match: ip access-group name LIST-2
       set cos 6

  Class-map: class-default (match-any)

Service-policy input: PMAP-2
  Hardware programming status: Successful

  Class-map: cmap-2 (match-any)
    Match: ip access-group name LIST-2
       set dscp 10

  Class-map: class-default (match-any)

switch#
```

10.2.7.60 show policy-map interface type qos

The **show policy-map interface** command displays contents of the policy maps applied to specified interfaces or to the control plane.

Command Mode

EXEC

Command Syntax

show policy-map interface **INTERFACE_NAME** [**type qos**] [**TRAFFIC**]

Parameters

- **INTERFACE_NAME** Filters policy map list by interfaces. Options include:
  - **ethernet** ***e_range*** Ethernet ports for which command displays policy maps.
  - **port-channel** ***p_range*** Port channels for which command displays policy maps.
- **TRAFFIC** Filters policy maps by the traffic they manage. Options include:
  - ***no parameter*** Policy maps that manage interfaces ingress traffic (same as **input** option).
  - **input** Policy maps that manage interfaces ingress traffic.

Example

This command displays the policy maps applied to interfaces Ethernet ***7*** and ***8***.

```
switch# show policy-map interface ethernet 7-8
Service-policy input: PMAP-1
  Hardware programming status: Successful

  Class-map: cmap-1 (match-any)
    Match: ip access-group name LIST-2
       set cos 6

  Class-map: class-default (match-any)

Service-policy input: PMAP-2
  Hardware programming status: Successful

  Class-map: cmap-2 (match-any)
    Match: ip access-group name LIST-2
       set dscp 10

  Class-map: class-default (match-any)

switch#
```

10.2.7.61 show policy-map type copp

The **show policy-map type copp** command displays contents of control plane policy maps. Control-plane policy maps are applied to the control plane; copp-system-policy is the only supported policy map.

Command options filter the output to display contents of all policy maps, contents of a specified policy map, or contents of a single class map within a specified policy map.

Command Mode

EXEC

Command Syntax

show policy-map type copp copp-system-policy [**CMAP_NAME**]

Parameters

CMAP_NAME Name of class map displayed by the command.

- ***no parameter*** Command displays all class maps in specified policy map.
- ***class_name*** Command displays specified class map.

Example

This command displays the contents of the copp-system-bpdu class map in the copp-system-policy policy maps.

```
switch# show policy-map type copp copp-system-policy class copp-system-bpdu
  Class-map: copp-system-bpdu (match-any)
       shape : 5000 pps
       bandwidth : 5000 pps switch>
```

10.2.7.62 show policy-map type pbr

The **show policy-map pbr** command displays contents of Policy-Based Routing (PBR) policy maps. PBR policy maps are applied to Ethernet interfaces, port channel interfaces or switch virtual interfaces (SVIs).

Command options filter the output to either display contents of all policy maps, contents of a specified policy map, or summary contents of all or a specified policy map.

Command Mode

EXEC

Command Syntax

show policy-map type pbr [**PMAP_NAME**][**DATA_LEVEL**]

Parameters

- **PMAP_NAME** Name of policy map displayed by the command.
  - ***no parameter*** Command displays all policy maps.
  - ***policy_map*** Command displays specified policy map.
- **DATA_LEVEL** Type of information the command displays. Values include:
  - ***no parameter*** Command displays all class maps in specified policy map.
  - **summary** Command displays summary data for the specified policy map.

Example

This command displays the contents of all PBR policy maps in ***running-config***.

```
switch# show policy-map type pbr
Service policy PMAP1
Configured on:
Applied on:
10: Class-map: CMAP1 (match-any)
Match: 10 ip access-group PBRgroup1
Match: 20 ip access-group PBRgroup2
Match: 30 ip access-group PBRgroup3
Configured actions: set nexthop 172.16.10.12
20: Class-map: CMAP2 (match-any)
Match: 10 ip access-group PBRgroup1
Match: 10 ip access-group PBRgroup4
Match: 20 ip access-group PBRgroup5
Configured actions: set nexthop 192.168.15.15 switch#
```

10.2.7.63 show policy-map type qos counters

The **show policy-map counters** command displays the quantity of packets that are filtered by the ACLs that comprise a specified QoS policy map.

Command Mode

EXEC

Command Syntax

show policy-map [**type qos**] ***pmap_name*** [**TRAFFIC**] **counters** [**INFO_LEVEL**]

Parameters

- ***pmap_name*** Name of policy map displayed by the command.
- **TRAFFIC** Filters policy maps by the traffic they manage. Options include:
  - ***no parameter*** Policy maps that manage interfaces ingress traffic (same as **input** option).
  - **input** Policy maps that manage interfaces ingress traffic.
- **INFO_LEVEL** amount of information that is displayed. Options include:
  - ***no parameter*** displays summarized information about the policy map.
  - **detail** displays detailed policy map information.

10.2.7.64 show policy-map type qos

The **show policy-map qos** command displays contents of QoS policy maps. QoS policy maps are applied to Ethernet or port channel interfaces.

Command options filter the output to either display contents of all policy maps, contents of a specified policy map, or contents of a single class map within a specified policy map.

Command Mode

EXEC

Command Syntax

show policy-map [**type qos**][**PMAP_NAME** [**CMAP_NAME**]]

Parameters

- **PMAP_NAME** Name of policy map displayed by the command.
  - ***no parameter*** Command displays all policy maps.
  - ***policy_map*** Command displays specified policy map.
- **CMAP_NAME** Name of class map displayed by the command. This option is available only when the command includes a policy map name.
  - ***no parameter*** Command displays all class maps in specified policy map.
  - ***class_name*** Command displays specified class map.

Example

This command displays the contents of all QoS policy maps in ***running-config***.

```
switch# show policy-map type qos
Service-policy input: PMAP-1
  Hardware programming status: Successful

  Class-map: xeter (match-any)
    Match: ip access-group name LIST-1
       set cos 6

  Class-map: class-default (match-any)

Service-policy PMAP-2

  Class-map: class-default (match-any)

switch#
```

10.2.7.65 show traffic-policy

The **show traffic-policy** command displays traffic policy information on the interface.

Command Mode

EXEC

Command Syntax

show traffic-policy NAME interface

show traffic-policy interface [**DETAILS**]

Parameters

DETAILS Details requested. Options include:

- **summary** Display summary information about the policy.
  - **errors** Display all configured remote grantees, associated profile name and latest update.
  - **details** Display all interfaces on which the policy has been configured.

Examples

- This command displays the sumary information configured on the switch interfaces.

```
switch(config-traffic-policies)# show traffic-policy interface summary
Traffic policy samplePolicy
   Configured on interfaces: Ethernet1/1, Ethernet2/1, Ethernet3/1, ...
   Applied on interfaces for IPv4 traffic: Ethernet1/1, Ethernet2/1, Ethernet3/1, ...
   Applied on interfaces for IPv6 traffic:
   Total number of rules configured: 3
      match SIMPLE ipv4
      match ipv4-all-default ipv4
      match ipv6-all-default ipv6
```

- This command displays information about the traffic policy named ***samplePolicy***.

```
switch(config-traffic-policies)# show traffic-policy samplePolicy interface
Traffic policy samplePolicy
   Configured on interfaces: Ethernet1/1, Ethernet2/1, Ethernet3/1, ...
   Applied on interfaces for IPv4 traffic: Ethernet1/1, Ethernet2/1, Ethernet3/1, ...
   Applied on interfaces for IPv6 traffic:
   Total number of rules configured: 3
      match SIMPLE ipv4
         Source prefix: 192.0.2.0/24
                        198.51.100.0/24
         Destination prefix: 203.0.113.0/24
         Protocol: tcp
            Source port: 50-100
                         110-200
         Actions: Drop
      match ipv4-all-default ipv4
      match ipv6-all-default ipv6
```

- This command displays all interfaces on which samplePolicy has been configured.

```
switch(config-traffic-policies)# show traffic-policy interface detail
Traffic policy samplePolicy
   Configured on interfaces: Ethernet1/1, Ethernet2/1, Ethernet3/1, Ethernet4/1
   Applied on interfaces for IPv4 traffic: Ethernet1/1, Ethernet2/1, Ethernet3/1, Ethernet4/1
   Applied on interfaces for IPv6 traffic:
   Total number of rules configured: 3
      match SIMPLE ipv4
         Source prefix: 192.0.2.0/24
                        198.51.100.0/24
         Destination prefix: 203.0.113.0/24
         Protocol: tcp
            Source port: 50-100
                         110-200
         Actions: Drop
      match ipv4-all-default ipv4
      match ipv6-all-default ipv6
```

- This command displays installation errors for a match statement. The example has no errors.

```
switch(config-traffic-policies)# show traffic-policy interface errors
Traffic policy samplePolicy
   Failed on interface for IPv4 traffic:
   Failed on interface for IPv6 traffic:
```

Chapter 11

Interface Configuration

This chapter describes Ethernet ports supported by Arista switches as well as channel groups, port channels, port channel interfaces, Link Aggregation Control Protocol (LACP), and Multi-Chassis Link Aggregation. This chapter contains the following sections:

- Ethernet Ports
- Port Channels and LACP
- Multi-Chassis Link Aggregation
- Data Transfer
- Octal Port Renumber to Four Interfaces

11.1 Ethernet Ports

This section describes Ethernet ports supported by Arista switches. Topics covered in this section include:

- Ethernet Ports Introduction
- Ethernet Standards
- Ethernet Physical Layer
- Interfaces
- MRU Enforcement
- Ethernet Configuration Procedures
- Ethernet Configuration Commands

11.1.1 Ethernet Ports Introduction

Arista switches support a variety of Ethernet network interfaces. This chapter describes the configuration and monitoring options available in Arista switching platforms.

11.1.2 Ethernet Standards

Ethernet, standardized in ***IEEE 802.3***, is a group of technologies used for communication over local area networks. Ethernet communication divides data streams into frames containing addresses (source and destination), payload, and Cyclical Redundancy Check (CRC) information.

IEEE 802.3 also describes two types of optical fiber: Single-Mode Fiber (SMF) and Multi-Mode Fiber (MMF). MMF range limits from ***50*** to ***500*** meters range.

11.1.2.1 100 Gigabit Ethernet

The 100 Gigabit Ethernet (100GbE) standard defines an Ethernet implementation with a nominal data rate of 100 billion bits per second over 10x10G, 4x25G, or 1x100G. 100 Gigabit Ethernet implements full duplex point to point links connected by network switches. Arista switches support 100GBASE-10SR through MXP ports.

11.1.2.2 40 Gigabit Ethernet

The 40 Gigabit Ethernet (40GbE) standard defines an Ethernet implementation with a nominal data rate of 40 billion bits per second over multiple 10 gigabit lanes. 40 Gigabit Ethernet implements full duplex point to point

links connected by network switches. 40 gigabit Ethernet standards are named ***40GBASE-xyz***, as interpreted by 40GBASE-xyz Interpretation.

Table 54: 40GBASE-xyz Interpretation

x	y	z
Non-fiber media type, or fiber wavelength	PHY encoding	Number of WWDM wavelengths or XAUI Lanes
C = Copper F = Serial SMF K = Backplane L = Long (1310 nm) S = Short (850 nm)	R = LAN PHY (64B/66B)	No value = 1 (serial) 4 = 4 WWDM wavelengths or XAUI Lanes

11.1.2.3 10 Gigabit Ethernet

The 10 Gigabit Ethernet (10GbE) standard defines an Ethernet implementation with a nominal data rate of 10 billion bits per second. 10 Gigabit Ethernet implements full duplex point to point links connected by network switches. Half duplex operation, hubs and CSMA/CD do not exist in 10GbE. The standard encompasses several PHY standards; a networking device may support different PHY types through pluggable PHY modules. 10GbE standards are named ***10GBASE-xyz***, as interpreted by 10GBASE-xyz Interpretation.

Table 55: 10GBASE-xyz Interpretation

x	y	z
media type or wavelength, if media type is fiber	PHY encoding type	Number of WWDM wavelengths or XAUI Lanes
C = Copper (twin axial) T = Twisted Pair S = Short (850 nm) L = Long (1310 nm) E = Extended (1550 nm) Z = Ultra extended (1550 nm)	R = LAN PHY (64B/66B) X = LAN PHY (8B/10B) W = WAN PHY(*) (64B/66B)	If omitted, value = 1 (serial) 4 = 4 WWDM wavelengths or XAUI Lanes

11.1.2.4 Gigabit Ethernet

The Gigabit Ethernet (GbE), defined by ***IEEE 802.3-2008***, describes an Ethernet version with a nominal data rate of one billion bits per second. GbE cables and equipment are similar to those used in previous standards. While full-duplex links in switches is the typical implementation, the specification permits half-duplex links connected through hubs.

Gigabit Ethernet physical layer standards that Arista switches support include 1000BASE-X (optical fiber), 1000BASE-T (twisted pair cable), and 1000BASE-CX (balanced copper cable).

- 1000BASE-SX is a fiber optic standard that utilizes multi-mode fiber supporting 770 to 860 nm, near infrared (NIR) light wavelength to transmit data over distances ranging from ***220*** to ***550*** meters. 1000BASE-SX is typically used for intra-building links in large office buildings, co-location facilities and carrier neutral Internet exchanges.
- 1000BASE-LX is a fiber standard that utilizes a long wavelength laser (***1,2701,355*** nm), with a RMS spectral width of 4 nm to transmit data up to ***5*** km. 1000BASE-LX can run on all common types of multi-mode fiber with a maximum segment length of ***550*** m.
- 1000BASE-T is a standard for gigabit Ethernet over copper wiring. Each 1000BASE-T network segment can be a maximum length of ***100*** meters.

11.1.2.5 10/100/1000 BASE-T

Arista switches provide 10/100/1000 BASE-T Mbps Ethernet out of band management ports. Auto-negotiation is enabled on these interfaces. Speed (10/100/1000), duplex (half/full), and flow control settings are available using the appropriate **`speed`** and **`flowcontrol`** commands.

11.1.2.6 Power over Ethernet (PoE)

Selected Arista switches provide Power over Ethernet (PoE) to power connected devices. Arista's PoE implementation is compliant with IEEE standards ***802.3af*** and ***802.3at***, and includes partial support for ***802.3bt***.

When a standards-compliant Powered Device (PD) is connected to a PoE-enabled Ethernet port, it is recognized by a specific resistor signature, and its power class is determined by hardware negotiation; more granular power adjustments can then be managed by Link Layer Discovery Protocol (LLDP).

11.1.2.7 Link Fault Signaling

Link Fault Signaling (LFS) is a mechanism by which remote link faults are transmitted to the peer over the link that is experiencing problems by configuring specific actions. LFS operates between the remote Reconciliation Sublayer (remote RS) and the local Reconciliation Sub-layer (local RS). Faults that are detected between the remote RS and the local RS are treated by the local RS as Local Faults.

LFS enables monitoring FCS and Symbol errors on an interface and if they exceed a configured threshold, one of the following three actions are enabled.

- Disable the error on the interface
- Generate system log messages
- Generate a link fault

11.1.3 Ethernet Physical Layer

The Ethernet Physical Layer (PHY) includes hardware components connecting a switchs MAC layer to the transceiver, cable, and ultimately a peer link partner.

Data exist in digital form at the MAC layer. On the line side of the PHY, data exist as analog signals: light blips on optical fiber or voltage pulses on copper cable. Signals may be distorted while in transit and recovery may require signal processing. Ethernet physical layer components include a PHY and a transceiver.

11.1.3.1 PHYs

The PHY provides translation services between the MAC layer and transceiver. It also helps to establish links between the local MAC layer and peer devices by detecting and signaling fault conditions. The PHY line-side interface receives Ethernet frames from the link partner as analog waveforms. The PHY uses signal processing to recover the encoded bits, then sends them to the MAC layer.

PHY line-side interface components and their functions include:

- Physical Medium Attachment (PMA): Framing, octet synchronization, scrambling / descrambling.
- Physical Medium Dependent (PMD): Consists of the transceiver.
- Physical Coding Sublayer (PCS): Performs auto-negotiation and coding (8B/10B or 64B/66B).

The MAC sublayer of the PHY provides a logical connection between the MAC layer and the peer device by initializing, controlling, and managing the connection with the peer.

Ethernet frames transmitted by the switch are received by the PHY system-side interface as a sequence of digital bits. The PHY encodes them into a media-specific waveform for transmission through the line-side interface and transceiver to the link peer. This encoding may include signal processing, such as signal pre-distortion and forward error correction.

PHY system-side interface components and their functions include:

- **10 Gigabit Attachment Unit Interface (XAUI):** Connects an Ethernet MAC to a 10G PHY.
- **Serial Gigabit Media Independent Attachment (SGMII):** Connects an Ethernet MAC to a 1G PHY.

11.1.3.2 Transceivers

A transceiver connects the PHY to an external cable (optical fiber or twisted-pair copper) and through a physical connector (LC jack for fiber or RJ-45 jack for copper).

- Optical transceivers convert the PHY signal into light pulses that are sent through optical fiber.
- Copper transceivers connect the PHY to twisted-pair copper cabling.

Arista Small Form-Factor Pluggable (SFP+) and Quad Small Form Factor Pluggable (QSFP+) modules and cables provide high-density, low-power Ethernet connectivity over fiber and copper media. Arista offers transceivers that span data rates, media types, and transmission distances.

Arista 10 Gigabit Ethernet SFP+ Modules

- 10GBASE-SR (Short Reach)
  - Link length maximum 300 meters over multi-mode fiber.
- Optical interoperability with 10GBASE-SRL.
- 10GBASE-SRL (Short Reach Lite)
  - Link length maximum 100 meters over multi-mode fiber.
  - Optical interoperability with 10GBASE-SR.
- 10GBASE-LRL (Long Reach Lite)
  - Link length maximum 1 km over single-mode fiber.
  - Optical interoperability with 10GBASE-LR (1 km maximum).
- 10GBASE-LR (Long Reach)
  - Link length maximum 10 km over single-mode fiber.
  - Optical interoperability with 10GBASE-LRL (1 km maximum).
- 10GBASE-LRM (Long Reach Multimode)
  - Link length maximum 220 meters over multi-mode fiber (50 um and 62.5 um).
- 10GBASE-ER (Extended Reach)
  - Link length maximum 40 km over single-mode fiber.
- 10GBASE-ZR (Ultra-Extended Reach)
  - Link length maximum 80 km over single-mode fiber.
- 10GBASE-DWDM (Dense Wavelength Division Multiplexing)
  - Link length maximum 80 km over single-mode fiber (40 color options).
  - Tunable SFP+ Optics Module, Full C-Band 50 GHz ITU Grid, up to 80km over duplex SMF.

Arista 10 Gigabit Ethernet CR Cable Modules

- 10GBASE-CR SFP+ to SFP+ Cables
  - Link lengths of 0.5, 1, 1.5, 2, 2.5, 3, 5, and 7 meters over twinax copper cable.
  - Includes SFP+ connectors on both ends.
- 4 x 10GbE QSFP+ to 4 x SFP+ twinax copper cables.
  - Link lengths of 0.5, 1, 2, 3, and 5 meters over twinax copper cable

Arista 25 Gigabit Ethernet Modules

- 25GBASE-CR SFP28 Cable

    - Capable of 10G/25G with link length of 1 to 5 meters.
- AOC-S-S-25G SFP28 to SFP28 25GbE Active Optical Cable.
    - Link length of 3 to 30 meters.
- SFP-25G-SR SFP28 Optics Module
    - Link length up to 70m over OM3 MMF or 100m over OM4 MMF.
- SFP-25G-LR SFP28 Optics Module
    - Link length up to 10 kilometers over duplex SMF

Arista 40 Gigabit Ethernet QSFP+ Cables and Optics

- 40GBASE-SR4 QSFP+ Transceiver.
    - Link length maximum 100 meters over parallel OM3 or 150 meters over OM4 MMF.
    - Optical interoperability with 40GBASE-XSR4 (100/150 meter maximum).
- 40GBASE-XSR4 QSFP+ Transceiver.
    - Link length maximum 300 meters over parallel OM3 or 450 meters over OM4 MMF.
    - Optical interoperability with 40GBASE-SR4 (100/150 meter maximum).
- 40GBASE-LR4 QSFP+.
    - Link length maximum 10 km over duplex single-mode fiber.
- 40GBASE-CR4 QSFP+ to QSFP+ twinax copper cables.
    - Link lengths of 1, 2, 3, 5, and 7 meters over twinax copper cable.
- 40G-SRBD Bidirectional QSFP+ Optic.
    - Link length maximum up to 100 meters over parallel OM3 or 150 meters over OM4 MMF.
- 40G Univ QSFP+ Optic.
    - Link length maximum up to 150 meters over duplex OM3/OM4 and 500 meters over duplex SMF.
- 40GBASE-LRL QSFP+ Optic.
    - Link length maximum up to 1 kilometer over duplex SMF.
- 40GBASE-PLRL4 QSFP+ Optic.
    - Link length maximum up to 1 kilometer over parallel SMF (4x10G LR up to 1 km).
- 40GBASE-PLR4 QSFP+ Optic
    - Link length maximum up to 1 kilometer over parallel SMF (4x10G LR up to 1 km).
- 40GBASE-ER QSFP+ Optic.
    - Link length maximum up to 40 kilometers duplex SMF.

Arista Gigabit Ethernet SFP Options

- 1000BASE-SX (Short Haul).
    - Multi-mode fiber.
    - Link length maximum 550 meter.
- 1000BASE-LX (Long Haul).
    - Single-mode fiber.
    - Link length maximum 10 km (single mode).
- 1000BASE-T (RJ-45 Copper).
    - Category 5 cabling.
    - Full duplex 1000Mbps connectivity.

Arista 100 Gigabit Ethernet QSFP Modules

- 100GBASE-SR4 QSFP transceiver.
  - Link length up to 70 meters over parallel OM3 or 100 meters over OM4 multi-mode fiber.
- 100GBASE-SWDM4 QSFP transceiver.
  - Link length up to 70 meters over OM3 or 100 meters over OM4 duplex multi-mode fiber.
- 100GBASE-SRBD BIDI QSFP transceiver.
  - Link length up to 70 meters over OM3 or 100 meters over OM4 duplex multi-mode fiber.
- 100GBASE-PSM4 40G/100G dual speed QSFP Optics Module.
  - Link length up to 500 meters over parallel single-mode fiber.
- 100GBASE-CWDM4 40G/100G dual speed QSFP Optics Module.
  - Link length up to 2 km over duplex single-mode fiber.
- 100GBASE-LRL4 QSFP Optics Module.
  - Link length up to 2 km over duplex single-mode fiber.
- 100GBASE-LR4 QSFP Optics Module.
  - Link length up to 10 km over duplex single-mode fiber.
- 100GBASE-ERL4 QSFP Optics Module.
  - Link length up to 40 km over duplex single-mode fiber.
- 100G DWDM QSFP transceiver.
  - Link length up to 80 km over single-mode fiber.
- 100GBASE-CR4 QSFP to QSFP Twinax Copper Cable.
  - Link length of 1 to 5 meters.
- 100GBASE-CR4 QSFP to 4 x 25GbE SFP Twinax Copper Cable.
  - Link length of 1 to 5 meters.

Internal Ports

Several Arista switches include internal ports that connect directly to an external cable through an RJ-45 jack. Internal ports available on Arista switches include:

- 100/1000BASE-T (7048T-A)
- 100/1000/10GBASE-T (7050-T)

AOC Cables

- AOC-Q-Q-100G QSFP 100GbE Active Optical Cable.
  - Link length of 3 to 30 meters
- AOC-Q-Q-40G QSFP+ to QSFP+ 40GbE Active Optical Cable.
  - Link length of 3 to 100 meters.
- AOC-S-S-25G SFP28 to SFP28 25GbE Active Optical Cable.
  - Link length of 3 to 30 meters.

11.1.3.2.1 400GBASE-ZR Transceivers

400GBASE-ZR transceiver is the industry’s first multi-vendor DWDM standard, a Digital Coherent Optical module with tunable laser, using DWDM multiplexing and 16QAM modulation and capable of delivering 400G per port over distances up to 120 km.

Key software features supported:

- Compliant with CMIS4.0/CMIS4.1 (CMIS5.0) and Coherent CMIS
- Frequency tuning (100GHz and 75GHz grids)
- DOM monitoring, including VDM (Versatile Diagnostics Monitoring) pages, defined in CMIS4.0
- Coherent alarms and faults, including pages, defined by Coherent CMIS
- Configurable Tx output power
- A separate command for shutting down Tx output path for unidirectional mode
- 1-sec update of pre-FEC BER, OSNR, ESNR
- Support of 4x100G mode for the 400GBASE-ZR transceivers (*EOS release 4.25.2F*)
- Support of 400GBASE-ZR with Open Forward Error Correction (O-FEC) (*EOS release 4.25.2F*)
- Override a transceiver slot's maximum power limit (*EOS release 4.25.2F*)

Platform Compatibility

The 400GBASE-ZR transceiver is a power class 8 module with power consumption up to 20W, the highest power consumption among 400G transceivers.

In theory, every 400GBASE OSFP or QSFP-DD switch is qualified to host 400GBASE-ZR transceivers. However, the actual number of 400GBASE-ZR modules that can be plugged in the switch simultaneously may vary, depending on platform (modular vs fixed) and ASIC type. It is always recommended to discuss installation of 400GBASE-ZR modules with Arista support.

Using 400GBASE-ZR in Combination with OSFP-LS

Arista EOS allows using OSFP-LS (pluggable line system in the OSFP form factor), instead of traditional DCI line systems.

11.1.3.2.1.1Configuring 400GBASE-ZR Transceivers

Laser Frequency Configuration

All coherent modules, including 400GBASE-ZR, require their laser frequency to be explicitly configured. To configure laser frequency in 400GBASE-ZR, use the **`transceiver frequency`** command under the defined interface, providing frequency in Gigahertz, in the range ***191.3-196.1*** THz. If laser frequency is not configured or is configured outside the valid range, all interfaces associated with the port are put into error disabled state.

```
switch# conf
switch(config)#
switch(config)# interface Ethernet12/1
switch(config-if-Et12/1)# transceiver frequency 193100
switch(config-if-Et12/1)#
```

Configured and Operational frequency settings can be verified using the **`show interface transceiver hardware`** command.

Note:

- It can take up to ***90*** seconds for the 400GBASE-ZR module to fully complete frequency tuning.
- The frequency plan for 400GBASE-ZR modules support channel spacings of ***100GHz*** or ***75GHz***. It is recommended to use operating frequency channel definitions in chapter 15 of [4].

Transmit Output Power Configuration

CMIS4.0 defines support of configurable transmit output power as an optional feature. Check the output of **`show interface transceiver eeprom`** command for page 04h, registers ***196-201*** to see if it is supported.

```
Programmable output power advertisement (04h:196-201):
    Lane programmable output power supported (04h:196): true
```

```
    Min programmable output power (04h:198-199): -14 dB
    Max programmable output power (04h:200-201): -10 dB
```

Transmit Output Power can be configured using **transceiver transmitter signal power** command.

```
switch# conf
switch(config)# interface Ethernet14/1
switch(config-if-Et14/1)# transceiver transmitter signal-power ?
switch(config-if-Et14/1)# transceiver transmitter signal-power -10
```

Configured Tx Power is verified using the **show interface transceiver hardware** command. Actual operational Tx Power can be verified using **show interface transceiver** command.

Note:

- 400GBASE-ZR does not have a recommended level of transmit laser power. In most cases, it is OK to stay with default power.
- Some vendors may not populate the range of supported output power correctly. The range ***-10 to -14 dBm*** should be supported. Contact Arista Support if you are planning to configure output power outside of this range.

Transmit Output Disable and Unidirectional Mode

In the coherent optics, **shutdown** command impacts both transmit and receive path and can't be used to set up unidirectional mode. To shut down Tx output path, use the **transceiver transmitter disabled** command:

```
switch# conf
switch(config)# interface Ethernet14/1
switch(config-if-Et14/1)# transceiver transmitter DISABLED
```

To re-enable transmit output, use the **no** or **default** version of the above configuration command.

Configuring 4x100G mode for 400GBASE-ZR transceivers

4x100G mode connects 4 host ethernet interfaces, over electrical interfaces 100GAUI-2 to the single optical lane, acting as 4x1 muxponder. To enable the mode on a 400GBASE-ZR transceiver, configure the speed on each of the 4 interfaces.

```
switch(config-if-Et4/1)# speed 100g-2
switch(config-if-Et4/3,4/5,4/7)# speed 100g-2

switch# show interfaces ethernet 4/1-5/8 status
Port    Name   Status      Vlan  Duplex Speed  Type      Flags Encapsulation
Et4/1          connected    1     full   100G   400GBASE-ZR
Et4/3          connected    1     full   100G   400GBASE-ZR
Et4/5          connected    1     full   100G   400GBASE-ZR
Et4/7          connected    1     full   100G   400GBASE-ZR
Et5/1          connected          1     full   100G   400GBASE-ZR
Et5/3          connected          1     full   100G   400GBASE-ZR
Et5/5          connected          1     full   100G   400GBASE-ZR
Et5/7          connected    1     full   100G   400GBASE-ZR
```

Each of the ethernet interfaces can be independently shut down without affecting the transmission over the other interfaces.

```
switch(config-if-Et4/1,4/3,4/5)# shutdown
switch# show interfaces ethernet 4/1-5/8 status
Port    Name   Status      Vlan  Duplex Speed  Type      Flags Encapsulation
Et4/1          disabled     1     full   100G   400GBASE-ZR
Et4/3          disabled     1     full   100G   400GBASE-ZR
Et4/5          disabled     1     full   100G   400GBASE-ZR
Et4/7          connected    1     full   100G   400GBASE-ZR
Et5/1          notconnect         1     full   100G   400GBASE-ZR
Et5/3          notconnect         1     full   100G   400GBASE-ZR
```

```
Et5/5          notconnect        1     full   100G   400GBASE-ZR
Et5/7          connected    1     full   100G   400GBASE-ZR

switch# show interfaces ethernet 4/1 transceiver
If device is externally calibrated, only calibrated values are printed.
N/A: not applicable, Tx: transmit, Rx: receive.
mA: milliamperes, dBm: decibels (milliwatts).
                              Bias   Optical   Optical
       Temp    Voltage    Current    Tx Power  Rx Power
Port   (Celsius)  (Volts)    (mA)   (dBm)     (dBm)   Last Update
-----  ---------  --------   --------  --------  --------    -------------------
Et4/1      59.00      3.25     279.70     -9.46     -8.29    0:00:01 ago
```

Shutting down all 4 interfaces will however disable the laser:

```
switch(config-if-Et4/7)# shutdown
switch# show interfaces ethernet 4/1-5/8 status
Port    Name   Status      Vlan  Duplex Speed  Type      Flags Encapsulation
Et4/1          disabled    1     full   100G   400GBASE-ZR
Et4/3          disabled    1     full   100G   400GBASE-ZR
Et4/5          disabled    1     full   100G   400GBASE-ZR
Et4/7          disabled    1     full   100G   400GBASE-ZR
Et5/1          notconnect        1     full   100G   400GBASE-ZR
Et5/3          notconnect        1     full   100G   400GBASE-ZR
Et5/5          notconnect        1     full   100G   400GBASE-ZR
Et5/7          notconnect   1     full   100G   400GBASE-ZR

switch# show interfaces ethernet 4/1 transceiver
If device is externally calibrated, only calibrated values are printed.
N/A: not applicable, Tx: transmit, Rx: receive.
mA: milliamperes, dBm: decibels (milliwatts).
                              Bias   Optical   Optical
       Temp    Voltage    Current    Tx Power  Rx Power
Port   (Celsius)  (Volts)    (mA)   (dBm)     (dBm)   Last Update
-----  ---------  --------   --------  --------  --------    -------------------
Et4/1      44.00      3.33       0.00    -30.00    -30.00    0:00:01 ago
```

Configuring Open Forward Error Correction on 400GBASE-ZR

Open FEC (O-FEC) is a Forward Error Correction encoder and decoder, specified in the Open ZR+ MSA [2], with an overhead of 15.3% and a net coding gain of 11.6dB for 16QAM modulation. Compared to the Concatenated FEC (C-FEC), which is a default FEC specified for 400GBASE-ZR coherent transceivers by CMIS4.0, O-FEC provides higher coding gain (11.6dB for O-FEC vs 10.8 dB for C-FEC /16QAM), and it can correct pre-FEC BER of up to 2e-2.

The **`error-correction encoding <FEC>`** command configures the transceiver's optical transmission to use O-FEC:

```
switch(config-if-Et4/1)# error-correction encoding open
```

Transceiver slot's Maximum Power Limit

The **`transceiver power ignore`** command allows EOS to bypass the power check for the transceivers to be operational.

```
Switch(config-if-Et4/1)# transceiver power ignore
! You can risk damaging hardware by using transceiver modules with high power
 consumption. We recommend that you do this only under direction from Arista
 Networks.
```

11.1.3.2.1.2Show Commands

- **show interface transceiver eeprom**

  The **`show interface transceiver eeprom`** command displays the parsed capabilities.

  Example

For 400GBASE-ZR, parsing of frequency tuning, power tuning (page 04h) and VDM configuration pages (20h-23h) is added.

```
switch# show interface Ethernet15/1 transceiver eeprom
Ethernet15 EEPROM:
...
  Frequency tuning support (04h:128-129):
    Grid spacing capabilities (04h:128):
      100 GHz grid supported (04h:128): true
      12.5 GHz grid supported (04h:128): false
      25 GHz grid supported (04h:128): false
      3.125 GHz grid supported (04h:128): false
      33 GHz grid supported (04h:128): false
      50 GHz grid supported (04h:128): false
      6.25 GHz grid supported (04h:128): false
      75 GHz grid supported (04h:128): true
    Tunable wavelength (04h:128): true
    Fine tuning support (04h:129): false
  Supported channel boundaries (04h:130-161):
    100 GHz grid (04h:150-153):
      Lowest channel (04h:150-151): -18
      Lowest frequency (04h:150-151): 191300000 MHz
      Highest channel (04h:152-153): 30
      Highest frequency (04h:152-153): 196100000 MHz
    75 GHz grid (04h:158-161):
      Lowest channel (04h:158-159): -72
      Lowest frequency (04h:158-159): 191300000 MHz
      Highest channel (04h:160-161): 120
      Highest frequency (04h:160-161): 196100000 MHz
  Programmable output power advertisement (04h:196-201):
    Lane programmable output power supported (04h:196): false
  VDM configuration (20h:128-255;21h:128-255):
    VDM group 1 (20h:128-255):
      Parameter 1 (20h:128-129):
        Lane (20h:128): 0
        Threshold ID (20h:128): 0
        Parameter type (20h:129): Laser temperature
      Parameter 3 (20h:132-133):
        Lane (20h:132): 0
        Threshold ID (20h:132): 2
        Parameter type (20h:133): eSNR host input
      Parameter 4 (20h:134-135):
        Lane (20h:134): 1
        Threshold ID (20h:134): 2
        Parameter type (20h:135): eSNR host input
      Parameter 5 (20h:136-137):
        Lane (20h:136): 2
        Threshold ID (20h:136): 2
        Parameter type (20h:137): eSNR host input
...
       Parameter 84 (21h:166-167):
        Lane (21h:166): 0
        Threshold ID (21h:166): 14
        Parameter type (21h:167): MER
  Number of VDM groups supported (2Fh:128): 2
```

- **show interface transceiver dom**

  The **show interface transceiver dom** command displays the most important current performance data on the media (line) side.

Example

```
switch# show interface Ethernet11/1 transceiver dom
Ch: Channel, N/A: not applicable, TX: transmit, RX: receive
mA: milliamperes, dBm: decibels (milliwatts), C: Celsius, V: Volts

Port 11
Last update: 0:00:05 ago
                                          Value
                                        ----------------
   Case temperature                        66.59 C
   Voltage                                  3.26 V
   TX power                               -10.23 dBm
   RX total power                         -11.61 dBm
   RX channel power                       -11.94 dBm
   Pre-FEC BER                           1.82e-03
   Post-FEC errored frames ratio         0.00e+00
   Chromatic dispersion (short link)         0.00 ps/nm
   Chromatic dispersion (long link)          0.00 ps/nm
   Differential group delay                  9.31 ps
   SOPMD                                     0.00 ps^2
   Polarization dependent loss               0.40 dB
   Received OSNR estimate                   35.10 dB
   Received ESNR estimate                   17.50 dB
   Carrier frequency offset                  0.00 MHz
   Error vector magnitude                  100.00 %
   SOP rate of change                        0.00 krad/s
   Laser temperature                        59.54 C
   Laser frequency                      193100.00 GHz
```

- BER: Bit Error Rate
- FEC: Forward Error Correction
- OSNR: Optical Signal to Noise Ratio
- ESNR: Electrical Signal to Noise Ratio
- SOP: State of Polarization
- SOPMD: State of Polarization Mode Dispersion

- **show interfaces hardware**

  The **show interfaces hardware** command displays the speed capabilities of a module. A 400GBASE-ZR supports 4x100G mode if the 100G-2/full speed is supported.

  Example

```
switch# show interfaces ethernet 4/1 hardware
*  = Requires speed group setting change
Ethernet4/1
  Model: DCS-7280PR3K-24
  Type: 400GBASE-ZR
  Speed/duplex: 100G-2/full,400G-8/full(default)
  Flowcontrol: rx-(off,on),tx-(off)
  Modulation: 16QAM
  Error correction: C-FEC(16QAM(default)), O-FEC(16QAM)
```

  The **show interface status** command displays the ethernet interfaces:

```
switch#show interfaces ethernet 4/1,4/3,4/5,4/7 status
Port      Name  Status    Vlan  Duplex Speed  Type      Flags Encapsulation
Et4/1           connected 1     full   100G   400GBASE-ZR
Et4/3           connected 1     full   100G   400GBASE-ZR
Et4/5           connected 1     full   100G   400GBASE-ZR
Et4/7           connected 1     full   100G   400GBASE-ZR
```

- **show transceiver status interface**

The **show transceiver status interface** command displays the most important alarms, faults and interface status and the existence of 4 host interfaces.

Example

For 400GBASE-ZR modules, media and host side coherent alarms, host-side pre-FEC BER, defined in the Coherent CMIS and post-FEC BER have been added to the command's output:

```
switch# show transceiver status interface Ethernet 4/1,4/3,4/5,4/7
                               Current State        Changes       Last Change
                               -------------        -------       -----------
Port 4
  Transceiver                400GBASE-ZR             3    2:32:34 ago
  Transceiver SN             204653947
  Presence                   present
  Adapters                   none
  Bad EEPROM checksums                                    0    never
  Resets                                                  0    2:32:41 ago
  Interrupts                                              0    never
  Data path firmware fault   ok                          0    never
  Module firmware fault      ok                          0    never
  Temperature high alarm     ok                          0    never
  Temperature high warn      ok                          0    never
  Temperature low alarm      ok                          0    never
  Temperature low warn       ok                          0    never
  Voltage high alarm         ok                          0    never
  Voltage high warn          ok                          0    never
  Voltage low alarm          ok                          2    2:32:19 ago
  Voltage low warn           ok                          2    2:32:19 ago
  Module state               ready                       6    0:02:21 ago
  Data path 1 state          activated                   12         0:01:33 ago
  Data path 2 state          activated                   12    0:01:33 ago
  Data path 3 state          activated                   12    0:01:33 ago
  Data path 4 state          activated                   12    0:01:33 ago
  Data path 5 state          activated                   12    0:01:33 ago
  Data path 6 state          activated                   12    0:01:33 ago
  Data path 7 state          activated                   12    0:01:33 ago
  Data path 8 state          activated                   12    0:01:33 ago
  RX LOS                     ok                           4    0:01:33 ago
  TX fault                   ok                           0    never
  RX CDR LOL                 ok                           4    0:01:31 ago
  TX power high alarm        ok                           0    never
  TX power high warn         ok                           4    0:02:12 ago
  TX power low alarm         ok                           4    0:02:21 ago
  TX power low warn          ok                           6    0:02:12 ago
  TX bias high alarm         ok                           0    never
  TX bias high warn          ok                           0    never
  TX bias low alarm          ok                           0    never
  TX bias low warn           ok                           0    never
  RX power high alarm        ok                           0    never
  RX power high warn         ok                           0    never
  RX power low alarm         ok                           0    never
  RX power low warn          ok                           0    never
  TX loss of alignment       ok                           0    never
  TX out of alignment        ok                           0    never
  TX clock monitor unit LOL  ok                           0    never
  TX reference clock LOL     ok                           0    never
  TX deskew LOL              ok                           0    never
  TX FIFO error              ok                           0    never
  RX demodulator LOL         ok                           4    0:01:30 ago
  RX CD compensation LOL     ok                           4    0:01:30 ago
  RX loss of alignment       ok                           0    never
  RX out of alignment        ok                           0    never
  RX deskew LOL              ok                           0    never
  RX FIFO error              ok                           0    never
  RX FEC excessive degrade   ok                           0    never
  RX FEC detected degrade    ok                           0    never
  Freq tuning in progress    idle                         8    0:01:32 ago
  Freq tuning busy           ok                           0    never
  Freq tuning invalid channel      ok                          0    never
  Freq tuning completed      no                           8    0:01:30 ago
Ethernet4/1
  Operational speed          100Gbps
  Pre-FEC bit error rate     0.00e+00
  Post-FEC errored frames ratio 0.00e+00
  TX LOS
 Host lane 1              ok                       2    0:02:21 ago
```

```
 Host lane 2              ok                       2    0:02:21 ago
  TX CDR LOL
 Host lane 1              ok                       2    0:02:21 ago
 Host lane 2              ok                       0    never
  TX adaptive input EQ fault
 Host lane 1              ok                       2    0:02:21 ago
 Host lane 2              ok                       2    0:02:21 ago
Ethernet4/3
  Operational speed          100Gbps
  Pre-FEC bit error rate     0.00e+00
  Post-FEC errored frames ratio 0.00e+00
  TX LOS
 Host lane 3              ok                       2    0:02:21 ago
 Host lane 4              ok                       2    0:02:21 ago
  TX CDR LOL
 Host lane 3              ok                       2    0:02:21 ago
 Host lane 4              ok                       2    0:02:21 ago
  TX adaptive input EQ fault
 Host lane 3              ok                       2    0:02:21 ago
 Host lane 4              ok                       2    0:02:21 ago
Ethernet4/5
  Operational speed          100Gbps
  Pre-FEC bit error rate     0.00e+00
  Post-FEC errored frames ratio 0.00e+00
  TX LOS
 Host lane 5              ok                       2    0:02:21 ago
 Host lane 6              ok                       2    0:02:21 ago
  TX CDR LOL
 Host lane 5              ok                       0    never
 Host lane 6              ok                       2    0:02:21 ago
  TX adaptive input EQ fault
 Host lane 5              ok                       0    never
 Host lane 6              ok                       2    0:02:21 ago
Ethernet4/7
  Operational speed          100Gbps
  Pre-FEC bit error rate     0.00e+00
  Post-FEC errored frames ratio 0.00e+00
  TX LOS
 Host lane 7              ok                       2    0:02:21 ago
 Host lane 8              ok                       2    0:02:21 ago
  TX CDR LOL
 Host lane 7              ok                       2    0:02:21 ago
 Host lane 8              ok                       2    0:02:21 ago
  TX adaptive input EQ fault
 Host lane 7              ok                       2    0:02:21 ago
 Host lane 8              ok                       0    never
```

- **Wavelength/Frequency and Output Power Status**

  The **show interface transceiver hardware** command displays the configured and programmed wavelength and output power. It takes a short time for the configured wavelength or output power to be programmed into the transceiver. The configured wavelength/output power and programmed wavelength/output power should not differ for extended periods of time.

```
switch# show interface Ethernet23/1 trans hardware
Name: Et23/1
Media type: 400GBASE-ZR
Maximum module power (W): 20.0
Maximum slot power (W): 20.0
Configured frequency (GHz): 193100.0
Computed wavelength (nm): 1552.52
Operational frequency (GHz): 193,100.0
Operational wavelength (nm): 1552.52
Configured TX power (dBm): -10.0
Operational TX power (dBm): -10.0
```

Note: Configured transmit output power and operational transmit output power are only displayed if configurable output power is supported by the module.

11.1.3.2.1.3Troubleshooting

1. Check transceiver type – make sure it is 400GBASE-ZR
2. Check peer transceiver – make sure it is also 400GBASE-ZR.
3. Check that channel/frequency is configured and interfaces are not in errdisabled state
4. Check that the selected frequency is matching on both sides of the optical link
5. Check that transmit output is not disabled
6. If Tx power is not configured, check that it is in the range of -6 dB to -12 dB.
7. If Tx power is configured, check that matches its configured values on both sides of the optical link.
8. Check Rx power. The best performance of an optical link is achieved when the received power is -10dB.
9. Collect the output of CLI commands listed in the “Show commands” section before making a request for support from the development team.
10. Check that pre-FEC BER (‘show transceiver dom’ command) is in the correctable range (less than 1e-2).
11. ‘show transceiver status interface’ - check that module state is ‘Ready’, datapath state is ‘Activated’. Check for possible alarms and faults

Link Issues

If a link has issues, the following commands and files are useful for debugging and Arista TAC.

- **`show interfaces <interfaceName> phy detail`**
- **`show interfaces <interfaceName> transceiver detail`**
- **`show idprom transceiver <interfaceName> ext`**
- files in **`/var/log/agents/*`**
- files in **`/var/log/qt/*`**
- **`/var/log/messages*`**

11.1.3.2.1.4Limitations

400GBASE-ZR transceivers are relatively new. The modules could be Arista branded or 3rd party, from multiple vendors. For 3rd party optics, some optional properties (Tx output power, support of 75GHz grid) or DOM properties in the VDM pages, may not be implemented.

11.1.3.3 MXP Ports

MXP ports provide embedded optics that operate in one of three modes: 10GbE (12 ports), 40GbE (3 ports), and 100GbE (1 port). Each mode requires a specified cable is implemented through configuration commands. MXP ports utilize multi-mode fiber to provide support over 150 meters.

- 100GbE mode requires an MTP-24 to MTP-24 cable, which uses 20 of 24 fibers to carry 100Gbe across 10 send and 10 receive channels. When connecting two 100GbE MXP ports, the TX lanes must be crossed with the RX lanes.
- 40GbE mode requires an MTP cable that provides a split into three MTP-12 ends. The cable splits the MXP port into three MTP-12 ends, each compatible with standards based 40GBASE-SR4 ports over OM3 or OM4 fiber up to 100m or 150m.
- 10GbE mode requires an MTP cable that provides a split into 12x10G with LC connectors to adapt the MXP port into 12x10GbE. The cable splits the MXP port into twelve LC ends for using SR or SRL optics over multimode OM3/OM4 cables.

11.1.4 Interfaces

Arista switches provide two physical interface types that receive, process, and transmit Ethernet frames: Ethernet interfaces and Management interfaces.

Each Ethernet interface is assigned a 48-bit MAC address and communicates with other interfaces by exchanging data packets. Each packet contains the MAC address of its source and destination interface.

Ethernet interfaces establish link level connections by exchanging packets. Interfaces do not typically accept packets with a destination address of a different interface.

Ethernet data packets are frames. A frame begins with preamble and start fields, followed by an Ethernet header that includes source and destination MAC addresses. The middle section contains payload data, including headers for other protocols carried in the frame. The frame ends with a 32-bit Cyclic Redundancy Check (CRC) field that interfaces use to detect data corrupted during transmission.

11.1.4.1 Ethernet Interfaces

Ethernet speed and duplex configuration options depend on the media type of the interface:

- 40G, OSFP, QSFP+, QSFP-DD, and QSFP200: Default operation is as four 10G ports. **`Speed`** command options support configuration as a single 40G port.
- 1000BASE-T / 2.5GBASE-T / 5GBASE-T / 10GBASE-T (copper): Default configuration enables the **Clause 28 auto-negotiation** for the port to negotiate the speed based on peer capabilities. Depending on individual SKU capabilities, these ports support 10M/100M/1G/2.5G/5G and 10G rates and half / full-duplex mode of operation. The Speed auto SPEED command limits the port advertisements only to a specific speed. The **`SPEED`** commands disables the Clause 28 auto-negotiation and uses the specified speed as the forced speed setting.**`sa`** commands.
- 10GBASE-T (SFP+): Port operates as a single 10G port. **`speed`** commands do not affect configuration.
- 100G CFP2: Default operation is 100G. It cannot be split, and its speed cannot be changed.
- 100G MXP: Operates as three 40G ports on the 7050 platform. On the 7050 platforms, available speed/duplex settings are three 40G ports or twelve 10G ports. Adjustments are made with **`speed`** commands.
- 100G QSFP100: Available speeds are transceiver-dependent. The QSFP100 transceiver supports a single 100G port, four 25G ports, or two 50G ports; the QSFP+ transceiver supports one 40G port or four 10G ports; the CWDM transceiver supports all five configurations. Adjustments are made using **`speed`** commands.
- The **SFP1000BASE-T** transceivers advertise one speed at a time only. Hence, the desired speed and negotiation must be configured explicitly using the following commands:
  - speed auto: auto-negotiated 1Gbps (this is because no speed is specified and we are defaulting to advertise 1G).
  - speed auto 1Gfull / speed auto: auto-negotiated 1Gbps (note that per BASE-T standard, 1G must be negotiated).
  - speed auto 100full: auto-negotiated 100Mbps.
  - speed 100full: non-negotiated / forced 100Mbps.
- The **SFP10GBASE-Ts** transceivers advertise one speed at a time only, similar to SFP1000BASE-Ts, unlike native BASE-T ports. The **`no and default speed`** commands configure the PHY to advertise only 10G.

For information relating to transceivers, see Transceivers.

11.1.4.2 Subinterfaces

Subinterfaces divide a single ethernet or port channel interface into multiple logical L3 interfaces based on the 802.1q tag (VLAN ID) of incoming traffic. Subinterfaces are commonly used in the L2/L3 boundary device, but they can also be used to isolate traffic with 802.1q tags between L3 peers by assigning each subinterface to a different VRF.

While subinterfaces can be configured on a port channel interface (the virtual interface associated with a port channel), the following restrictions apply:

- An L3 interface with subinterfaces configured on it should not be made a member of a port channel.
- An interface that is a member of a port channel should not have subinterfaces configured on it.
- A subinterface cannot be made a member of a port channel.

Subinterfaces on multiple ports can be assigned the same ***VLAN ID***, but there is no bridging between subinterfaces (or between subinterfaces and SVIs), and each subinterface is considered to be in a separate bridge domain.

The following features are supported on subinterfaces:

- Unicast and multicast routing
- BGP, OSPF, ISIS, PIM
- ACL
- VRF
- VRRP
- SNMP
- Subinterface counters (on some platforms)
- VXLAN (on some platforms)
- MPLS (on some platforms)
- GRE (on some platforms)
- PBR (on some platforms)
- QoS (on some platforms)
- Inheriting QoS settings (trust mode and default DSCP) from the parent interface
- Inheriting MTU setting from parent interface

The following are not supported on subinterfaces:

- Per-subinterface MTU setting
- Per-subinterface SFLOW settings
- Per-subinterface mirroring settings

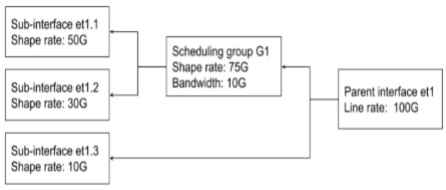
11.1.4.2.1 Shared Shaper across Multiple Subinterfaces

Sub-interfaces can be grouped into logical units called scheduling groups, which are shaped as a single unit. Each scheduling group may be assigned a scheduling policy which defines a shape rate in kbps and optionally a guaranteed bandwidth, also in kbps.

The guaranteed bandwidth is used if the sum of the shape rates of all scheduling groups and sub-interfaces that are not part of groups for a parent interface exceeds the available bandwidth. Each sub-interface within that scheduling group may have its own independent shape rate which are applied in a hierarchical manner.

Adding a sub-interface to a scheduling group results in allocation of dedicated Virtual Output Queues (VOQ) for the sub-interface.

11.1.4.2.1.1 Configuring Shared Shaper



Scheduling groups and scheduling policies are configured in the **qos scheduling** CLI mode.

1. First, configure one or more sub-interfaces.
2. Then, create a scheduling policy with the desired shape rate and optional guaranteed bandwidth:

```
switch(config)# qos scheduling
switch(config-qos-scheduling)# scheduling policy P1
switch(config-qos-scheduling-policy-P1)# shape rate 75000000
```

```
switch(config-qos-scheduling-policy-P1)# bandwidth guaranteed 10000000
```

3. The shape rate and guaranteed bandwidth may also be defined as percents of the next highest level of the hierarchy (in this case line rate):

```
switch(config-qos-scheduling-policy-P1)# shape rate 75 percent
switch(config-qos-scheduling-policy-P1)# bandwidth guaranteed percent 10
```

4. Create the scheduling group on the parent interface:

```
switch(config)# qos scheduling
switch(config-qos-scheduling)# interface et1
switch(config-qos-scheduling-intf-Ethernet1)# scheduling group G1
```

5. Assign a policy to the scheduling group:

```
switch(config-qos-scheduling-intf-Ethernet1-group-G1)# policy P1
```

6. Assign members to the scheduling group (members may be put all on one line or on separate lines):

```
switch(config-qos-scheduling-intf-Ethernet1-group-G1)# members et1.1 et1.2
```

11.1.4.2.1.2Show Commands

- QoS configuration on one or more scheduling groups. Both group name and parent interface name are optional, in which case all groups are displayed as shown in the example below:

  Example

```
switch# show qos scheduling group G1 Ethernet1
Interface: Et1
Scheduling Group Name: G1
Bandwidth: 10.1 / 10.0 (Gbps)
Shape Rate: 75.2 / 75.0 (Gbps)

Member          Bandwidth         Shape Rate
                  (units)           (units)
------       -------------  ------------------
Et1.1           - /  - (-)  50.1 / 50.0 (Gbps)
Et1.2           - /  - (-)  30.1 / 30.0 (Gbps)
```

- QOS configuration on a parent interface will show scheduling groups configured on the parent interface.

  Example

```
switch# show qos interface Ethernet1
Ethernet1:

   Trust Mode: DSCP
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

Scheduling Group        Bandwidth           Shape Rate
                          (units)             (units)
----------------    ------------------  ------------------
G1                  10.1 / 10.0 (Gbps)  75.1 / 75.0 (Gbps)
```

- The entire scheduling hierarchy for a subinterface can be displayed. This displays the shape rate for each transmit queue and then the shape rate and guaranteed bandwidth at every other level of the hierarchy.

Example

```
switch# show qos scheduling hierarchy Ethernet1.1
Interface     Hierarchy Level             Bandwidth             Shape Rate
                                          (units)               (units)
---------    -------------------------  -------------------    ----------------
Et1/1.100     tx queue (0)                  - /    -            100 / 100 (Mbps)
              tx queue (2)                  - /    -            200 / 200 (Mbps)
              subinterface (Et1/1.1)      100 /  100 (Mbps)     500 / 500 (Mbps)
              group (G1)                  10 /   10 (Gbps)       75 /  75 (Gbps)
              parent (Et1/1) (100 Gbps)    - /    -              - /   -  ( - )
```

11.1.4.3 Agile Ports

Agile Ports are a feature of the 7150S Series that allows the user to configure adjacent blocks of 4 x SFP + interfaces as a single 40G link. The set of interfaces that can be combined to form a higher speed port is restricted by the hardware configuration. Only interfaces that pass through a common PHY component can be combined. One interface within a combinable set is designated as the primary port.

When the primary interface is configured as a higher speed port, all configuration statements are performed on that interface. All other interfaces in the set are subsumed and not individually configurable when the primary interface is configured as the higher speed port. This feature allows the 7150S-24 to behave as a 4x40G switch (using 16 SFP+) and the remaining SFP+ provide 8 x 10Gports. On the 7150S-52 this allows up to 13x 40G (all 52 ports grouped as 40G) and on the 7150S-64 Agile Ports allows the switch to be deployed with up to 16 native 40G interfaces - 4 are QSFP+ and the remaining 12 as 4xSFP+ groups.

11.1.4.4 Management Interfaces

The management interface is a Layer 3 host port that is typically connected to a PC for performing out of band switch management tasks. Each switch has one or two management interfaces. Only one port needs to manage the switch; the second port, when available, provides redundancy.

Management interfaces are 10/100/1000 BASE-T interfaces. By default, auto-negotiation is enabled on management interfaces. All combinations of speed 10/100/1000 and full or half duplex is enforceable on these interfaces through **speed** commands.

Management ports are enabled by default. The switch cannot route packets between management ports and network (Ethernet interface) ports because they are in separate routing domains. When the PC is multiple hops from the management port, packet exchanges through Layer 3 devices between the management port and PC may require the enabling of routing protocols.

The Ethernet management ports are accessed remotely over a common network or locally through a directly connected PC. An IP address and static route to the default gateway must be configured to access the switch through a remote connection.

11.1.4.5 Tunable SFP

Tuning of DWDM 10G SFP+ transceivers (10GBASE-DWDM) includes:

- Tuning transceiver wavelength/frequency by channel number.
- Showing wavelengths/frequencies for specified channels supported by the transceiver.
- Showing current wavelength/frequency settings of the transceiver interface.

For information relating to tuning the transceiver wavelength/frequency by channel number, refer to the command `transceiver channel`. To show the current wavelength/frequency settings for specified channels, refer to the command `show interfaces transceiver channels`. To show the current wavelength/frequency settings of an interface, refer to the command `show interfaces transceiver hardware`.

11.1.5 MRU Enforcement

Maximum Receive Unit (MRU) enforcement provides the ability to drop frames that exceed a configured threshold on the ingress interface.

11.1.5.1 Configuring MRU Enforcement

MRU is configurable per-interface, and can be configured on Ethernet and Port-Channel interfaces. Frames with size greater than the configured MRU value drop on the ingress, and do not forward to the destined egress interface. MRU enforcement happens at the Ethernet interface and applies to both L2 and L3 traffic. Note that FCS (frame check sequence) is included in the frame size.

Ethernet Interface

```
switch(config)# interface ethernet 1
switch(config)# l2 mru 9000
```

Frames with size greater than ***9000*** bytes ingressing into ***Ethernet1*** are dropped.

Port-Channel Interface

Members of a Port-Channel interface inherit the Port-Channel interface's MRU value. Members' MRU configured in the ***Ethernet interface configuration mode*** has no effects.

```
switch(config)# interface ethernet 1 - 4
switch(config-if-Et1-4)# channel-group 10 mode active
switch(config)# interface port-Channel 10
switch(config-if-Po10)#
```

Frames with size greater than ***9000*** bytes ingressing into ***Ethernet1***, ***2***, ***3***, and ***4*** are dropped.

Sub-interfaces

MRU configured on an Ethernet interface or Port-Channel are applied to all of its sub-interfaces.

Default Behaviours

By default, MRU is set to the maximum value.

- For DCS-7280R3 and DCS-7500R3 series
  - The maximum MRU is ***10240*** bytes
- For other supported platforms
  - In TapAgg mode, the maximum MRU is ***10240*** bytes.
  - In non TapAgg mode, the maximum MRU is ***10200*** bytes.

11.1.5.2 MRU Enforcement Limitations

- The number of features for which the hardware counters can be enabled simultaneously is limited by the availability of counter hardware resources in the system. When the configured hardware features exceed the available counter resources, not all counters for all features are available.
  - At lower scales of the feature key (for example, the number of interfaces in the case of subinterface/ SVI counters), you can typically concurrently support up to six (6) ingress features and three (3) egress features. As scales grow beyond table sizes, single features use multiple tables.
- Route counters do not support the counting of host routes or ALPM routes.
- Packets sent from the CPU are not included in egress counters.

11.1.5.3 MRU Enforcement Show commands

The MRU on an interface is found in the show interface output.

```
switch(config)# show interface ethernet 1
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 444c.a8b7.1ed8 (bia 444c.a8b7.1ed8)
  Member of Port-Channel10
  Ethernet MTU 10178 bytes, Ethernet MRU 1500 bytes, BW 10000000 kbit
  Full-duplex, 10Gb/s, auto negotiation: off, uni-link: disabled.
```

Counter

MRU dropped packets are counted per-chip.

The Ethernet interfaces corresponding chip are found in the **show platform fap mapping** output.

```
switch(config)# show platform fap mapping interface Ethernet 1
Jericho0 (FapId: 0  BaseSystemCoreId: 0)
Port             SysPhyPort   Voq   Core  FapPort  OtmPort BaseQPair QPairs Xlge NifPort
-------------------------------------------------------------------------------------------
Ethernet1        100          2608  0      2       0        0         8     8     33
```

Reassembly Errors use per-chip counters from the show hardware counter drop output.

```
switch(config)# show hardware counter drop
Type  Chip       CounterName       : Count    : First Occurrence    : Last Occurrence
-------------------------------------------------------------------------------------------
A     Jericho0   ReassemblyErrors  : 12132989 : 2020-09-22 17:05:45 : 2020-09-22 17:22:40
```

11.1.6 Ethernet Configuration Procedures

These sections describe Ethernet and Management interface configuration procedures:

- Physical Interface Configuration Modes
- Assigning a MAC Address to an Interface
- Port Groups (QSFP+ and SFP+ Interface Selection)
- Referencing Modular Ports
- Referencing Multi-lane Ports
- Hitless Speed Change with Dynamic Logical Ports
- QSFP+ Ethernet Port Configuration
- QSFP100 Ethernet Port Configuration
- CFP2 Ethernet Port Configuration
- Default QSFP Mode Support
- MXP Ethernet Port Configuration
- Port Speed Capabilities
- Agile Ports
- Subinterface Configuration
- Maximum Latency Tail-drop Thresholds
- Autonegotiated Settings
- Displaying Ethernet Port Properties
- Ingress Counters
- Configuring Ingress Traffic-Class Counters
- Hardware Counter Support
- Configuring Power over Ethernet (PoE)
- Configuring Link Fault Signaling
- Configuring Hardware TCAM

- CPU Traffic Policy
- TCAM Profile for Configurable Port Qualifier Sizing

11.1.6.1 Physical Interface Configuration Modes

The switch provides two configuration modes for modifying Ethernet parameters:

- ***Interface-Ethernet*** mode configures parameters for specified Ethernet interfaces.
- ***Interface-Management*** mode configures parameters for specified management Ethernet interfaces.

Physical interfaces cannot be created or removed.

Multiple interfaces can be simultaneously configured. Commands are available for configuring Ethernet specific, layer 2, Layer 3, and application layer parameters. Commands that modify protocol specific settings in Ethernet configuration mode are listed in the protocol chapters.

- The `interface ethernet` command places the switch in ***Ethernet-interface*** configuration mode.
- The `interface management` command places the switch in ***management*** configuration mode.

Examples

- This command places the switch in ***Ethernet-interface*** mode for ***interface ethernet 5-7*** and ***10***.

```
switch(config)# interface ethernet 5-7,10
switch(config-if-Et5-7,10)#
```

- This command places the switch in ***management-interface*** mode for ***management interface 1***.

```
switch(config)# interface management 1
switch(config-if-Ma1)#
```

11.1.6.2 Assigning a MAC Address to an Interface

Ethernet and Management interfaces are assigned a MAC address when manufactured. This address is the burn-in address. The `mac-address` command assigns a MAC address to the configuration mode interface in place of the burn-in address. The **`no mac-address`** command reverts the interfaces current MAC address to its burn-in address.

Examples

- This command assigns the MAC address of ***001c.2804.17e1*** to ***Ethernet interface 7***.

```
switch(config-if-Et7)# mac-address 001c.2804.17e1
```

- This command displays the MAC address of ***interface ethernet 7***. The active MAC address is ***001c.2804.17e1***. The burn-in address is ***001c.7312.02e2***.

```
switch(config-if-Et7)# show interface ethernet 7
Ethernet7 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2804.17e1 (bia 001c.7312.02e2)
  Description: b.e45
switch(config-if-Et7)#
```

11.1.6.3 Port Groups (QSFP+ and SFP+ Interface Selection)

Several of Arista's fixed switches limit the number of 10G data lanes in operation through the use of port groups. A port group is a set of interfaces that can be configured as four SFP+ interfaces or a single QSFP + interface. When configured in SFP+ mode, the port group enables four standalone 10GbE interfaces using SFP+ optics. When configured in QSFP+ mode, the port group enables a single QSFP+ interface (in addition to the dedicated QSFP+ ports), which can operate as a single 40GbE port, or as four 10GbE ports with the appropriate breakout cabling.

Hardware port groups are used on the following systems:

- DCS-7050Q-16
- DCS-7050QX-32S

Use the `hardware port-group` command to select the interface mode for the specified port group.

Note: The **`hardware port-group`** command restarts the forwarding agent, which disrupts traffic on all switch ports.

Example

These commands configure the DCS-7050-Q16 switch to enable four SFP+ interfaces and one extra QSFP+ interface by enabling the SFP+ interfaces in ***port-group 1*** and the QSFP+ interface in ***port-group 2***.

```
switch(config)# hardware port-group 1 select Et17-20
switch(config)# hardware port-group 2 select Et16/1-4
```

The show hardware port-group command displays the status of ports in the port groups.

Example

This command displays the status of the flexible ports within the two port groups on a DCS-7050Q-16 switch.

```
switch# show hardware port-group

Portgroup: 1    Active Ports: Et17-20
Port            State
----------------------------------------
Ethernet17      Active
Ethernet18      Active
Ethernet19      Active
Ethernet20      Active
Ethernet15/1    ErrDisabled
Ethernet15/2    ErrDisabled
Ethernet15/3    ErrDisabled
Ethernet15/4    ErrDisabled

Portgroup: 2    Active Ports: Et16/1-4
Port            State
----------------------------------------
Ethernet16/1Active
Ethernet16/2Active
Ethernet16/3Active
Ethernet16/4Active
Ethernet21ErrDisabled
Ethernet22ErrDisabled
Ethernet23ErrDisabled
Ethernet24ErrDisabled
```

11.1.6.3.1 DCS-7050Q-16

The DCS-7050Q-16 has 14 dedicated QSFP+ ports, plus two port groups. The port groups support either two additional QSFP+ ports or eight SFP+ ports as shown in DCS-7050Q-16 Port Groups.

Table 56: DCS-7050Q-16 Port Groups

Port Group 1 **Active Interface(s)**		**Port Group 2** **Active Interface(s)**	
In SFP+ Mode	In QSFP+ Mode (Default)	In SFP+ Mode	In QSFP+ Mode (Default)
Et17-20 (four SFP+ ports)	Et15/1-4 (one QSFP+ port)	Et21-24 (four SFP+ ports)	Et16/1-4 (one QSFP+ port)

11.1.6.3.2 DCS-7050QX-32S

The DCS-7050QX-32S has 31 dedicated QSFP+ ports, plus one port group. The port group supports either one additional QSFP+ port or four SFP+ ports as shown in DCS-7050QX-32S Port Groups.

Table 57: DCS-7050QX-32S Port Groups

Port Group 1 **Active Interface(s)**	
In SFP+ Mode	In QSFP+ Mode (Default)
Et1-4 (four SFP+ ports)	Et5/1-4 (one QSFP+ port)

11.1.6.4 Referencing Modular Ports

Arista modular switches provide port access through installed line cards. The maximum number of line cards on a modular switch varies with the switch series and model.

Several CLI commands modify modular parameters for all ports on a specified line card or controlled by a specified chip. This manual uses these conventions to reference modular components:

- ***card_x*** refers to a line card.
- ***module_y*** refers to a QSFP+ module.
- ***port_z*** refers to a line card or module port.

Commands that display Ethernet port status use the following conventions:

- **SFP ports: *card_x/port_z*** to label the line card-port location of modular ports.
- **QSFP ports: *card_x/module_y/port_z*** to label the line card-port location of modular ports.

QSFP+ Ethernet Port Configuration describe QSFP+ module usage.

Example

This command displays the status of interfaces ***1*** to ***9*** on ***line card 4***:

```
switch# show interface ethernet 4/1-9 status
Port      Name              Status       Vlan         Duplex  Speed Type
Et4/1                       connected    1              full    10G Not Present
Et4/2                       connected    1              full    10G Not Present
Et4/3                       connected    1              full    10G Not Present
Et4/4                       connected    1              full    10G Not Present
Et4/5                       connected    1              full    10G Not Present
Et4/6                       connected    1              full    10G Not Present
Et4/7                       connected    1              full    10G Not Present
```

```
Et4/8                         connected    1            full    10G Not Present
Et4/9                         connected    1            full    10G Not Present switch>
```

11.1.6.5 Referencing Multi-lane Ports

EOS supports two types of Ethernet ports:

- single-lane (also called fixed-lane)
- multi-lane (also called flexible-lane)

Single-lane (or fixed-lane) ports are always modeled as a single interface within EOS. While the speed of the interface may be configurable, the physical port can never be broken out into multiple lower-speed interfaces. Single-lane ports use the following naming scheme:

- Ethernet <port #> (for fixed switches)
- Ethernet <module #>/<port #> (for modular switches)

Multi-lane (or flexible lane) ports are made up of multiple parallel lanes, each served by its own laser. Multi-lane ports can be configured to combine the lanes and operate as a single native high-speed interface (a 40GbE or 100GbE interface), or to operate each lower-speed interface independently (four 10GbE or 25GbE interfaces). Multi-lane ports use the following naming scheme:

- Ethernet ***port*** **#/*lane*** **#** (for fixed switches)

Ethernet ***module*** **#/*port*** **#/*lane*** **#** (for modular switches)

The operational state displayed for each lane of a multi-lane port is determined by the configuration applied to the primary lane(s), as shown in Lane States. When broken out into multiple lower-speed interfaces, all lanes will be active in parallel, and each will display its operational state as **connected** or **not connected**. In high-speed mode, only the primary lane(s) will be displayed as active, with the remaining lanes showing as **errdisabled**. The exception is the CFP2 module: when it is configured as a single 100GbE port, the primary lane is displayed as active in the CLI while the other lanes are hidden.

Table 58: Lane States

Parent Port Configured Mode	Primary Lane(s)	Secondary Lanes
single high-speed interface	active (connected/not connected)	inactive (errdisabled)
multi-interface breakout	active (connected/not connected)	active (connected/not connected)

A multi-lane port is configured as a single high-speed interface or multiple breakout interfaces by using the **speed** command on the primary lane(s) of the port. For specific configuration instructions and details regarding the primary lane(s) of a specific interface, refer to the configuration section for the appropriate interface type:

- QSFP+ Ethernet Port Configuration
- QSFP100 Ethernet Port Configuration
- CFP2 Ethernet Port Configuration
- MXP Ethernet Port Configuration

Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280CR3, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series

platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

11.1.6.6 Hitless Speed Change with Dynamic Logical Ports

Higher speed ports on several switches can be broken out into multiple interfaces that can be configured at lower speeds. Each of these interfaces is called a Logical Port (LP). Switches such as the DCS-7260CX3-64, allocate and deallocate the logical ports dynamically to optimize hardware resources. Care must be taken for hitless speed changes, therefore, as a change of speed on a Dynamic Logical Port (DLP) may impact the status of one or more related DLP(s).

Checking Status of DLPs

The following example is applicable to a switch that supports a breakout of a 100G physical port to 4x 25G DLPs.

The command displays the status of Et 45/1-4, when a 100G physical port is broken out to 4x 25G DLPs.

```
switch(config)# show interfaces ethernet 45/1-4 status
Port    Name   Status    Vlan  Duplex Speed  Type      Flags
Et45/1         notconnect   1     full   25G Not Present
Et45/2         notconnect   1     full   25G Not Present
Et45/3         notconnect   1     full   25G Not Present
Et45/4         notconnect   1     full   25G Not Present
```

The following example displays the inactive interfaces (DLPs) when the 100G physical port is configured for the 100G speed.

```
switch(config)# show interfaces ethernet 45/1-4 status
Port    Name   Status    Vlan  Duplex Speed  Type      Flags
Et45/1         notconnect 1     full   100G   Not Present
Et45/2         inactive   1     unconf unconf Not Present
Et45/3         inactive   1     unconf unconf Not Present
Et45/4         inactive   1     unconf unconf Not Present
```

By default, inactive interfaces will not be displayed by this command. To enable showing them, use the following command.

```
switch(config)# [no] service interface inactive expose
```

Only systems without DLP allocation have enough logical ports for every interface to be active simultaneously.

The following example displays the logical port pool information and current logical port allocation status for a fully loaded switch.

```
switch> show hardware logical-port pool status
Pool Max Free Configured Interfaces
---- --- ---- ---------- ------------------------------------------------------
   1  18    2         16 Et2/5/1-2/8/4,3/5/1-3/8/4,4/5/1-4/8/4,5/9/1-5/12/4
   2  18    2         16 Et2/1/1-2/4/4,3/9/1-3/12/4,4/1/1-4/4/4,5/5/1-5/8/4
   3  18    2         16 Et2/9/1-2/12/4,3/1/1-3/4/4,4/9/1-4/12/4,5/1/1-5/4/4
   4  18    2         16 Et2/13/1-2/16/4,3/13/1-3/16/4,4/13/1-4/16/4,5/13/1-5/16/4
   5  18    2         16 Et6/13/1-6/16/4,7/13/1-7/16/4,8/13/1-8/16/4,9/13/1-9/16/4
   6  18    2         16 Et6/1/1-6/4/4,7/9/1-7/12/4,8/1/1-8/4/4,9/9/1-9/12/4
   7  18    2         16 Et6/5/1-6/8/4,7/1/1-7/4/4,8/9/1-8/12/4,9/1/1-9/4/4
   8  18    2         16 Et6/9/1-6/12/4,7/5/1-7/8/4,8/5/1-8/8/4,9/5/1-9/8/4
```

The following example displays the logical port pool information and current logical port allocation status for a switch with only Linecard ***3***, ***4***, ***5***, and ***7*** inserted.

```
switch> show hardware logical-port pool status
Pool Max Free Configured Interfaces
---- --- ---- ---------- ------------------------------------------------------
   1  18    8         10 Et3/5/1-3/8/4,4/2/1-8,5/3/1-8
   2  18   12          6 Et3/9/1-3/12/4,4/1/1-8,5/2/1-8
   3  18    8         10 Et3/1/1-3/4/4,4/3/1-8,5/1/1-8
   4  18    7         11 Et3/13/1-3/16/4,4/4/1-8,5/4/1-8
```

```
5  18   16          2 Et7/13/1-7/16/4
6  18   14          4 Et7/9/1-7/12/4
7  18   14          4 Et7/1/1-7/4/4
8  18   12          6 Et7/5/1-7/8/4
```

11.1.6.7 QSFP+ Ethernet Port Configuration

Each QSFP+ module contains four data lanes which can be used individually or combined to form a single, higher-speed interface. This allows a QSFP+ Ethernet port to be configured as a single 40GbE interface or as four 10GbE interfaces.

When the four lanes are combined to form a 40GbE interface, display commands will show ***lane /1*** as **connected** or **not connected**, and will show lanes ***/2*** through ***/4*** as **errdisabled**.

The following sections describe the configuration of QSFP+ ports.

11.1.6.7.1 Configuring a QSFP+ Module as a Single 40GbE Interface

To configure the port as a single 40GbE interface, combine the modules four data lanes by using the speed command (**speed forced 40g full**) on the ports /1 lane (the primary lane).

Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the QSFP+ Ethernet interface.

```
switch(config)# interface ethernet 5/1/1
```

2. Enter the **speed 40gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/1/1)# speed 40gfull
```

3. Use the show interfaces status command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)# show interfaces status
Port Name Status Vlan Duplex Speed Type Flags
Et1 connected 2 full 1G 10GBASE-T
    <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1 connected 1 full 40G 40GBASE-SR4
Et5/1/2 errdisabled 1 unconf unconf 40GBASE-SR4
Et5/1/3 errdisabled 1 unconf unconf 40GBASE-SR4
Et5/1/4 errdisabled 1 unconf unconf 40GBASE-SR4
     <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

11.1.6.7.2 Configuring a QSFP+ Module as Four 10GbE Interfaces

To configure the port as four 10GbE interfaces, use the speed command (**speed 10000full**) on the ***ports /1*** lane (the primary lane).

Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the QSFP+ Ethernet interface.

```
switch(config)# interface ethernet 5/1/1
```

2. Enter the `speed` **10000full** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)# speed 10000full
```

3. Use the `show interfaces status` command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)# show interfaces status
PortNameStatusVlanDuplexSpeedTypeFlags
Et1connected2full1G10GBASE-T
Et5/1/1connected1full10G40GBASE-SR4
Et5/1/2connected1full10G40GBASE-SR4
Et5/1/3connected1full10G40GBASE-SR4
Et5/1/4connected1full10G40GBASE-SR4
```

11.1.6.8 QSFP100 Ethernet Port Configuration

Each QSFP100 module contains four data lanes which can be used individually or combined to form a single, higher-speed interface. This allows a QSFP100 Ethernet port to be configured as a single 100GbE interface, a single 40GbE interface, or four 10GbE interfaces. The default mode is a single 100GbE interface.

The 7060X, 7260X and 7320X platforms also allow a QSFP100 port to be configured as two 50GbE interfaces or four 25GbE interfaces.

When the lanes are combined to form a higher-speed interface, display commands show the primary lane(s) as **connected** or **not connected**, and shows the other lanes as **errdisabled**.

The following sections describe the configuration of QSFP+ ports.

11.1.6.8.1 Configuring a QSFP100 Module as a Single 100GbE Interface

By default, the QSFP100 module operates as a single 100GbE interface; using the **default speed** or **no speed** command on the primary lane restores the default behavior.

To explicitly configure the port as a single 100GbE interface, combine the modules four data lanes by using the speed command (**speed 100gfull**) on the ports ***/1*** lane (the primary lane).

Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the QSFP100 Ethernet interface.

```
switch(config)# interface ethernet 5/1/1
```

2. Enter the `speed` **100gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/1/1)# speed 100gfull
```

3. Use the show interfaces status command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)# show interfaces status
Port    Name     Status    Vlan    Duplex     Speed    Type    Flags
Et1              connected    2     full      1G     10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

```
Et5/1/1          connected    1     full      100G   100GBASE-SR4
Et5/1/2          errdisabled  1    unconf    unconf  100GBASE-SR4
Et5/1/3          errdisabled  1    unconf    unconf  100GBASE-SR4
Et5/1/4          errdisabled  1    unconf     unconf 100GBASE-SR4
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

11.1.6.8.2 Configuring a QSFP100 Module as Two 50GbE Interfaces

To configure the port as a two 50GbE interfaces, configure the modules four data lanes by using the **speed** command (**speed 50gfull**) on the ports ***/1*** and ***/3*** lanes. This configuration is available on 7060X, 7260X and 7320X platforms.

Note:

Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280CR3, s7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the QSFP100 Ethernet interface.

```
switch(config)# interface ethernet 5/1/1
```

2. Enter the **speed 50gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/1/1)# speed 50gfull
```

3. Repeat the above steps for lane ***/3***.

```
switch(config-if-Et5/1/1)#interface ethernet 5/1/3
switch(config-if-Et5/1/3)#speed 50gfull
```

4. Use the show interfaces status command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)# show interfaces status
Port       Name     Status       Vlan      Duplex    Speed    Type       Flags
Et1                 connected     2          full     1G     10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1             connected     1          full     50G    100GBASE-SR4
Et5/1/2             errdisabled   1          unconf  unconf  100GBASE-SR4
Et5/1/3             connected     1          full     50G    100GBASE-SR4
Et5/1/4             errdisabled   1          unconf  unconf  100GBASE-SR4
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

11.1.6.8.3 Configuring a QSFP100 Module as a Single 40GbE Interface

To configure the port as a single 40GbE interface, combine the modules four data lanes by using the speed command (**speed 40gfull**) on the ports ***/1*** lane (the primary lane).

Note:

Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280CR3, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the QSFP100 Ethernet interface.

```
switch(config)# interface ethernet 5/1/1
```

2. Enter the **speed 40gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/1/1)# speed 40gfull
```

3. Use the show interfaces status command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)# show interfaces status
Port      Name      Status       Vlan     Duplex    Speed     Type      Flags
Et1                 connected    2        full       1G     10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1             connected    1        full       40G    100GBASE-SR4
Et5/1/2             errdisabled  1        unconf    unconf  100GBASE-SR4
Et5/1/3             errdisabled  1        unconf    unconf  100GBASE-SR4
Et5/1/4             errdisabled  1        unconf    unconf  100GBASE-SR4
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

11.1.6.8.4 Configuring a QSFP100 Module as Four 10GbE Interfaces

To configure the port as four 10GbE interfaces, use the speed command (**speed 10000full**) on the ports ***/1*** lane (the primary lane).

Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the QSFP100 Ethernet interface.

```
switch(config)# interface ethernet 5/1/1
```

2. Enter the **speed 10000full** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/1/1)# speed 10000full
```

3. Use the show interfaces status command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)# show interfaces status
Port       Name       Status       Vlan     Duplex    Speed      Type          Flags
Et1                   connected    2        full       1G     10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1               connected    1        full       10G    100GBASE-SR4
Et5/1/2               connected    1        full       10G    100GBASE-SR4
Et5/1/3               connected    1        full       10G    100GBASE-SR4
Et5/1/4               connected    1        full       10G    100GBASE-SR4
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

11.1.6.9 CFP2 Ethernet Port Configuration

Each CFP2 module contains ten data lanes. The configuration options available on the port depend on the optic inserted:

- CFP2-100G-LR4 optics operate only in 100GbE mode.
- CF2-100G-ER4 optics operate only 100GbE mode.
- CFP2-100G-XSR10 optics can be configured as a single 100GbE interface or as ten 10GbE interfaces.

When the port is configured as ten 10GbE interface, each lane is active and visible in CLI display commands. When the lanes are combined to form a single 100GbE interface, display commands will show the primary lane as **connected** or **not connected**; all other lanes will be hidden.

The following sections describe the configuration of CFP2 ports.

11.1.6.9.1 Configuring a CFP2 Module a as a Single100GbE Interface

To configure the port as a single 100GbE interface (the default configuration), combine the modules ten data lanes by using the speed command (**speed 100gfull**) on the ports ***/1*** lane (the primary lane).

This configuration is available for all pluggable optics.

Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280CR3, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for ***lane /1*** of the ***CFP2*** Ethernet interface.

```
switch(config)# interface ethernet 5/1/1
```

2. Enter the **speed 100gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/1/1)# speed 100gfull
```

3. Use the show interfaces status command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)# show interfaces status
Port      Name      Status      Vlan     Duplex    Speed     Type      Flags
Et1                 connected    2       full       1G     10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1             connected    1       full      100G    100GBASE-SR1
Et5/2/1             connected    1       full      100G    100GBASE-SR1
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

11.1.6.9.2 Configuring a CFP2 Module as Ten 10GbE Interfaces

To configure the port as four 10GbE interfaces, use the speed command (**speed 10000full**) on the ports ***/1*** lane (the primary lane).

This configuration is available only for CFP2-100G-XSR10 optics.

Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280CR3, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the CFP2 Ethernet interface.

```
switch(config)# interface ethernet 5/1/1
```

2. Enter the **speed 10000full** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/1/1)# speed 10000full
```

3. Use the show interfaces status command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)# show interfaces status
```

```
Port     Name    Status     Vlan Duplex   Speed   Type       Flags
Et1             connected    2    full      1G   10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1         connected    1    full      10G  100GBASE-SR1
Et5/1/2         connected    1    full      10G  100GBASE-SR1
Et5/1/3         connected    1    full      10G  100GBASE-SR1
Et5/1/4         connected    1    full      10G  100GBASE-SR1
Et5/1/5         connected    1    full      10G  100GBASE-SR1
Et5/1/6         connected    1    full      10G  100GBASE-SR1
Et5/1/7         connected    1    full      10G  100GBASE-SR1
Et5/1/8         connected    1    full      10G  100GBASE-SR1
Et5/1/9         connected    1    full      10G  100GBASE-SR1
Et5/1/10        connected    1    full      10G  100GBASE-SR1
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

11.1.6.10 Default QSFP Mode Support

QSFP+ transceiver supports 40G and 4x10G. This feature provides support for changing the default QSFP mode between 40G and 4x10G on all ports with QSFP+ transceivers.

11.1.6.10.1 Configuration

On all front panel ports which support this feature, the following global configuration command changes their default QSFP mode from 40G to 4x10G,

`transceiver qsfp default-mode 4x10G`

The **`no`** or **`default`** version of the command reverts the default QSFP mode back to 40G.

Note: This configuration command is not honored on ports that do not support this feature (such as QSFP100 ports with external PHY always choose 40G as the default QSFP mode).

11.1.6.10.2 Show Commands

There is no explicit show command to display the default QSFP mode. Use the **`show running-config`** command to determine whether the default QSFP mode is 4x10G or 40G. Use the **`show interfaces hardware`** command to check the default QSFP mode on the given ports.

- When default QSFP mode is configured as 4x10G, the output of **`show running-config`** contains **`transceiver qsfp default-mode 4x10G`**. In the output of the **`show interfaces hardware`** command, 10G is shown as the default speed.

```
switch(config)# transceiver qsfp default-mode 4x10G
switch(config)# show running-config | grep 4x10G
transceiver qsfp default-mode 4x10G
switch(config)# show interfaces ethernet 35/1 hardware
*  = Requires speed group setting change
Ethernet35/1
  Model:          DCS-7280CR3-32P4
  Type:           40GBASE-CR4
  Speed/Duplex:   10G/full(default),40G/full,auto
  Flowcontrol:    rx-(off,on,desired),tx-(off)
```

- When default QSFP mode configuration is reverted the default, the output of **`show running-config`** command does not contain **`transceiver qsfp default-mode 4x10G`**. In the output of **`show interfaces hardware`** command, 40G is shown as the default speed.

```
switch(config)# no transceiver qsfp default-mode
switch(config)# show running-config | grep 4x10G
switch(config)# show interfaces ethernet 35/1 hardware
*  = Requires speed group setting change
Ethernet35/1
```

```
  Model:           DCS-7280CR3-32P4
  Type:            40GBASE-CR4
  Speed/Duplex:    10G/full,40G/full(default),auto
  Flowcontrol:     rx-(off,on,desired),tx-(off)
```

11.1.6.10.3 Limitations

There is support for the **`no transceiver qsfp default-mode 4x10G`** command on the DCS-7300X Series, but the default QSFP mode still remains as 4x10G.

11.1.6.11 MXP Ethernet Port Configuration

Each MXP module contains twelve data lanes which can be used individually or combined to form one or more higher-speed interfaces. This allows an MXP Ethernet port to be configured as a single 100GbE interface, up to twelve 10GbE interfaces, or a mixture of 40GbE and 10GbE ports.

MXP ports do not use pluggable optics: instead, an MTP-24 ribbon is inserted directly into the port. The remote end of the MTP 24 ribbon must then be broken out using a splitter cable or cartridge based on the operational mode and speed of the MXP port.

When four lanes of an MXP interface are combined to form a 40GbE port, CLI commands show the primary lane of that group as **connected or not connected** and the other three lanes as **errdisabled**.

The following sections describe the configuration of MXP interfaces.

11.1.6.11.1 Configuring an MXP Module as a Single 100GbE Interface

To configure the port as a single 100GbE interface (the default configuration), enter the speed command (**speed 100gfull**) on the ports ***/1*** lane (the primary lane). This combines lanes ***1-10*** and disables lanes ***11*** and ***12***.

Under this configuration, CLI display commands will show lane ***/1*** as **connected** or **not connected**, and show lanes ***/2-/12*** as **errdisabled**.



Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280CR3, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the MXP Ethernet interface.

```
switch(config)# interface ethernet 5/49/1
```

2. Enter the **speed 100gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/49/1)# speed 100gfull
```

3. Use the show interfaces status command to confirm the change in configuration.

```
switch(config-if-Et5/49/1)# show interfaces status
Port          Name        Status        Vlan        Duplex     Speed      Type          Flags
Et1                       connected      2           full        1G      10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/49/1                  connected      1           full       100G     100GBASE-SR1
Et5/49/2                  errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/3                  errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/4                  errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/5                  errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/6                  errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/7                  errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/8                  errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/9                  errdisabled    1           unconf     unconf   100GBASE-SR1
```

```
Et5/49/10                 errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/11                 errdisabled    1           unconf     unconf   100GBASE-SR1
Et5/49/12                 errdisabled    1           unconf     unconf   100GBASE-SR1
```

11.1.6.11.2 Configuring an MXP Module With 40GbE Interfaces

Each set of four lanes on the MXP module is independently configurable as a single 40GbE interface or four 10GbE interfaces. To configure four lanes as a single 40GbE interface, enter the speed command (**speed forced 40gfull**) on the groups primary lane ***(/1***, ***/5***, or ***/9***). To revert a group of four lanes to functioning as four independent 10GbE interfaces, enter the **speed 10000full** command on the primary lane of the group.

When four lanes of an MXP interface are combined to form a 40GbE port, CLI commands will show the primary lane of that group as **connected** or **not connected** and the other three lanes as **errdisabled**. In groups of four lanes which are configured as four independent 10GbE interfaces, each lane will be displayed in the CLI as **connected** or **not connected**.

Note that a **speed forced 100gfull** command entered on the /1 lane takes precedence over **speed 40gfull** commands on the ***/5*** and ***/9*** lanes.



Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280CR3, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

The example below shows the steps for configuring an MXP module as three 40GbE interfaces.

1. Enter interface Ethernet configuration mode for lane ***/1*** of the MXP Ethernet interface.

```
switch(config)# interface ethernet 5/49/1
```

2. Enter the **speed 40gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/49/1)# speed 40gfull
```

3. Repeat the above steps for ***lanes /5*** and ***/9***.

```
switch(config-if-Et5/49/1)# interface ethernet 5/49/5
switch(config-if-Et5/49/5)# speed 40gfull
switch(config-if-Et5/49/5)# interface ethernet 5/49/9
switch(config-if-Et5/49/9)# speed 40gfull
```

4. Use the `show interfaces status` command to confirm the change in configuration.

```
switch(config-if-Et5/49/9)# show interfaces status
Port Name Status Vlan Duplex Speed Type Flags
Et1          connected       2     full      1G      10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/49/1     connected       1     full     40G      100GBASE-SR1
Et5/49/2     errdisabled     1     unconf   unconf   100GBASE-SR1
Et5/49/3     errdisabled     1     unconf   unconf   100GBASE-SR1
Et5/49/4     errdisabled     1     unconf   unconf   100GBASE-SR1
Et5/49/5     connected       1     full     40G      100GBASE-SR1
Et5/49/6     errdisabled     1     unconf   unconf   100GBASE-SR1
Et5/49/7     errdisabled     1     unconf   unconf   100GBASE-SR1
Et5/49/8     errdisabled     1     unconf   unconf   100GBASE-SR1
Et5/49/9     connected       1     full     40G      100GBASE-SR1
Et5/49/10    errdisabled     1     unconf   unconf   100GBASE-SR1
Et5/49/11    errdisabled     1     unconf   unconf   100GBASE-SR1
Et5/49/12    errdisabled     1     unconf   unconf   100GBASE-SR1
```

11.1.6.11.3 Configuring an MXP Module as Twelve 10GbE Interfaces

Each lane of an MXP port functions as a 10GbE interface when it is not included in a higher-speed interface configuration (either actively or as an **errdisabled** port).

To explicitly configure the port as twelve 10GbE interfaces, use the `speed` command (**speed 10000full**) on all twelve lanes of the port.

When each lane is configured as an independent 10GbE interface, CLI display commands show each lane as **connected** or **not connected**.



Note: Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280CR3, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption. On 7160 series platforms, use of the **speed** command is hitless, but if the command changes the number of port lanes, packets may be dropped on unrelated ports.

1. Enter interface Ethernet configuration mode for all twelve lanes of the MXP Ethernet interface.

```
switch(config)# interface ethernet 5/49/1-12
```

2. Enter the **speed 10000full** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for ***60*** seconds or more.

```
switch(config-if-Et5/49/1-12)# speed 10000full
```

3. Use the `show interfaces status` command to confirm the change in configuration.

```
switch(config-if-Et5/49/1-12)# show interfaces status
Port       Name      Status       Vlan      Duplex       Speed      Type       Flags
Et1                  connected      2         full          1G      10GBASE-T
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1              connected      1         full          10G     100GBASE-SR1
Et5/1/2              connected      1         full          10G     100GBASE-SR1
Et5/1/3              connected      1         full          10G     100GBASE-SR1
Et5/1/4              connected      1         full          10G     100GBASE-SR1
Et5/1/5              connected      1         full          10G     100GBASE-SR1
Et5/1/6              connected      1         full          10G     100GBASE-SR1
Et5/1/7              connected      1         full          10G     100GBASE-SR1
Et5/1/8              connected      1         full          10G     100GBASE-SR1
Et5/1/9              connected      1         full          10G     100GBASE-SR1
Et5/1/10             connected      1         full          10G     100GBASE-SR1
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

11.1.6.12 Port Speed Capabilities

The supported speeds supported on each Arista platform per interface type are described in Supported Speeds (GbE).

Table 59: Supported Speeds (GbE)

Platform	SFP+	SFP28	QSFP+	QSFP100	MXP	CFP2
7050	100M, 1, 10	N/A	1, 10, 40	N/A	N/A	N/A
7050X	100M, 1, 10	N/A	1, 10, 40	N/A	10, 40	N/A
7050X2	100M, 1, 10	N/A	1, 10, 40	N/A	N/A	N/A
7050X3	100M, 1, 10	1, 10, 25	N/A	10, 25, 40, 50, 100	N/A	N/A
7250X	N/A	N/A	1, 10, 40	N/A	N/A	N/A
7060X	100M, 1, 10	N/A	N/A	10, 25, 40, 50, 100	N/A	N/A
7060X2	100M, 1, 10	1, 10, 25	N/A	10, 25, 40, 50, 100	N/A	N/A
7260X3	100M, 1, 10	N/A	N/A	10, 25, 40, 50, 100	N/A	N/A
7300X	100M, 1, 10	N/A	1, 10, 40	N/A	N/A	N/A
7300X3	N/A	1, 10, 25	N/A	10, 25, 40, 50, 100	N/A	N/A
7320X	N/A	N/A	N/A	10, 25, 40, 50, 100	N/A	N/A
7150S	1, 10	N/A	1, 10, 40	N/A	N/A	N/A
7048T	1, 10	N/A	N/A	N/A	N/A	N/A
7500	1, 10	N/A	1, 10, 40	N/A	N/A	N/A
7500E	1, 10	N/A	1, 10, 40	10, 40, 100	10, 40, 100	100
7500R	1, 10	1, 10, 25	1, 10, 40	10, 25, 40, 50, 100	N/A	N/A
7280SE	1, 10	N/A	1, 10, 40	10, 40, 100	10, 40, 100	N/A
7280QR	N/A	N/A	1, 10, 40	10, 25, 40, 50, 100	N/A	N/A
7280SR (R2)	1, 10	1, 10, 25	N/A	10, 25, 40, 50, 100	N/A	100, 200

7280CR	N/A	N/A	N/A	10, 25, 40, 50, 100	N/A	N/A
7010T	100M, 1, 10	N/A	N/A	N/A	N/A	N/A

11.1.6.13 Agile Ports

An agile port is an interface that can function as a 10G port or can subsume a predefined set of 10G interfaces to form an interface with higher speed capabilities.

The set of interfaces that can be combined to form a higher speed port is restricted by the hardware configuration. Only interfaces that pass through a common PHY component can be combined. One interface within a combinable set is designated as the primary port.

- To view the set of available agile ports and the subsumable interfaces that comprise them, enter `show platform fm6000 agileport map`.
- To configure the primary port as a higher speed port, enter `speed` **40gfull** or `speed` **auto 40gfull**.
- To revert the primary port and its subsumed ports to 10G interfaces, enter **no** `speed`.

11.1.6.14 Subinterface Configuration

For a subinterface to be operational on an Ethernet or port channel interface, the parent interface must be configured as a routed port and be administratively up, and a VLAN must be configured on the subinterface. If the parent interface goes down, all subinterfaces automatically go down as well, but will come back up with the same configuration once the parent interface is up.

Note that a port channel should not contain Ethernet interfaces with subinterfaces configured on them, and that subinterfaces cannot be members of a port channel.

Subinterfaces are named by adding a period followed by a unique subinterface number to the name of the parent interface. Note that the subinterface number has no relation to the ID of the VLAN corresponding to the subinterface.

Subinterfaces are available on the following platforms:

- DCS-7050X
- DCS-7060X
- DCS-7250X
- DCS-7260X
- DCS-7280E
- DCS-7300X
- DCS-7320X
- DCS-7500E

11.1.6.14.1 Creating a Subinterface

To create a subinterface on an Ethernet or port channel interface:

1. Bring up the parent interface and ensure that it is configured as a routed port.

```
switch(config)# interface Ethernet1/1
switch(config-if-Et1/1)# no switchport
switch(config-if-Et1/1)# no shutdown
```

2. Configure a VLAN on the subinterface. The `encapsulation dot1q vlan` command is also used for VLAN translation, but in this context it associates a VLAN with the subinterface.

```
switch(config-if-Et1/1)# interface Ethernet1/1.1
switch(config-if-Et1/1.1)# encapsulation dot1q vlan 100
```

3. Configure an IP address on the subinterface (optional) and ensure that it is up.

```
switch(config-if-Et1/1)# ip address 10.0.0.1/24
switch(config-if-Et1/1)# no shutdown
switch(config-if-Et1/1)#
```

11.1.6.14.2 Creating a Range of Subinterfaces

A range of subinterfaces can also be configured simultaneously. The following example configures subinterfaces ***1*** to ***100*** on ***Ethernet interface 1/1***, and assigns ***VLANs 501*** through ***600*** to them. Note that the range of interfaces must be the same size as the range of VLAN IDs.

Example

```
switch(config)# interface eth1/1.1-100
switch(config-if-Et1/1.1-100)# no shutdown
switch(config-if-Et1/1.1-100)# encapsulation dot1q vlan {501,600}
switch(config-if-Et1/1.1-100)# exit
switch(config)#
```

11.1.6.14.3 Parent Interface Configuration

For subinterfaces to function, the parent interface must be administratively up and configured as a routed port.

Some settings are inherited by subinterfaces from the parent interface. These include QoS (trust mode and default DSCP) and MTU.

Additionally, on the DCS-7050X, DCS-7250X, and DCS-7300X platforms, the parent interface may be configured with an IP address. In this case, untagged packets are treated as incoming traffic on the parent interface

11.1.6.14.4 Configuring Routing Features on a Subinterface

Once a subinterface is created, the following features can be configured on it:

- Unicast and multicast routing
- BGP, OSPF, ISIS, PIM
- VRF
- VRRP
- SNMP
- Inheritance of QoS (trust mode and default DSCP) and MTU settings from the parent interface

Additionally, these features can be configured on subinterfaces on Arad (DCS-7500E and DCS-7280E) platforms:

- Subinterface counters on ingress
- VXLAN
- MPLS
- GRE
- PBR

- QoS

11.1.6.14.5 Displaying Subinterface Information

Subinterface information is displayed using the same show commands as for other interfaces.

Examples

- This command displays summary information for all IP interfaces on the switch, including subinterfaces.

```
switch# show ip interfaces brief
 Interface            IP Address     Status    Protocol     MTU
 Ethernet1/1          10.1.1.1/24up      up        1500
 Ethernet1/1.1        10.0.0.1/24up      up        1500
 Ethernet1/2          unassigned         up          up     1500
```

- This command displays information for subinterface Ethernet ***1/1.1***.

```
switch# show interface ethernet 1/1.1
Ethernet1/1.1 is down, line protocol is lowerlayerdown (notconnect)
  Hardware is Subinterface, address is 001c.735d.65dc
  Internet address is 10.0.0.1/24
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  IP MTU 1500 bytes , BW 10000000 kbit
  Down 59 seconds switch>
```

- This command displays status information for all subinterfaces configured on the switch.

```
switch# show interfaces status sub-interfaces
Port       Name    Status       Vlan     Duplex       Speed       Type                 Flags
Et1.1              connect      101        full       10G         dot1q-encapsulation
Et1.2              connect      102        full       10G         dot1q-encapsulation
Et1.3              connect      103        full       10G         dot1q-encapsulation
Et1.4              connect      103        full       10G         dot1q-encapsulation
```

11.1.6.15 Maximum Latency Tail-drop Thresholds

The maximum latency for Tail-drop thresholds can be configured in interface configuration mode or in the QoS profile. **QoS profile** configuration mode is a group change mode.

Example

These commands configure the maximum latency value for VOQ tail-drop threshold on ***transmit queue 3*** on ***interface Ethernet1***. The latency value can be specified to a maximum of ***50*** ms. Both milliseconds and microseconds may be used.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# tx-queue 3
switch(config-if-Et1-txq-3)# latency maximum <1-50000> microseconds
switch(config-if-Et1-txq-3)# latency maximum <1-50> milliseconds
switch(config)#
```

11.1.6.16 Autonegotiated Settings

In autonegotiation, the transmission speed, duplex setting, and flow control parameters used for Ethernet-based communication can be automatically negotiated between connected devices to establish optimized common settings.

11.1.6.16.1 Speed and Duplex

The `speed` command affects the transmission speed and duplex setting for the configuration mode interface. When a **`speed`** command is in effect on an interface, autonegotiation of speed and duplex settings is disabled for the interface; to enable autonegotiation, use the **`speed auto`** command.

The scope and effect of the **`speed`** command depends on the interface type; see Ethernet Interfaces and Ethernet Configuration Procedures for detailed information on the speed settings for different interfaces.

11.1.6.16.2 Flow Control

Flow control is a data transmission option that temporarily stops a device from sending data because of a peer data overflow condition. If a device sends data faster than the receiver can accept it, the receiver's buffer can overflow. The receiving device then sends a PAUSE frame, instructing the sending device to halt transmission for a specified period.

Flow control commands configure administrative settings for flow control packets.

- The flowcontrol receive command configures the port's ability to receive flow control pause frames.
  - **off**: port does not process pause frames that it receives.
  - **on**: port processes pause frames that it receives.
  - **desired**: port autonegotiates; processes pause frames if peer is set to **send** or desired.
- The flowcontrol send command configures the port's ability to transmit flow control pause frames.
  - **off**: port does not send pause frames.
  - **on**: port sends pause frames.
  - **desired**: port autonegotiates; sends pause frames if peer is set to **receive** or **desired**.

Desired is not an available parameter option. Ethernet data ports cannot be set to **desired**. Management ports are set to **desired** by default and with the **`no flowcontrol receive`** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. Compatible Settings for Flow Control Negotiation lists the compatible flow control settings.

Table 60: Compatible Settings for Flow Control Negotiation

local port	peer port
receive on	send on or send desired
receive off	send off or send desired
receive desired	send on , send off, or send desired
send on	receive on or receive desired
send off	receive off or receive desired
send desired	receive on , receive off, or receive desired

Example

These commands set the flow control receive and send to **on** on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# flowcontrol receive on
switch(config-if-Et5)# flowcontrol send on
switch(config-if-Et5)#
```

11.1.6.17 Displaying Ethernet Port Properties

Show commands are available to display various Ethernet configuration and operational status on each interface. Ethernet settings that are viewable include:

- Port Type
- PHY Status
- Negotiated Settings
- Flow Control
- Capabilities

Port Type

The port type is viewable from the output of **show interfaces status**, **show interfaces hardware**, and **show interfaces transceiver properties** commands.

Examples

- This **`show interfaces status`** command displays the status of Ethernet interfaces ***1-5***.

```
switch# show interfaces status
Port      Name              Status       Vlan        Duplex  Speed Type
Et1                         connected    1             full    10G 10GBASE-SRL
Et2                         connected    1             full    10G 10GBASE-SRL
Et3                         connected    1             full    10G 10GBASE-SRL
Et4                         connected    1             full    10G 10GBASE-SRL
Et5                         notconnect   1             full    10G Not Present switch>
```

- This **`show interfaces hardware`** command displays the speed, duplex, and flow control capabilities of Ethernet interfaces ***2*** and ***18***.

```
switch# show interfaces ethernet 2,18 hardware
Ethernet2
  Model:         DCS-7150S-64-CL
  Type:          10GBASE-CR
  Speed/Duplex: 10G/full,40G/full,auto
  Flowcontrol:  rx-(off,on,desired),tx-(off,on,desired)
Ethernet18
  Model:         DCS-7150S-64-CL
  Type:          10GBASE-SR
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch>
```

- This command displays the media type, speed, and duplex properties for Ethernet interfaces ***1***.

```
switch# show interfaces ethernet 1 transceiver properties
Name : Et1
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL
```

PHY

PHY information for each Ethernet interface is viewed by entering the **show interfaces phy** command.

Example

This command summarizes PHY information for Ethernet interfaces ***1-3***.

```
switch#show interfaces ethernet 1-3 phy
Key:
```

```
   U    = Link up
   D    = Link down
   R    = RX Fault
   T    = TX Fault
   B    = High BER
   L    = No Block Lock
   A    = No XAUI Lane Alignment
   0123 = No XAUI lane sync in lane N

                                 State    Reset
Port           PHY state        Changes    Count PMA/PMD PCS   XAUI
-------------- ---------------- -------- -------- ------- ----- --------
Ethernet1      linkUp             14518     1750 U..     U.... U.......
Ethernet2      linkUp             13944     1704 U..     U.... U.......
Ethernet3detectingXcvr          3        1               D..A0123 switch>
```

Negotiated Settings

Speed, duplex, and flow control settings are displayed through the show interfaces hardware, PHY information for each Ethernet interface is viewed by entering the **show interfaces hardware**, **show interfaces flow-control**, and **show interfaces status** commands.

Examples

- This command displays speed/duplex and flow control settings for Ethernet interface ***1***.

```
switch#show interfaces ethernet 1 hardware
Ethernet1
  Model:        DCS-7150S-64-CL
  Type:         10GBASE-SR
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch>
```

- This command shows the flow control settings for Ethernet interfaces ***1-2***.

```
switch#show flow-control interface ethernet 1-2
Port       Send FlowControl  Receive FlowControl  RxPause       TxPause
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------------- -------------
Et1        off      off      off      off         0             0
Et2        off      off      off      off         0             0 switch>
```

- This command displays the speed type and duplex settings for management interfaces ***1-2***.

```
switch#show interfaces management 1-2 status
Port      Name               Status       Vlan        Duplex  Speed Type
Ma1                          connected    routed      a-full a-100M 10/100/1000
Ma2                          connected    routed      a-full   a-1G 10/100/1000 switch>
```

11.1.6.18 Ingress Counters

The Ingress counters enables the switch to count the ingress traffic on the Layer 3 ports of the switch.

Any ingress traffic on Layer 3 sub-interfaces and VLAN interface with IPv4 and IPv6 addresses are accounted irrespective of the routing decision. The VLAN counters are supported on DCS- 7050x, DCS-7250x, and DCS-7300x series switches and not supported on any routed ports.

11.1.6.18.1 Configuring Ingress Counters

The hardware counter feature in command enables the switch to count the ingress traffic on the Layer 3 port of the switch. Any traffic on Layer 3 sub-interfaces and VLAN interface with IPv4 and IPv6 addresses are accounted irrespective of the routing decision.

Examples

- This command configures the ingress traffic count on the sub-interfaces. The **no** form of the command disables the counter configuration from the switch ports.

```
switch# hardware counter feature subinterface in
```

- This command configures the ingress traffic count on the VLAN interface. The **no** form of the command disables the counter configuration from the VLAN configured switch ports.

```
switch# hardware counter feature vlan-interface in
```

11.1.6.18.2 Displaying the Ingress Counter Information

The show interface counters command displays the Layer 3 ingress traffic count information. Run this command to view the traffic counts on a sub-interface or VLAN interface of the switch. The clear counters command resets the counters to ***0***.

Example

This command displays the ingress traffic count on a VLAN interface ***vl12***.

```
switch# show interface vl12 counters incoming
L3 Interface InOctets InUcastPkts InMcastPkts
Vl12           3136          47           2
```

11.1.6.19 Configuring Ingress Traffic-Class Counters

Ingress traffic class counter support is enabled in order to display per traffic-class counters on ingress interfaces, and supported on routed-ports and subinterfaces. Both packet and octet counts are displayed.

Examples

- This command enables traffic-class counter support.

```
switch(config)# hardware counter feature traffic-class in
```

- This command enables TCAM profile tc-counters if this profile is configured.

```
switch(config)# hardware tcam profile tc-counters
```

11.1.6.20 Hardware Counter Support

Hardware counter support allows enabling counters for features using programmable hardware counter resources.

Hardware counter support can be used to count the following feature specific counters:

- Ingress VLAN-interface counters count the packets/octets ingressing through a VLAN interface.
- Egress VLAN-interface counters count the packets/octets egressing a VLAN interface.
- Ingress subinterface counters count the packets/octets ingressing through a subinterface.
- Egress subinterface counters count the packets/octets egressing a subinterface.
- VXLAN VNI counters count the packets/octets encapsulated/decapsulated per VNI.
- VXLAN VTEP counters count the packets/octets encapsulated/decapsulated per VTEP.
- Route counter for IPv4/IPv6 routes count the packets/octets routed using the route entry.
- Ingress GRE Tunnel Interface counters count the packets/octets ingressing through the GRE Tunnel Interface.
- Egress GRE Tunnel Interface counters count the packets/octets egressing through the GRE Tunnel Interface.

11.1.6.20.1 Hardware Counter Support Configuration

You can enable the Hardware Counter feature on a per-feature basis using the **`hardware counter feature`** command. The **`no hardware counter feature`** disables the feature. Multiple ingress and egress hardware counter features can work concurrently. Review MRU Enforcement Limitations for details.

Note: Enabling the hardware counter features may impact the transit traffic.

11.1.6.20.1.1 Ingress VLAN Interface Counters

The **`hardware counter feature vlan-interface in`** command enables the counting of ingress VLAN interface counters. When configured, the switch counts the number of unicast packets, multicast packets, and total octets ingressing through the VLAN interface for every VLAN interface configured in the system.

Example

```
switch# [no]hardware counter feature vlan-interface in
```

11.1.6.20.1.2 Egress VLAN Interface Counters

The **`hardware counter feature vlan-interface out`** command enables the counting of egress VLAN interface counters. When configured, the switch counts the number of unicast packets, multicast packets, and total octets egressing the VLAN interface for every VLAN interface configured in the system.

Example

```
switch# [no]hardware counter feature vlan-interface out
```

11.1.6.20.1.3 Ingress SubInterface Counters

The **`hardware counter feature subinterface in`** command enables the counting of ingress subinterface counters. When configured, the switch counts the number of unicast packets, multicast packets, and total octets ingressing through the subinterface for every L3 subinterface configured in the system.

Example

```
switch# [no]hardware counter feature subinterface in
```

11.1.6.20.1.4 Egress SubInterface Counters

The **`hardware counter feature subinterface out`** command enables the counting of egress subinterface counters. When configured, the switch counts the number of unicast packets, multicast packets, and total octets egressing the subinterface for every L3 subinterface configured in the system.

Example

```
switch# [no]hardware counter feature subinterface out
```

11.1.6.20.1.5 VXLAN VNI Encapsulation Counters

The **hardware counter feature vni encap** command enables the counting of per VNI encap counters. When enabled, the switch counts the number of packets and bytes egressing a VNI through VTI, encap BUM packets, and the number of encap packets dropped due to any reason.

Example

```
switch(config)# [no]hardware counter feature vni encap
```

11.1.6.20.1.6 VXLAN VNI Decapsulation Counters

The **hardware counter feature vni decap** command enables the per VNI decap counters. When enabled, the switch counts the number of packets/octets received from VTEP and decapsulated for each VNI.

Example

```
switch(config)# [no]hardware counter feature vni decap
```

11.1.6.20.1.7 Route Counters

The **hardware counter feature route** command configures the route counters for the specified IP version and ip-address/ip-prefix. When enabled, the switch counts the number of packets/bytes hitting a route. The VRF name is optional and default VRF is assumed if no VRF is specified.

Example

```
switch(config)# [no]hardware counter feature route[ipv4|ipv6]vrf
 <name>][ip-address|ip-address-prefix]
```

11.1.6.20.1.8 GRE Tunnel Interface Encapsulation Counters

The **hardware counter feature gre tunnel interface out** command enables GRE Tunnel Interface encap counters. When enabled, the switch counts the number of unicast GRE packets and total octets getting encapsulated on the Tunnel Interface.

Example

```
switch(config)# [no]hardware counter feature gre tunnel interface out
```

11.1.6.20.1.6 GRE Tunnel Interface Decapsulation Counters

The **hardware counter feature gre tunnel interface in** command enables GRE Tunnel Interface decap counters. When enabled, the switch counts the number of unicast GRE packets and total octets getting decapsulated on the Tunnel Interface.

Example

```
switch(config)# [no]hardware counter feature gre tunnel interface in
```

11.1.6.20.2 MRU Enforcement Show commands

The MRU on an interface is found in the show interface output.

```
switch(config)# show interface ethernet 1
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 444c.a8b7.1ed8 (bia 444c.a8b7.1ed8)
  Member of Port-Channel10
  Ethernet MTU 10178 bytes, Ethernet MRU 1500 bytes, BW 10000000 kbit
  Full-duplex, 10Gb/s, auto negotiation: off, uni-link: disabled.
```

Counter

MRU dropped packets are counted per-chip.

The Ethernet interfaces corresponding chip are found in the **show platform fap mapping** output.

```
switch(config)# show platform fap mapping interface Ethernet 1
Jericho0 (FapId: 0  BaseSystemCoreId: 0)
Port             SysPhyPort   Voq   Core  FapPort  OtmPort BaseQPair QPairs Xlge NifPort
-------------------------------------------------------------------------------------------
Ethernet1        100          2608  0      2       0        0         8     8     33
```

Reassembly Errors use per-chip counters from the show hardware counter drop output.

```
switch(config)# show hardware counter drop
Type  Chip       CounterName       : Count    : First Occurrence    : Last Occurrence
-------------------------------------------------------------------------------------
A     Jericho0   ReassemblyErrors  : 12132989 : 2020-09-22 17:05:45 : 2020-09-22 17:22:40
```

11.1.6.20.3 MRU Enforcement Limitations

- The number of features for which the hardware counters can be enabled simultaneously is limited by the availability of counter hardware resources in the system. When the configured hardware features exceed the available counter resources, not all counters for all features are available.
  - At lower scales of the feature key (for example, the number of interfaces in the case of subinterface/ SVI counters), you can typically concurrently support up to six (6) ingress features and three (3) egress features. As scales grow beyond table sizes, single features use multiple tables.
- Route counters do not support the counting of host routes or ALPM routes.
- Packets sent from the CPU are not included in egress counters.

11.1.6.21 Configuring Power over Ethernet (PoE)

Power over Ethernet (PoE) is enabled by default on all Ethernet ports of PoE-capable switches, and the switch will detect IEEE-compliant Powered Devices (PDs) when they are plugged into a port and supply power appropriately.

Limitations

- Ethernet ports will not detect non IEEE-compliant devices by default, and may not be able to detect or power them even if configured to do so.
- If attached PDs overload the switch, it will power off. This can occur when an attached PD increases its power demand via LLDP, when too many PDs are connected to the switch, or when a power supply fails on a heavily loaded dual-supply switch.
- Power-cycling the switch will cause temporary loss of power to attached PDs.
- PoE is not available on management interfaces.

Disabling PoE on an Interface

On switches which support PoE, it is enabled by default on all Ethernet ports but can be disabled per-port with the poe disabled command.

Example

These commands disable PoE on Ethernet interface 5.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# poe disabled
switch(config-if-Et5)#
```

PoE Power Settings

When an IEEE-compliant powered device (PD) is connected to a PoE-enabled Ethernet port, it is recognized by a specific resistor signature, and its initial power needs are determined by hardware negotiation, after which further negotiation is managed through the Link Layer Discovery Protocol (LLDP). For details, see Configuring LLDP for Power over Ethernet.

PoE power output can be limited on a port using the poe limit command. The power limit represents the power output at the Ethernet port; actual power delivered to the PD will be lower due to power loss along the Ethernet cable.

Note: LLDP uses Power Via MDI type-length-value elements (TLVs) to allow the switch to dynamically negotiate power needs with PDs. LLDP will not include Power Via MDI TLVs for the interface if a power limit has been configured on it.

Examples

- These commands limit nominal PoE power output on Ethernet interface ***5*** to ***10*** W.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# poe limit 10 watts
switch(config-if-Et5)#
```

- These commands limit nominal PoE power output on Ethernet interface ***7*** to ***4*** W.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# poe limit class 1
switch(config-if-Et7)#
```

Detecting Legacy PDs

IEEE-compliant Powered Devices (PDs) are recognized by a specific resistance signature to a test signal sent by the switch, but non-compliant (legacy or proprietary) PDs may use a capacitive signature instead. By default, legacy PD detection is disabled, and legacy devices are not powered.

To configure an interface to use hardware detection for these non-compliant PoE devices and attempt to power them, use the poe legacy detect command.

Note: Non IEEE-compliant PDs are not officially supported. Arista cannot guarantee compatibility with such devices, and they may not be detected even when legacy detection is enabled on the port they are connected to.

Example

These commands configure ***interface ethernet 5*** to attempt to detect and power non-compliant PDs.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# poe legacy detect
switch(config-if-Et5)#
```

Displaying PoE Information

To display PoE information for a specific interface range or for all Ethernet interfaces, use the show poe command.

Example

This command displays PoE information for ***Ethernet interface 46***.

```
switch(config)# show poe interface ethernet 46
show poe interface ethernet 46
PSELLDPPowerGrantedPort
Port Enabled Enabled Limit  Power   State   Class  Power Current Voltage Temperature
---- ------- ------- ------ ------- ------- ------ ----- ------- ------- -----------
46      True    True 15.40W 15.40W  powered class0  1.40W 27.00mA 55.04V  41.25C
switch(config-if-Et7)#
```

11.1.6.22 Configuring Link Fault Signaling

As part of the Link Fault Signaling (LFS) configuration, a new configuration mode called the Ethernet Operations Administration and Management (EOAM) mode is introduced. The EOAM profile has a link-error sub-mode wherein the threshold, action, and the period is configured for both FCS and Symbol errors. The period can be in seconds or in number of frames. The default values are threshold ***0***, action syslog, and period ***0*** seconds. If the errors exceed the threshold within the given period, the configured action is executed. The recovery time configures the recovery timeout value for link fault signaling. Only one EOAM profile is associated with a port.

The following steps enable configuring the LFS parameters:

1. Enable the EOAM mode.

```
switch(config)# monitor ethernet oam
```

2. Create an EOAM profile named as ***profile1***.

```
switch(config-eoam)# profile profile1
```

3. Enter the EOAM ***link-error sub-mode***.

```
switch(config-eoam-profile-profile1)# link-error
```

4. Enter the commands in the profile ***link-error submode*** to configure a specific LFS parameter.

```
switch(config-eoam-profile-profile1-link-error)# symbol action errdisable
switch(config-eoam-profile-profile1-link-error)# symbol period 300 frames
switch(config-eoam-profile-profile1-link-error)# symbol threshold 20
switch(config-eoam-profile-profile1-link-error)# recovery-time 40
```

5. Apply the EOAM profile ***profile1*** to the Ethernet interface ***1/1***.

```
switch(config)# interface ethernet 1/1
```

```
switch(config-if-Et1/1)# monitor ethernet oam profile profile1
```

11.1.6.23 Configuring Hardware TCAM

Ternary Content-Addressable Memory (TCAM) is a specialized type of high-speed memory that increase the speed of route look-up, packet classification, packet forwarding and access control list-based commands. The **hardware tcam** command is used to configure and place the switch in the TCAM mode. In this mode the user can configure few TCAM related commands such as feature, profile and system.

In the TCAM mode, use the feature command to configure the reservation of TCAM banks for the features like ACL, IPsec, flow-spec, l2-protocol, PBR, QoS, TCP-MSS-ceiling, traffic-policy. The profile command configures a new TCAM profile, or just copy the TCAM profile which is already created using the hardware tcam profile command such as default, mirroring-acl, pbr-match-nexthop-group, qos, tap-aggregation-default, tap-aggregation-extended, tc-counters, test, vxlan-routing. Similarly, the system command configures the system-wide TCAM profiles.

Example

This command places the switch in Hardware TCAM configuration mode.

```
switch(config)# hardware tcam
switch(config-hw-tcam)#
```

These are the commands allowed to configure in Hardware TCAM mode.

Examples

- This command allow the switch to configure the TCAM feature.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# feature
```

- This command allow the switch to configure TCAM profile.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile
```

- This command allow the switch to configure TCAM system profile.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# system
```

11.1.6.24 CPU Traffic Policy

Create a TCAM profile to enable actions such as permit, deny, and police in hardware before the traffic gets to the kernel for processing. The action is taken based on IP packet header information such as DSCP, L4 port values, fragmentation bits, etc.

TCAM Profile Example

The following file extract displays an example of a TCAM profile, **cpu-traffic-policy**, with features configured to enable CPU traffic policy.

```
hardware tcam
   profile cpu-traffic-policy
      feature acl port ip
         sequence 45
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops l4-src-
port src-ip tcp-control ttl
```

```
         action count drop
         packet ipv4 forwarding bridged
         packet ipv4 forwarding routed
         packet ipv4 forwarding routed multicast
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet ipv4 non-vxlan forwarding routed decap
         packet ipv4 vxlan eth ipv4 forwarding routed decap
         packet ipv4 vxlan forwarding bridged decap
      feature acl port ipv6
         sequence 25
         key field dst-ipv6 ipv6-next-header ipv6-traffic-class l4-dst-port l4-
ops-3b l4-src-port src-ipv6-high src-ipv6-low tcp-control
         action count drop mirror
         packet ipv6 forwarding bridged
         packet ipv6 forwarding routed
         packet ipv6 forwarding routed multicast
         packet ipv6 ipv6 forwarding routed decap
      feature acl subintf ip
         sequence 40
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops-18b l4-
src-port src-ip tcp-control ttl
         action count drop mirror
         packet ipv4 forwarding routed
      feature acl subintf ipv6
         sequence 15
         key field dst-ipv6 ipv6-next-header l4-dst-port l4-src-port src-ipv6-
high src-ipv6-low tcp-control
         action count drop mirror redirect
         packet ipv6 forwarding routed
      feature counter lfib
         sequence 85
      feature mpls
         sequence 5
         key size limit 160
         action drop redirect set-ecn
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding mpls
         packet mpls non-ip forwarding mpls
      feature mpls pop ingress
         sequence 90
      feature pbr mpls
         sequence 65
         key size limit 160
         key field mpls-inner-ip-tos
         action count drop redirect
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding mpls
         packet mpls non-ip forwarding mpls
      feature qos ip
         sequence 75
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops l4-src-
port src-ip
         action set-dscp set-policer set-tc
         packet ipv4 forwarding routed
         packet ipv4 forwarding routed multicast
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet ipv4 non-vxlan forwarding routed decap
      feature qos ipv6
```

```
         sequence 70
         key field dst-ipv6 ipv6-next-header ipv6-traffic-class l4-dst-port l4-
src-port src-ipv6-high src-ipv6-low
         action set-dscp set-policer set-tct
         packet ipv6 forwarding routed
      feature traffic-policy cpu ipv4
         sequence 1
         key size limit 160
         key field dst-ip ip-frag ip-protocol l4-dst-port l4-src-port src-ip
         action count set-drop-precedence set-policer
      feature traffic-policy cpu ipv6
         sequence 2
         key field dst-ipv6 ipv6-next-header l4-dst-port l4-src-port src-ipv6-
high src-ipv6-low
         action count set-drop-precedence set-policer
   system profile cpu-traffic-policy
```

Configuring the Policy

The following configures **cpu-traffic-policy** under the traffic-policies global configuration mode.

```
traffic-policies
  cpu traffic-policy <name> vrf <vrf-list>
  traffic-policy <name>
    match <rule-name> <ipv4 | ipv6> [ <after | before> <rule-name> ]
      source prefix       A.B.C.D/E [ A.B.C.D/E ]
      destination prefix  A.B.C.D/E [ A.B.C.D/E ]
      protocol <protocol-list> | {tcp,udp}
                          [ source port <port-list> |
                                        field-set <port-set-name> ] |
                          [ destination port <port-list> |
                                        field-set <port-set-name> ]
      actions
         drop
         police rate <rate> <unit>
         count
```

The following applies the policies to the VRFs. The policy is applied to all interfaces belonging to the VRF.

```
traffic-policies
  cpu traffic-policy <name> vrf all
```

Default Rules

The **ipv4-all-default** and **ipv6-all-default** are programmed by default and cannot be reordered, or removed. Only the associated action can be changed. The **permit** rule is never installed and is the default action for packet processing.

Examples

L3 Protocol Match Criteria (protocol neighbors bgp)

The following displays the configuration of a profile to match on a set of configured protocol peers which allows only TCP traffic from those BGP peers.

```
traffic-policies
  cpu traffic-policy <name> vrf <vrf-list>
  traffic-policy <name>
    match <rule-name> <ipv4 | ipv6> [ <after | before> <rule-name> ]
      source prefix       A.B.C.D/E [ A.B.C.D/E ]
      destination prefix  A.B.C.D/E [ A.B.C.D/E ]
      protocol <protocol-list> | {tcp,udp}
```

```
                           [ source port <port-list> |
                                        field-set <port-set-name> ] |
                           [ destination port <port-list> |
                                        field-set <port-set-name> ]
      actions
         drop
         police rate <rate> <unit>
         count

traffic-policies
  cpu traffic-policy <traffic-policy-name> vrf all
    match <rule-name> <ipv4 | ipv6> [ <after | before> <rule-name> ]
      protocol neighbors bgp
      actions
         drop

traffic-policies
   cpu traffic-policy foo vrf all
      traffic-policy foo
         match BGP ipv4
            protocol neighbors bgp
```

The following policy permits dynamic peers from the listen-range ***2.0.0.0/24*** and all IPv6 BGP LL peers.

```
traffic-policies
   traffic-policy cpuPolicy
      match BGP ipv4
         protocol neighbors bgp
      match BGP-DYN ipv4
         source prefix 2.0.0.0/24
      match BGP-LL ipv6
         source prefix fe80::/64
```

Note: Configuring any match criteria will result in a CLI error when combined with the **`protocol neighbor bgp`** match criteria as these conflict with the expanded BGP rules.

The following displays error messages with conflicting configurations.

```
traffic-policies
   traffic-policy cpuPolicy
      match BGP ipv4
         protocol neighbors bgp
         source prefix 1.0.0.1/32
! The 'source prefix' subcommand is not supported when the 'protocol neighbors'
 subcommand is configured.

traffic-policies
   traffic-policy cpuPolicy
      match BGP ipv4
         source prefix 1.0.0.1/32
         protocol neighbors bgp
! The 'protocol neighbors' subcommand is not supported when any other match
 subcommands are configured.
```

Deny other BGP Traffic

The following displays a policy to whitelist some BGP sessions and other management IPs or IP protocol. The policy is inadequate as attempts to create BGP sessions from ***1.0.0.0/24*** (for example) will pass through this policy unfiltered.

```
traffic-policies
   cpu traffic-policy CPU vrf all
   traffic-policy CPU
```

```
      match BGP ipv4
         protocol neighbors bgp
      match BFD ipv4
         source address 1.0.0.0/24 1.0.1.0/24 1.0.4.0/24
      match ipv4-all-default ipv4
         actions
            drop
      match ipv6-all-default ipv6
         actions
            drop
```

The following displays a policy to whitelist some BGP sessions and other management IPs or IP protocol. The policy is effective and easier to implement for administrators.

```
traffic-policies
   cpu traffic-policy CPU
   traffic-policy CPU
      match BGP ipv4
         protocol neighbors bgp
      match BGP-REST ipv4
         protocol tcp udp destination port 179
         actions
            drop
      match BFD ipv4
         source address 1.0.0.0/24 1.0.1.0/24 1.0.4.0/24
      match ipv4-all-default ipv4
         actions
            drop
      match ipv6-all-default ipv6
         actions
            drop
```

The following displays a policy to whitelist some BGP sessions and other management IPs or IP protocol. The policy is effective and easier to implement for administrators. The clause matches on any inbound traffic destined for the configured BGP port(s).

```
traffic-policies
   cpu traffic-policy CPU
   traffic-policy CPU
      match BGP ipv4
         protocol neighbors bgp
      match BGP-REST ipv4
         protocol bgp
         actions
            drop
      match BFD ipv4
         source address 1.0.0.0/24 1.0.1.0/24 1.0.4.0/24
      match ipv4-all-default ipv4
         actions
            drop
      match ipv6-all-default ipv6
         actions
            drop
```

Actions

Drop: packet is discarded on a match

Police: tracks the usage for each BGP neighbor, and holds it to a maximum rate.

The **police rate <rate-value> [rate-unit]** command sets the policing rate. The optional **rate-unit** are bps, kbps (default), mbps, and gbps. The following configures a policing rate of 15 kbps for each BGP neighbor.

```
traffic-policies
   cpu traffic-policy CPU vrf all
   traffic-policy CPU
      match BGP ipv4
         protocol neighbors bgp
         actions
            police rate 15 kbps
```

Count

The following enables counters for the CPU traffic policy to count all packets matching the rule in which the **count** action is configured.

```
no hardware counter feature acl out <address-family>

hardware counter feature traffic-policy cpu

traffic-policies
   cpu traffic-policy CPU vrf all
   traffic-policy CPU
      match BGP ipv4
         protocol neighbors bgp
         actions
            count
```

Use-case Scenarios
Securing BGP Neighbors

The following implements a policy to secure BGP neighbors.

```
traffic-policies
   cpu traffic-policy CPU-DEFAULT vrf all
   traffic-policy CPU-DEFAULT
      match ICMPV6 ipv6
         protocol icmpv6
      match BGP ipv4
         protocol neighbors bgp
      match BGP-OTHER ipv4
         protocol bgp
         actions
            drop
      match BGP6 ipv6
         protocol neighbors bgp
      match BGP-OTHER6 ipv6
         protocol bgp
         actions
            drop
      match OSPF ipv4
         protocol ospf
      match ipv4-all-default ipv4
         actions
            drop
      match ipv6-all-default ipv6
         actions
            drop
```

Whitelist BFD Neighbors (not neighbor-specific)

Add the following rule to Whitelist BFD Neighbors.

```
traffic-policies
   traffic-policy CPU-DEFAULT
      match BFD-DPORT ipv4
         protocol udp destination port 3784-3785
      match BFD-SPORT ipv4
         protocol udp source port 49152
      match BFD-DPORT6 ipv6
         protocol udp destination port 3784-3785
      match BFD-SPORT6 ipv6
         protocol udp source port 49152
      match ipv4-all-default ipv4
         actions
            drop
      match ipv6-all-default ipv6
         actions
            drop
```

Whitelist a few Common Control-plane Protocols and Deny others

Add the following rule to Whitelist protocols such as ICMPv6 while denying others.

```
traffic-policies
   cpu traffic-policy CPU-DEFAULT vrf all
   traffic-policy CPU-DEFAULT
      match ICMPV6 ipv6
         protocol icmpv6
      match BGP ipv4
         protocol neighbors bgp
      match BGP-OTHER ipv4
         protocol bgp
         actions
            drop
      match BGP6 ipv6
         protocol neighbors bgp
      match BGP-OTHER6 ipv6
         protocol bgp
         actions
            drop
      match OSPF ipv4
         protocol ospf
      match PIM4 ipv4
         protocol pim
      match PIM6 ipv6
         protocol pim
      match ipv4-all-default ipv4
         actions
            drop
      match ipv6-all-default ipv6
         actions
            drop
```

11.1.6.25 TCAM Profile for Configurable Port Qualifier Sizing

A TCAM profile can be created from scratch, or the feature can be added to a copy of the default TCAM profile. When creating a profile from scratch, care must be taken to ensure that all needed TCAM features are included in the profile.

Modifying a Default TCAM Profile to allow Dynamic Sizing for ACL Labels

The following commands create a copy of the default TCAM profile, name it **port-qualifier-size**, and configure it to support a 6-bit dynamic sizing for ACL labels for IPKGV. Once the profile has been created, it can be applied to the system.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile port-qualifier-size copy default
switch(config-hw-tcam-profile-port-qualifier-size)# feature port-qualifier-size
 ip copy
system-feature-source-profile
switch(config-hw-tcam-profile-port-qualifier-size-feature-port-qualifier-size
)# port-qualifier-size 6
switch(config-hw-tcam-profile-port-qualifier-size-feature-port-qualifier-size
)# exit
switch(config-hw-tcam-profile-port-qualifier-size)# feature acl port ip
switch(config-hw-tcam-profile-port-qualifier-size)# exit
switch(config-hw-tcam)# exit
switch(config)#
```

Verifying the Configuration of Dynamic Qualifier Size

The following command verifies the configuration.

```
switch# show running-config
hardware tcam
   profile port-qualifier-size
      feature acl port ip
         port qualifier size 6 bits
```

Verifying the Configuration of System Profile

The following command verifies the system profile was configured and applied successfully. The profile name appears in both the Configuration and Status columns.

```
switch# show harware tcam profile
              Configuration               Status
Linecard1     port-qualifier-size         port-qualifier-size
Linecard2     port-qualifier-size         port-qualifier-size
Linecard3     port-qualifier-size         port-qualifier-size
```

When the profile does not get applied correctly, the Status column shows Error.

```
switch#show harware tcam profile
                Configuration              Status
Linecard1       port-qualifier-size        ERROR
Linecard2       port-qualifier-size        ERROR
Linecard3       port-qualifier-size        ERROR
```



Note: The packet type, the key field must be set. The dynamic qualifier size can be set for ACL. Not all hardware platforms support this feature. Modular systems require all linecards to support this feature for the profile to be applicable.

11.1.7 Ethernet Configuration Commands

Global Configuration Commands

- clear counters
- clear vxlan counters

- hardware port-group
- hardware counter feature
- hardware counter feature in (DCS-7050x, 7350x, 7300x)
- hardware tcam
- interface ethernet
- interface ethernet create
- interface management
- monitor ethernet oam
- transceiver channel
- transceiver qsfp default-mode

Hardware TCAM Commands

- feature
- system

Interface Configuration Commands Ethernet and Management Interfaces

- flowcontrol receive
- flowcontrol send
- link-debounce
- mac-address
- poe disabled
- poe legacy detect
- poe limit
- power budget
- power budget exceed action (warning | hold-down)
- speed

EOAM Configuration Commands

- link-error
- monitor ethernet oam profile
- profile

Link-error Configuration Commands

- action
- period
- phy link detection aggressive
- recovery-time
- threshold

Interface Display Commands

- show hardware counter
- show hardware port-group
- show interfaces counters
- show interfaces counters bins
- show interfaces counters errors
- show interfaces counters queue
- show interfaces counters rates
- show interfaces flow-control

- show interfaces hardware
- show interfaces hardware default
- show interfaces interactions
- show interfaces negotiation
- show interfaces phy
- show interfaces status
- show interfaces status errdisabled
- show interfaces transceiver
- show interfaces transceiver channels
- show interfaces transceiver hardware
- show interfaces transceiver properties
- show monitor ethernet oam profile
- show platform fm6000 agileport map
- show platform trident flexcounters
- show platform trident flexcounters l3intf
- show poe
- show route counters
- show vxlan counters vni

400GBASE-ZR Transceivers Display Commands

- show interface transceiver dom
- show interface transceiver eeprom
- show transceiver status interface

Shared Support across Multiple Subinterfaces Commands

- qos scheduling
- show qos scheduling

11.1.7.1 action

The **action** command configures the link monitoring action that is specified for the link fault signaling event.

The **no action** command removes the action type specified for the chosen link fault signaling. The **default action** command configures the link monitoring action as system log type.

Command Mode

Link-error Configuration

Command Syntax

{fcs | symbol} **action** [linkfault | errdisable | log]

no {fcs | symbol} **action** [linkfault | errdisable | log]

default {fcs | symbol} **action** [linkfault | errdisable | log]

Parameters

- **fcs** Inbound packets with frame check sequence (FCS) error.
- **symbol** Inbound packets with symbol error.
- **linkfault** The link fault action type.
- **errdisable** The errdisable action type.
- **log** The system log action type.

Related Commands

- period
- threshold

Example

These commands set the errdisable action type for the profile ***profile1*** in the Link-error configuration mode for symbol error.

```
switch(config)# monitor ethernet oam
switch(config-eoam)# profile profile1
switch(config-eoam-profile-profile1)# link-error
switch(config-eoam-profile-profile1-link-error)# symbol action errdisable
```

11.1.7.2 clear counters

The **`clear counters`** command in ***privileged EXEC*** mode resets the counters to zero for the specified interfaces including VLAN interfaces, Sub-Interfaces, and GRE Tunnel Interfaces.

Command Mode

Privileged EXEC

Command Syntax

`clear counters` [vlan ***vlanId*** | Tunnel ***tunnelID***][[route][ipv4 | ipv6][vrf]]

Parameters

- **vlan *vlanId*** Clears the VLAN Interface and subinterface Ingress and Egress counters.
- **Tunnel *tunnelID*** Allows the command to be used for clearing counters only for the specified VLAN interface.
- **route** Clears the route counters.
  - **ipv4** Allows targeting the clearing of IPv4 route counters.
  - **ipv6** Allows targeting the clearing of IPv6 route counters.
- **vrf** Allows specifying the VRF to only clear the route counters for the VRF.

Examples

- Clears the VLAN Interface and subinterface ingress and egress counters.

```
switch# clear counters[vlan vlanId]
```

- Clears the GRE Tunnel Interface Ingress and Egress counters.

```
switch# clear counters[tunnel TunnelId]
```

11.1.7.3 clear vxlan counters

Use the **`clear vxlan counters`** command to clear the VXLAN encap and decap counters.

Command Mode

EXEC

Command Syntax

`clear vxlan counters` [[vni | [vtep [vtep-ip-address | unlearnt]]]

Parameters

- **vni** Clears the VXLAN counters.
- **vtep** Clears the VTEP counters.

- **vtep-ip-address** Clears the counters for the specified VTEP.
- **unlearnt** Clears the counters for unlearned VTEPs. Unlearned VTEPs track the counts of packets received by the switch before the VTEP is learned.

11.1.7.4 feature

The **feature** command allows the user to reserve the number of TCAM banks for the following features such as ACL, flow-spec, IPsec, l2-protocol, PBR, QoS, TCP-MSS-ceiling, traffic-policy.

The **exit** command returns the switch to global configuration mode.

Command Mode

Hardware TCAM

Command Syntax

feature

Example

This commands allows the switch to configure the TCAM flow-spec feature for IPv4 ports.

```
switch(config-hw-tcam)# feature flow-spec port ipv4 bank maximum count 12
```

11.1.7.5 flowcontrol receive

The **flowcontrol receive** command configures administrative settings for inbound flowcontrol packets. Ethernet ports use flow control to delay packet transmission when port buffers run out of space. Ports transmit a pause frame when their buffers are full, signaling their peer ports to delay sending packets for a specified period.

The flowcontrol receive command configures the configuration mode port's ability to receive flowcontrol pause frames.

- ***off***: port does not process pause frames that it receives.
- ***on***: port processes pause frames that it receives.
- ***desired***: port autonegotiates flow control; processes pause frames if the peer is set to send **desired**.

Desired is not an available parameter option. Ethernet data ports cannot be set to desired. Management ports are set to desired by default and with the **no flowcontrol receive** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. The table below lists the compatible flow control settings.

Table 61: Compatible Settings for Flow Control Negotiation – Local Port Receiving

local port	peer port
receive on	send on or send desired
receive off	send off or send desired
receive desired	send on , send off, or send desired

The **no flowcontrol receive** and **default flowcontrol receive** commands restore the default flowcontrol setting for the configuration mode interface by removing the corresponding **flowcontrol receive** command from **running-config**. The default setting is **off** for Ethernet data ports and desired for Management ports.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

`flowcontrol receive` STATE

`no flowcontrol receive`

`default flowcontrol receive`

Parameters

STATE flow control pause frame processing setting. Options include:

- **On**
- **Off**

Example

These commands set the flow control received on Ethernet interface ***5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# flowcontrol receive on
switch(config-if-Et5)#
```

11.1.7.6 flowcontrol send

The **`flowcontrol send`** command configures administrative settings for outbound flow control packets. Ethernet ports use flow control to delay packet transmission when port buffers run out of space. Ports transmit a pause frame when their buffers are full, signaling their peer ports to delay sending packets for a specified period.

The **`flowcontrol send`** command configures the configuration mode port's ability to transmit flow control pause frames.

- **off**: port does not send pause frames.
- **on**: port sends pause frames.
- **desired**: port autonegotiates flow control; sends pause frames if the peer is set to **receive desired**.

Desired is not an available parameter option. Ethernet data ports cannot be set to **desired**. Management ports are set to **desired** by default and with the **`no flowcontrol send`** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. Compatible Settings for Flow Control Negotiation Local Port Transmitting lists the compatible flow control settings.

Table 62: Compatible Settings for Flow Control Negotiation Local Port Transmitting

local port	peer port
send on	receive on or receive desired
send off	receive off or receive desired
send desired	receive on , receive off, or receive desired

The **`no flowcontrol send`** and **`default flowcontrol send`** commands restore the default flow control setting for the configuration mode interface by removing the corresponding **`flowcontrol`**

send command from ***running-config***. The default setting is **off** for Ethernet data ports and **desired** for Management ports.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

flowcontrol send **STATE**

no flowcontrol send

default flowcontrol send

Parameters

STATE Flow control send setting. Options include:

- **on**
- **off**

Example

These commands set the flow control sent on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# flowcontrol send on
switch(config-if-Et5)#
```

11.1.7.7 hardware counter feature

You can enable on a per feature basis the **hardware counter feature** command from the global configuration mode. The **no hardware counter feature** disables the feature. Multiple ingress and egress hardware counter features can work concurrently.

Command Mode

Global configuration mode

Command Syntax

hardware counter feature [[vlan-interface | subinterface] [in | out]][vni [encap | decap]][[route [ipv4 | ipv6][vrf ***name***][ip-address | ip-address-prefix]][[gre tunnel interface][in | out]]

no hardware counter feature [[vlan-interface | subinterface] [in | out]][vni [encap | decap]][[route [ipv4 | ipv6][vrf *name*][ip-address | ip-address-prefix]][[gre tunnel interface][in | out]]

Parameters

- **vlan-interface** Enables VLAN interface counters
  - **in** Enables the counting of ingress VLAN interface counters. When configured, the switch counts the number of unicast packets, multicast packets and total octets ingressing through the VLAN interface for every VLAN interface configured in the system.
  - **out** Enables the counting of egress VLAN interface counters. When configured, the switch counts the number of unicast packets, multicast packets and total octets egressing the vlan interface for every vlan interface configured in the system.
- **subinterface** Enables the counting subinterface counters.
  - **in** Enables the counting of ingress subinterface counters. When configured, the switch counts the number of unicast packets, multicast packets and total octets ingressing through the subinterface for every L3 subinterface configured in the system.

    - **out** Enables the counting of egress subinterface counters. When configured, the switch counts the number of unicast packets, multicast packets and total octets egressing the subinterface for every L3 subinterface configured in the system.
- **vni** Enables the counting of per VNI counters.
    - **encap** Enables the per encap counters. When enabled, the switch counts the number of packets/octets coming from the edge, encapsulated on the device and directed towards the core.
    - **decap** Enables the per VTEP decap counters. When enabled, the switch counts the number of packets/octets coming from the core, decapsulated on the device and heading towards the edge per each VTEP.
- **route** Configures the route counters for the specified ip version and ip-address/ip-prefix. When enabled, the switch counts the number of packets/bytes hitting a route.
    - **ipv4** Identifies the IPv4 route.
    - **ipv6** Identifies the IPv6 route.
- **vrf *name*** The VRF name is optional and default VRF is assumed if no VRF is specified.
- **ip-address** Specifies the target IP address.
- **ip-address-prefix** Specifies the target IP address prefix.
- **gre tunnel interface** Enables GRE tunnel interface counters.
    - **in** Enables GRE tunnel interface decap counters. When enabled, the switch counts the number of unicast GRE packets and total octets getting decapsulated on the tunnel interface.
    - **out** Enables GRE Tunnel Interface encap counters. When enabled, the switch counts the number of unicast GRE packets and total octets getting encapsulated on the Tunnel Interface.

Examples

```
switch# hardware counters feature vlan-interface in
```

```
switch# hardware counter feature vlan-interface out
```

```
switch# hardware counter feature subinterface in
```

```
switch# hardware counter feature subinterface out
```

```
switch(config)# hardware counter feature vni encap
```

```
switch(config)# hardware counter feature vni decap
```

```
switch(config)# hardware counter feature vtep encap
```

```
switch(config)# hardware counter feature vtep decap
```

```
switch(config)# hardware counter feature gre tunnel interface in
```

```
switch(config)# hardware counter feature gre tunnel interface out
```

11.1.7.8 hardware counter feature in (DCS-7050x, 7350x, 7300x)

The **hardware counter feature** command enables the switch to count the ingress traffic on the Layer 3 port of the switch. Any traffic on Layer 3 sub-interfaces and VLAN interface with IPv4 and IPv6 addresses are accounted irrespective of the routing decision.

The **no hardware counter feature in** command disable the counter configuration from the switch ports. By default the ingress counter is disabled on the switch.

Command Mode

Global Configuration

Command Syntax

hardware counter feature [**INTERFACE**]in

no hardware counter feature [**INTERFACE**]in

Parameters

INTERFACE Layer 3 interface on the switch.

- **subinterface** Displays the subinterface traffic count.
- **vlan-interface** Displays the VLAN-interface traffic count.

Examples

- This command configures the ingress traffic count on the sub-interfaces.

```
switch# hardware counter feature subinterface in
```

- This command configures the ingress traffic count on the VLAN interface.

```
switch# hardware counter feature vlan-interface in
```

- These commands enable the QSFP+ interface in port *group 1* and SFP+ interfaces in port ***group 2*** on a DCS-7050Q-16 switch, display the port group status, and display interface status.

```
switch(config)# hardware port-group 1 select Et15/1-4
switch(config)# hardware port-group 2 select Et21-24
switch(config)# show hardware port-group

Portgroup: 1    Active Ports: Et17-20
Port            State
----------------------------------------
Ethernet17      ErrDisabled
Ethernet18      ErrDisabled
Ethernet19      ErrDisabled
Ethernet20      ErrDisabled
Ethernet15/1    Active
Ethernet15/2    Active
Ethernet15/3    Active
Ethernet15/4    Active

Portgroup: 2    Active Ports: Et16/1-4
Port            State
----------------------------------------
Ethernet16/1    Active
Ethernet16/2    Active
Ethernet16/3    Active
Ethernet16/4    Active
Ethernet21      ErrDisabled
Ethernet22      ErrDisabled
Ethernet23      ErrDisabled
Ethernet24      ErrDisabled
switch(config)# show interfaces status
Port      Name              Status       Vlan         Duplex  Speed Type
Et1/1                       connected    in Po621       full    40G 40GBASE-CR4
Et1/2                       errdisabled  inactive     unconf unconf 40GBASE-CR4
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et15/1                      connected    in Po711       full    40G 40GBASE-CR4
Et15/2                      errdisabled  inactive     unconf unconf Not Present
Et15/3                      errdisabled  inactive     unconf unconf Not Present
Et15/4                      errdisabled  inactive     unconf unconf Not Present
Et16/1                      errdisabled  inactive     unconf unconf Not Present
Et16/2                      errdisabled  inactive     unconf unconf Not Present
Et16/3                      errdisabled  inactive     unconf unconf Not Present
Et16/4                      errdisabled  inactive     unconf unconf Not Present
Et17                        errdisabled  inactive     unconf unconf Not Present
```

```
Et18                        errdisabled  inactive    unconf unconf Not Present
Et19                        errdisabled  inactive    unconf unconf Not Present
Et20                        errdisabled  inactive    unconf unconf Not Present
Et21                        connected    425           full    10G 10GBASE-SRL
Et22                        connected    611           full    10G 10GBASE-SRL
Et23                        connected    in Po998      full    10G 10GBASE-SLR
Et24                        connected    in Po998      full    10G 10GBASE-SLR
switch(config)#
```

- These commands enable the QSFP+ interface in port ***group 1*** **and SFP+ interfaces in port** ***group 2*** on a DCS-7050Q-16 switch, display the port group status, and display interface status.

```
switch(config)# hardware port-group 1 select Et15/1-4
switch(config)# hardware port-group 2 select Et21-24

switch(config)# show hardware port-group

Portgroup: 1    Active Ports: Et17-20
Port            State
-----------------------------------------
Ethernet17      ErrDisabled
Ethernet18      ErrDisabled
Ethernet19      ErrDisabled
Ethernet20      ErrDisabled
Ethernet15/1    Active
Ethernet15/2    Active
Ethernet15/3    Active
Ethernet15/4    Active

Portgroup: 2    Active Ports: Et16/1-4
Port            State
-----------------------------------------
Ethernet16/1    Active
Ethernet16/2    Active
Ethernet16/3    Active
Ethernet16/4    Active
Ethernet21      ErrDisabled
Ethernet22      ErrDisabled
Ethernet23      ErrDisabled
Ethernet24      ErrDisabled
switch(config)# show interfaces status
Port      Name              Status       Vlan        Duplex  Speed Type
Et1/1                       connected    in Po621      full    40G 40GBASE-CR4
Et1/2                       errdisabled  inactive    unconf unconf 40GBASE-CR4
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et15/1                      connected    in Po711      full    40G 40GBASE-CR4
Et15/2                      errdisabled  inactive    unconf unconf Not Present
Et15/3                      errdisabled  inactive    unconf unconf Not Present
Et15/4                      errdisabled  inactive    unconf unconf Not Present
Et16/1                      errdisabled  inactive    unconf unconf Not Present
Et16/2                      errdisabled  inactive    unconf unconf Not Present
Et16/3                      errdisabled  inactive    unconf unconf Not Present
Et16/4                      errdisabled  inactive    unconf unconf Not Present
Et17                        errdisabled  inactive    unconf unconf Not Present
Et18                        errdisabled  inactive    unconf unconf Not Present
Et19                        errdisabled  inactive    unconf unconf Not Present
Et20                        errdisabled  inactive    unconf unconf Not Present
Et21                        connected    425           full    10G 10GBASE-SRL
Et22                        connected    611           full    10G 10GBASE-SRL
Et23                        connected    in Po998      full    10G 10GBASE-SLR
Et24                        connected    in Po998      full    10G 10GBASE-SLR
switch(config)#
```

11.1.7.9 hardware port-group

The **`hardware port-group`** command configures a port group to activate a 40GBASE (QSFP+) interface or four 10GBASE (SFP+) interfaces, affecting QSFP+ and SFP+ availability.

The **`no hardware port-group`** and **`default hardware port-group`** commands restore a port groups default setting by removing the corresponding **`hardware port-group`** command from ***running-config***. The QSFP+ interface is active by default in each port group.

The **`hardware port-group`** command is available on DCS-7050Q-16 and DCS-7050QX-32S switches, and has different parameters on each platform.

Command Mode

Global Configuration

Command Syntax

`hardware port-group` ***group_number*** **`select`** **PORT_LIST**

`no hardware port-group` ***group_number*** **`select`** **PORT_LIST**

`default hardware port-group` ***group_number*** **`select`** **PORT_LIST**

Parameters

- ***group_number*** Label of the port group. Valid options are ***1*** and ***2*** on the 7050Q-16; only ***1*** is available on the 7050QX-32S.
- **PORT_LIST** Ports activated by command. Options vary by platform and depend on ***group_number*** value.
  - DCS-7050Q-16
    - ***Et15/1-4*** activates QSFP+ port on port ***group 1***. Available when ***group_number*** is ***1***.
    - ***Et16/1-4*** activates QSFP+ port on port ***group 2***. Available when ***group_number*** is ***2***.
    - ***Et17-20*** activates SFP+ ports on port ***group 1***. Available when ***group_number*** is ***1***.
    - ***Et21-23*** activates SFP+ ports on port ***group 2***. Available when ***group_number*** is ***2***.
  - DCS-7050QX-32S
    - ***Et1-4*** activates SFP+ ports on port ***group 1***. Available when *group_number* is ***1***.
    - ***Et5/1-4*** activates QSFP+ port on port ***group 1***. Available when ***group_number*** is ***1***.

11.1.7.10 hardware tcam

The **`hardware tcam`** command places the switch in Hardware TCAM configuration mode.

The **`exit`** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

`hardware tcam`

Related Commands

- profile
- interface ethernet

Example

This command places the switch in Hardware TCAM configuration mode.

```
switch(config)# hardware tcam
switch(config-hw-tcam)#
```

11.1.7.11 interface ethernet create

The **`interface ethernet create`** command is used to configure a range of Ethernet subinterfaces. The command places the switch in Ethernet-interface configuration mode for the specified range of subinterfaces.

Command Mode

Global Configuration

Command Syntax

`interface ethernet create` ***sub_range***

Parameters

sub_range Range of subinterfaces to be configured. Subinterfaces are named by adding a period followed by a unique subinterface number to the name of the parent interface.

Example

This command enters interface configuration mode for Ethernet subinterfaces ***1/1.1-100***:

```
switch(config)# interface ethernet create 1/1.100
switch(config-if-Et1/1.1-100)#
```

11.1.7.12 interface ethernet

The **`interface ethernet`** command places the switch in Ethernet-interface configuration mode for the specified interfaces. The command can specify a single interface or multiple interfaces.

Ethernet interfaces are physical interfaces and are not created or removed.

Interface management commands include:

- **description**
- **exit**
- **load-interval**
- **mtu**
- **shutdown** (Interfaces)

Ethernet management commands include:

- **flowcontrol**
- **mac-address**
- **speed**

Chapters describing supported protocols and other features list additional configuration commands available from Ethernet interface configuration mode.

Command Mode

Global Configuration

Command Syntax

`interface ethernet` ***e_range***

Parameters

e_range Ethernet interfaces (number, range, or comma-delimited list of numbers and ranges). Valid Ethernet numbers depend on the switchs available Ethernet interfaces.

Examples

- This command enters interface configuration mode for Ethernet interfaces ***1*** and ***2***:

```
switch(config)# interface ethernet 1-2
switch(config-if-Et1-2)#
```

- This command enters interface configuration mode for ***interface ethernet 1***:

```
switch(config)# interface ethernet 1
switch(config-if-Et1)#
```

11.1.7.13 interface management

The **`interface management`** command places the switch in management-interface configuration mode for the specified interfaces. The list can specify a single interface or multiple interfaces if the switch contains more than one management interface.

Management interfaces are physical interfaces and are not created or removed.

Interface management commands include:

- **description**
- **exit**
- **load-interval**
- **mtu**
- **shutdown** (Interfaces)

Ethernet management commands include:

- **flowcontrol**
- **mac-address**
- **speed**

Chapters describing supported protocols and other features list additional configuration commands available from ***management-interface*** configuration mode.

Command Mode

Global Configuration

Command Syntax

`interface management` ***m_range***

Parameter

m_range Management interfaces (number, range, or comma-delimited list of numbers and ranges).

Valid management numbers depend on the switchs available management interfaces. A value of ***0***, where available, configures the virtual management interface on a dual-supervisor modular switch. Management interface ***0*** accesses management port ***1*** on the active supervisor of a dual-supervisor modular switch.

Examples

- This command enters the ***interface configuration*** mode for management interfaces ***1*** and ***2***.

```
switch(config)# interface management 1-2
switch(config-if-Ma1-2)#
```

- This command enters the ***interface configuration*** mode for management interface ***1***:

```
switch(config)# interface management 1
switch(config-if-Ma1)#
```

11.1.7.14 link-debounce

The **`link-debounce`** command configures the link debounce time for the configuration mode interface. Link debounce time is the time that advertisements for new link states are delayed after the link state is established. By default, debounce time is set to zero, disabling link debounce.

Debounce times for link-up and link-down transitions can be independently configured.

- **Link-up debounce time:** the delay before an interface advertises link down to link up transitions.
- **Link-down debounce time:** the delay before an interface advertises link up to link down transitions.

The **no link-debounce** and **default link-debounce** commands restore the default debounce setting for the configuration mode interface by removing the corresponding **link-debounce** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

link-debounce time WAIT_TIME

no link-debounce time WAIT_TIME

default link-debounce time WAIT_TIME

Parameters

WAIT_TIME Link debounce period (milliseconds). All debounce values range from 0 (disabled) to 30000 (30 seconds). Options include:

- ***0*** - ***30000*** One debounce value assigned as both link up and link down.
- ***0*** - ***30000 0*** - ***30000*** Two debounce values: link up is first, link down is second.

Examples

- These commands set the link-up and link-down debounce period to ***10*** seconds on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# link-debounce time 10000
switch(config-if-Et5)#
```

- These commands set the link-up debounce to ***10*** seconds and the link-down debounce period to zero on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# link-debounce time 10000 0
switch(config-if-Et5)#
```

- These commands set the link-up debounce to ***0*** and the link-down debounce period to ***12.5*** seconds on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# link-debounce time 0 12500
switch(config-if-Et5)#
```

11.1.7.15 link-error

The **link-error** command places the Ethernet Operations, Administration, and Management (EOAM) profile in the EOAM link-error sub-mode.

The **no link-error** and **default link-error** commands exit from the EOAM link-error sub-mode.

Command Mode

EOAM Configuration

Command Syntax

link-error

no link-error

default link-error

Related Commands

- monitor ethernet oam profile
- show monitor ethernet oam profile

Example

These commands place the EOAM profile ***profile1*** in the ***link-error*** sub-mode.

```
switch(config)# monitor ethernet oam
switch(config-eoam)# profile profile1
switch(config-eoam-profile-profile1)# link-error
switch(config-eoam-profile-profile1-link-error)#
```

11.1.7.16 mac-address

The **mac-address** command assigns a MAC address to the configuration mode interface. An interfaces default MAC address is its burn-in address.

The **no mac-address** and **default mac-address** commands revert the interface to its default MAC address by removing the corresponding **mac-address** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

mac-address ***address***

no mac-address

default mac-address

Parameter

address MAC address assigned to the interface. Format is dotted hex notation (***H.H.H***). Disallowed addresses are ***0.0.0*** and ***FFFF.FFFF.FFFF***.

Example

This command assigns the MAC address of ***001c.2804.17e1*** to **interface ethernet 7**, then displays interface parameters, including the assigned address.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# mac-address 001c.2804.17e1
switch(config-if-Et7)# show interface ethernet 7
Ethernet3 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2804.17e1 (bia 001c.7312.02e2)
  Description: b.e45
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 seconds input rate 7.84 kbps (0.0% with framing), 10 packets/sec
  5 seconds output rate 270 kbps (0.0% with framing), 24 packets/sec
     1363799 packets input, 222736140 bytes
     Received 0 broadcasts, 290904 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     2264927 packets output, 2348747214 bytes
     Sent 0 broadcasts, 28573 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
```

```
     0 PAUSE output
switch(config-if-Et7)#
```

11.1.7.17 monitor ethernet oam profile

The **monitor ethernet oam profile** command applies the EOAM profile to the specific interface in interface configuration mode.

The **no monitor ethernet oam profile** and **default monitor ethernet oam profile** commands remove the EOAM profile from the interface.

Command Mode

Interface Configuration

Command Syntax

monitor ethernet oam profile ***name***

no monitor ethernet oam profile

default monitor ethernet oam profile

Parameters

name The EOAM profile name. An EOAM profile cannot be named as summary.

Related Commands

- link-error
- show monitor ethernet oam profile

Example

These commands apply the EOAM profile ***profile1*** to the **interface ethernet 1/1**.

```
switch(config)# interface ethernet 1/1
switch(config-if-Et1/1)# monitor ethernet oam profile profile1
```

11.1.7.18 monitor ethernet oam

The **monitor ethernet oam** command places the switch in the Ethernet Operations, Administration, and Management (EOAM) configuration mode.

The **no monitor ethernet oam** and **default monitor ethernet oam** commands exit from the EOAM configuration mode.

Command Mode

Global Configuration

Command Syntax

monitor ethernet oam

no monitor ethernet oam

default monitor ethernet oam

Example

This command places the switch in the ***EOAM*** configuration mode.

```
switch(config)# monitor ethernet oam
switch(config-eoam)#
```

11.1.7.19 period

The **period** command configures the link monitoring period that is specified for a link error in terms of number of frames or seconds.

The **no period** command removes the period type specified on the chosen link error. The **default period** command configures the link monitoring period as zero seconds.

Command Mode

Link-error Configuration

Command Syntax

{fcs | symbol} **period** num {***seconds*** | ***frames***}

no {fcs | symbol} **period** num {***seconds*** | ***frames***}

default {fcs | symbol} **period** num {***seconds*** | ***frames***}

Parameters

- **fcs** Inbound packets with Frame Check Sequence (FCS) error.
- **symbol** Inbound packets with symbol error.
- **num** The link monitoring period in frames or seconds. The frames value ranges from ***1*** to ***4000000000***. The seconds value ranges from ***2*** to ***200*** seconds. The default value is ***2*** seconds.
  - ***seconds*** The monitor errors per num seconds.
  - ***frames*** The monitor errors per num frames.

Related Commands

- action
- threshold

Example

These commands set the frames period type for the profile ***profile1*** in the Link-error configuration mode for ***300*** frames.

```
switch(config)# monitor ethernet oam
switch(config-eoam)# profile profile1
switch(config-eoam-profile-profile1)# link-error
switch(config-eoam-profile-profile1-link-error)# symbol period 300 frames
```

11.1.7.20 phy link detection aggressive

The **phy link detection aggressive** command allows configuration of interfaces for aggressive link-up declaration (~***50*** ms) on those interfaces. **no/default phy link detection aggressive** configuration reverts to a more reliable link-up with at least a two-second delay. When the aggressive mode is not supported, the system generates an advisory message.

Command Mode

Global Configuration

Command Syntax

phy link detection aggressive

no phy link detection aggressive

default phy link detection aggressive

Examples

- This command configures aggressive link detection.

```
switch(config-if-Et1)# phy link detection aggressive
switch(config-if-Et1)#
```

- This command configures aggressive link detection but fails on a platform that does not support the feature.

```
switch(config-if-Et1)# phy link detection aggressive
detection     detection not supported on this hardware platform
switch(config-if-Et1)#
```

11.1.7.21 poe disabled

Power over Ethernet (PoE) is enabled on all Ethernet ports by default on switches that support PoE. The **poe disabled** command disables PoE on the configuration-mode interface.

The **no poe disabled** and **default poe disabled** commands restore PoE on the interface by removing the corresponding **poe disabled** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

poe disabled

no poe disabled

default poe disabled

Example

These commands disable PoE on ***interface ethernet 7***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# poe disabled
switch(config-if-Et7)#
```

11.1.7.22 poe legacy detect

IEEE-compliant Powered Devices (PDs) are recognized by a specific resistance signature to a test signal sent by the switch, but non-compliant (legacy or proprietary) PDs may use a capacitive signature instead. The **poe legacy detect** command causes the configuration-mode interface to attempt to use hardware detection for these non-compliant PoE devices and power them. By default, legacy PD detection is disabled, and legacy devices are not powered.

Note: Non IEEE-compliant PDs are not officially supported. Arista cannot guarantee compatibility with such devices, and they may not be detected even when legacy detection is enabled on the port they are connected to.

The **no poe legacy detect** and **default poe legacy detect** commands restore the default behavior by removing the corresponding **poe legacy detect** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

poe legacy detect

no poe legacy detect

default poe legacy detect

Example

These commands configure ***interface ethernet 7*** to attempt to detect and power capacitive PDs.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#poe legacy detect
switch(config-if-Et7)#
```

11.1.7.23 poe limit

Power over Ethernet (PoE) power output is limited by the hardware-negotiated power level and by the total power capacity of the switch. The **poe limit** command sets an additional maximum power output for the configuration-mode interface. The power limit represents the power output at the Ethernet port; actual power delivered to the PD will be lower due to power loss along the Ethernet cable.

Note: If a power limit is set by this command, Power Via MDI TLVs will not be sent from the interface. See Configuring LLDP for Power over Ethernet for details.

The **no poe limit** and **default poe limit** commands restore the default power limitation by removing the corresponding **poe limit** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

poe limit {*class class_num* | *watt_num watts*}

no poe limit

default poe limit

Parameters

- ***class_num*** Specifies the power output limit by power class. Values range from 0-6 as follows:
  - Class 0 = 15.4 W
  - Class 1 = 4 W
  - Class 2 = 7 W
  - Class 3 = 15.4 W
  - Class 4 = 30 W
  - Class 5 = 45 W
  - Class 6 = 60 W
- ***watt_num*** Specifies the power output limit in watts. Values range from ***0-60***. A value of ***0*** watts prevents the port from providing PoE power.

Examples

- These commands limit nominal PoE power output on ***interface ethernet 7*** to ***10 W***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# poe limit 10 watts
switch(config-if-Et7)#
```

- These commands limit nominal PoE power output on ***interface ethernet 7*** to ***4 W***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# poe limit class 1
switch(config-if-Et7)#
```

11.1.7.24 power budget

Power over Ethernet (PoE) budget is not set on all Ethernet ports by default on switches that support PoE. The **power budget [n] watts** command sets a limit of 'n' watts available for use by the ports on the switch. Otherwise, available power for PoE is limited to the sum of all power provided by the PSUs minus the power reserved for the chassis.

The **no power budget** restores the default state.

Command Mode

Interface-Ethernet Configuration

Command Syntax

power budget [n] watts

no power budget

Example

The following command sets the PoE power budget to ***600W***.

```
switch(config)# power budget 600 watts
```

11.1.7.25 power budget exceed action (warning | hold-down)

When the Power over Ethernet (PoE) ports exceeds the budgeted amount, the switch issues a warning and continues to provide additional power above the budgeted limit. The **power budget exceed action hold-down** command causes the switch to issue a warning and limit the power to the configured limit.

The **no power budget** restores the default state.

Command Mode

Interface-Ethernet Configuration

Command Syntax

power budget exceed action hold-down

no power budget

Example

The following commands set the PoE power budget to ***600W*** and configure the switch to not exceed the limit.

```
switch(config)# power budget 600 watts
switch(config)# power budget exceed action hold-down
```

11.1.7.26 profile

The **profile** command creates an Ethernet Operations, Administration, and Management (EOAM) profile in the EOAM configuration mode.

The **no profile** and **default profile** commands exit from the EOAM configuration mode.

Command Mode

EOAM Configuration

Command Syntax

`profile` ***profile_name***

`no profile` ***profile_name***

`default profile` ***profile_name***

Parameter

profile_name The profile name that is specified.

Related Commands

- monitor ethernet oam profile
- show monitor ethernet oam profile

Guidelines

Run the **`shutdown`** or **`no shutdown`** command to bring the port back to the normal state.

Example

These commands create an EOAM profile ***profile1*** in the ***EOAM*** configuration mode.

```
switch(config)# monitor ethernet oam
switch(config-eoam)# profile profile1
switch(config-eoam-profile-profile1)#
```

11.1.7.27 qos scheduling

The **`qos scheduling`** command places the switch in the QoS scheduling mode under which the scheduling groups and scheduling policies are configured for shared shaper across multiple subinterfaces.

The **`no qos scheduling`** command removes the QoS scheduling configuration from the **`running-config`**.

Command Mode

QoS Scheduling Mode

Command Syntax

`qos scheduling`

`no qos scheduling`

Example

These commands create a scheduling policy with the desired shape rate and optional guaranteed bandwidth:

```
switch(config)# qos scheduling
switch(config-qos-scheduling)# scheduling policy P1
switch(config-qos-scheduling-policy-P1)# shape rate 75000000
switch(config-qos-scheduling-policy-P1)# bandwidth guaranteed 10000000
```

11.1.7.28 recovery-time

The **`recovery-time`** command configures the recovery timeout value for link fault signaling.

The **`no recovery-time`** command and the **`default recovery-time`** command removes the recovery timeout value specified for the chosen link error.

Command Mode

Link-error Configuration

Command Syntax

`recovery-time` ***value***

`no recovery-time` ***value***

`default recovery-time` ***value***

Parameters

value Specifies the recovery timeout value for LFS. The value ranges from ***20*** to ***200***.

Related Commands

- action
- period
- threshold

Example

These commands set the recovery time value of ***40*** for the ***profile profile1*** in the Link-error configuration mode.

```
switch(config)# monitor ethernet oam
switch(config-eoam)# profile profile1
switch(config-eoam-profile-profile1)# link-error
switch(config-eoam-profile-profile1-link-error)# recovery-time 40
```

11.1.7.29 show hardware counter

The **`show hardware counter`** command displays counter events across time intervals.

Command Mode

EXEC

Command Syntax

`show hardware counter`

Example

This command displays counter events across all time intervals, which are currently more than one standard deviation apart from a given time interval.

```
switch(config-handler-eventHandler1-counters)# show hardware counter events
-------------------------------------------------------------------------------------------
Interval | Event Name  | Chip | First  | Last  | Count | Z-Score
 |  | Name | Occurrence  | Occurrence  |  |
-------------------------------------------------------------------------------------------
5 Min  | MacCounters |  All | 2017-01-31 09:31:35 | 2017-01-31 09:44:32 |  5 | -6.9430
10 Min | MacCounters |  All | 2017-01-31 09:39:43 | 2017-01-31 09:44:32 |  3 | -4.8123
-------------------------------------------------------------------------------------------
switch(config-handler-eventHandler1-counters)#
```

11.1.7.30 show hardware port-group

The **`show hardware port-group`** command displays the status of DCS-7050Q-16 port-groups. Port groups contain one QSFP+ interface and a set of four SFP+ interfaces. In each port group, either the QSFP+ interface or the SFP+ interface set is enabled. The port groups are configured independent of each other.

- ***Port group 1*** contains ***interface 15*** (QSFP+) and ***interfaces 17-20*** (SFP+).
- ***Port group 2*** contains ***interface 16*** (QSFP+) and ***interfaces 21-24*** (SFP+).

Command Mode

EXEC

Command Syntax

`show hardware port-group`

Guidelines

The **`hardware port-group`** command is available on on DCS-7050Q-16 switches.

Example

This command displays the status of ports in the two port groups on a DCS-7050Q-16 switch.

```
switch# show hardware port-group

Portgroup: 1    Active Ports: Et15/1-4
Port            State
----------------------------------------
Ethernet17      ErrDisabled
Ethernet18      ErrDisabled
Ethernet19      ErrDisabled
Ethernet20      ErrDisabled
Ethernet15/1    Active
Ethernet15/2    Active
Ethernet15/3    Active
Ethernet15/4    Active

Portgroup: 2    Active Ports: Et16/1-4
Port            State
----------------------------------------
Ethernet16/1    Active
Ethernet16/2    Active
Ethernet16/3    Active
Ethernet16/4    Active
Ethernet21      ErrDisabled
Ethernet22      ErrDisabled
Ethernet23      ErrDisabled
Ethernet24      ErrDisabled switch>
```

11.1.7.31 show interfaces counters bins

The **`show interfaces counters bins`** command displays packet counters, categorized by packet length, for the specified interfaces. Packet length counters that the command displays include:

- 64 bytes
- 65-127 bytes
- 128-255 bytes
- 256-511 bytes
- 512-1023 bytes
- 1024-1522 bytes
- larger than 1522 bytes

Command Mode

EXEC

Command Syntax

`show interfaces` **[INTERFACE]** **`counters bins`**

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** All interfaces.

- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.

Related Commands

- show interfaces counters
- show interfaces counters errors
- show interfaces counters queue
- show interfaces counters rates

Example

This command displays packet counter results for ***interface ethernet 1*** and ***interface ethernet 2***.

```
switch#show interfaces ethernet 1-2 counters bins
Input
Port                64 Byte      65-127 Byte     128-255 Byte     256-511 Byte
-------------------------------------------------------------------------------
Et1                    2503         56681135          1045154          1029152
Et2                       8         50216275          1518179          1086297

Port          512-1023 Byte   1024-1522 Byte    1523-MAX Byte
-------------------------------------------------------------
Et1                  625825         17157823          8246822
Et2                  631173         27059077          5755101 switch>
```

11.1.7.32 show interfaces counters errors

The **`show interfaces counters errors`** command displays the error counters for the specified interfaces.

Command Mode

EXEC

Command Syntax

`show interfaces` **[INTERFACE]** **`counters errors`**

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** All interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.

Display Values

The table displays the following counters for each listed interface:

- **`FCS`**: Inbound packets with CRC error and proper size.
- **`Align`**: Inbound packets with improper size (undersized or oversized).
- **`Symbol`**: Inbound packets with symbol error and proper size.
- **`Rx`**: Total inbound error packets.
- **`Runts`**: Outbound packets that terminated early or dropped because of underflow.
- **`Giants`**: Outbound packets that overflowed the receiver and were dropped.
- **`Tx`**: Total outbound error packets.

Related Commands

- show interfaces counters
- show interfaces counters bins

- show interfaces counters queue
- show interfaces counters rates

Examples

This command displays the error packet counters on ***interface ethernet 1*** and ***interface ethernet 2***.

```
switch# show interfaces ethernet 1-2 counters errors
Port               FCS    Align    Symbol       Rx    Runts   Giants       Tx
Et1                  0        0         0        0        0        0        0
Et2                  0        0         0        0        0        0        0
 switch>
```

11.1.7.33 show interfaces counters queue

The **show interfaces counters queue** command displays the queue drop counters for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [INTERFACE] counters queue

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** All interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.

Related Commands

- show interfaces counters
- show interfaces counters bins
- show interfaces counters errors
- show interfaces counters rates

Example

This command displays the queue drop counters for ***interface ethernet 1*** and ***interface ethernet 2***.

```
switch# show interfaces ethernet 1-2 counters queue
Port                  InDrops
Et1                       180
Et2                       169 switch>
```

11.1.7.34 show interfaces counters rates

The **show interfaces counters rates** command displays the received and transmitted packet rate counters for the specified interfaces. Counter rates provided include megabits per second (Mbps), kilopackets per second (Kpps) and utilization percentage.

All port rates are approximately calculated. Note that, when displaying the rate information of a port channel, the rate value of the port channel differs from the sum of the rates for the member ports. The discrepancy is likely to be larger for port channels with fewer ports except for port channels with single ports. The rate values of individual member ports are less inaccurate than the rate values of the port channel as a whole.

Command Mode

EXEC

Command Syntax

`show interfaces` **[INTERFACE]** **`counters rates`**

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** All interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.

Related Commands

- show interfaces counters
- show interfaces counters bins
- show interfaces counters errors
- show interfaces counters queue

Example

This command displays rate counters for ***interface ethernet 1*** and ***interface ethernet 2***.

```
switch# show interfaces ethernet 1-2 counters rates
Port         Intvl   In Mbps      %  In Kpps  Out Mbps      % Out Kpps
Et1           0:05      53.3   0.5%        5      31.2   0.3%        2
Et2           0:05      43.3   0.4%        4       0.1   0.0%        0 switch#
```

11.1.7.35 show interfaces counters

The **`show interface counters`** command displays the Layer 3 ingress traffic count information. Run this command to view the traffic counts on a sub-interface or VLAN interface. The clear counters command resets the counters to zero. Counters displayed by the command include:

- inbound bytes
- inbound unicast packets
- inbound multicast packets
- inbound broadcast packets
- outbound bytes
- outbound unicast packets
- outbound multicast packets
- outbound broadcast packets

Command Mode

EXEC

Command Syntax

`show interfaces` [**INTERFACE**] **`counters`** [*incoming*]

Parameters

- **INTERFACE** Interface type and numbers. Options include:
  - ***no parameter*** All interfaces.
  - **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
  - **management *m_range*** Management interface range specified by ***m_range***.
  - **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.

- **subinterface** Displays the subinterface traffic counts.
- **vlan-interface** Displays the VLAN-interface traffic counts.

- ***incoming*** Displays the traffic count for the ingress port.

Note: When no interface is specified, the output starts with ingress and egress counters section for regular interfaces, followed by section for the ingress L3 Interface counters.

Related Commands

- show interfaces counters bins
- show interfaces counters errors
- show interfaces counters queue
- show interfaces counters rates

Examples

- This command displays byte and packet counters for ***interface ethernet 1*** and ***interface ethernet 2***

```
switch# show interfaces ethernet 1-2 counters
Port                InOctets     InUcastPkts     InMcastPkts     InBcastPkts
Et1               99002845169        79116358           75557            2275
Et2               81289180585        76278345           86422              11

Port               OutOctets    OutUcastPkts    OutMcastPkts    OutBcastPkts
Et1                4347928323         6085482          356173            2276
Et2                4512762190         5791718          110498              15 switch>
```

- This command displays the ingress traffic count on a ***VLAN interface vl12***.

```
switch# show interface vl12 counters incoming
L3 Interface InOctets InUcastPkts InMcastPkts
Vl12            3136          47            2
```

11.1.7.36 show interfaces flow-control

The **`show interfaces flow-control`** command displays administrative and operational flow control data for the specified interfaces. Administrative data is the parameter settings stored in ***running-config*** for the specified interface; the switch uses these settings to negotiate flow control with the peer switch. Operational data is the resolved flow control setting that controls the ports behavior.

Command Mode

EXEC

Command Syntax

`show` **[INTERFACE]** **`flow-control`**

Parameters

INTERFACE Interface type and number for which flow control data is displayed.

- ***no parameter*** All interfaces.
- **ethernet *e_range*** Ethernet interfaces in the specified range.
- **management *m_range*** Management interfaces in the specified range.

Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command shows the settings for Ethernet interfaces ***1-10***.

```
switch# show flow-control interface ethernet 1-10
Port       Send FlowControl  Receive FlowControl  RxPause       TxPause
```

```
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------------- -------------
Et1        off      off      off      off         0             0
Et2        off      off      off      off         0             0
Et3        off      off      off      off         0             0
Et4        off      off      off      off         0             0
Et5        off      off      off      off         0             0
Et6        off      off      off      off         0             0
Et7        off      off      off      off         0             0
Et8        off      off      off      off         0             0
Et9        off      off      off      off         0             0
Et10       off      off      off      off         0             0 switch#
```

11.1.7.37 show interfaces hardware default

The **show interfaces hardware default** command displays the static interface capability information of the specified interfaces. This command displays information related to the speed, auto-negotiation, error correction, and modulation capabilities (when applicable) of a system's ports. The command also provides information displayed by the show interfaces hardware command, such as model number, interface type, duplex mode, and flow control settings of the specific interface. Compared to the show interfaces hardware command, this command accounts for the capabilities of the system architecture only, and does not consider the capabilities of a transceiver.

Command Mode

EXEC

Command Syntax

show interfaces [**INTERFACE**] **hardware default**

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.

Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the static interface capability information at the default level.

```
switch> show interfaces hardware default
Ethernet1
  Model:     DCS-7020TR-48
  Type:      1000BASE-T
  Speed/Duplex: 100M/full,1G/full
  Flowcontrol:  rx-(off,on,desired),tx-(off)
  Autoneg CL28: 100M/full,1G/full
  Autoneg CL37: 1G/full switch>
```

- This command displays the static interface capability information for ***interface ethernet 4/1/1***

```
switch> show interfaces ethernet 4/1/1 hardware default
Ethernet4/1/1
  Model:     7500R2AK-36CQ-LC
  Type:      40GBASE-CR4
  Speed/Duplex: 1G/full,10G/full,25G/full,40G/full,50G/full,100G/full
  Flowcontrol:  rx-(off,on,desired),tx-(off)
  Autoneg CL28: 1G/full,10G/full
```

```
  Autoneg CL73:
IEEE:           25G/full,40G/full,100G/full
Consortium: 25G/full,50G/full
  Error Correction:
Reed-Solomon: 25G,50G,100G
Fire-code:   25G,50G
```

11.1.7.38 show interfaces hardware

The **show interfaces hardware** command displays the model number, interface type, duplex mode, and flow control settings of the specified interfaces. The capabilities command is available on Ethernet and management interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [INTERFACE] hardware

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** All interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.

Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the model number, interface type, duplex mode, and flow control settings for ***interface ethernet 2*** and ***interface ethernet 18***.

```
switch# show interfaces ethernet 2,18 hardware
Ethernet2
  Model:         DCS-7150S-64-CL
  Type:          10GBASE-CR
  Speed/Duplex: 10G/full,40G/full,auto
  Flowcontrol:  rx-(off,on,desired),tx-(off,on,desired)
Ethernet18
  Model:         DCS-7150S-64-CL
  Type:          10GBASE-SR
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch#
```

11.1.7.39 show interfaces interactions

The **show interfaces interactions** command aims to provide users a resource that explains various relationships between ethernet interfaces. It describes interactions in which a configuration on an interface causes another set of interfaces to become inactive or have reduced capabilities. Examples include a primary interface consuming subordinate interfaces to service a four-lane speed or platform restrictions that require four interfaces of a port to operate at the same speed.

Command Mode

EXEC

Command Syntax

show interfaces [***intf***] **interactions**

Parameters

intf Hardware Ethernet interface. You can restrict the output by interface name.

Examples

The information is displayed as a text dump. Each section of information appears on a separate indentation level. The first indent level is the interface, the second is the desired configuration, and the third is the list of interactions. For example, the output below describes the following:

1. If the user wants to configure ***Ethernet2/1*** for 40G, ***Ethernet2/2-4*** becomes inactive.
2. If the user wants to configure ***Ethernet2/1*** for 10G, it does not affect other interfaces.

```
switch# show interface et2/1 interactions
* = includes less than 10G speeds that the interface is capable of

Ethernet2/1:
  For speed 40G
    Ethernet2/2-4 become inactive
  For speed 10G*
    No interactions with other interfaces
```

The asterisk next to the 10G entry indicates that these same interactions apply for speeds lower than 10G that the interface supports. In other words, if this interface also supported 1G or 100M, the configuration does not affect other interfaces.

Types of Interactions

No Interactions

If there are no interactions for the given interfaces, the display shows **No interfaces interactions.** Depending on what configurations actually have interactions, the line could appear at a number of different indentation levels. The rules are:

1. If all specified interfaces have no interactions, print at the first indentation level.
2. If only some specified interfaces have no interactions and those interfaces have no interactions at any speeds, print at the second indentation level for those interfaces.
3. If only some specified interfaces have no interactions and those interfaces only have interactions at some speeds, print at the third indentation level for those speeds.

```
switch# show int et1,2 interactions
No interfaces interactions
switch# show int et1,2,5/1 interactions
* = includes less than 10G speeds that the interface is capable of

Ethernet1
  No interactions with other interfaces
Ethernet2
  No interactions with other interfaces
Ethernet5/1
  For speed 40G
    Ethernet5/2-4 become inactive
  For speed 10G*
    No interactions with other interfaces
```

Inactive Interfaces

If a configuration on an interface causes other interfaces to become inactive, a message similar to the ones bolded below appear. A common example of this is configuring a QSFP28 port for 100G, which results in the

/2,***/3***, and ***/4*** interfaces becoming inactive. Note that display information for inactive interfaces are included in the specified range, ignoring whether or not inactive interfaces are exposed.

```
switch#show interfaces et11/1,11/3 interactions
* = includes less than 10G speeds that the interface is capable of

Ethernet11/1:
  For speed 100G-4
    Ethernet11/2-4 become inactive
  For speed 50G-2
    Ethernet11/2,11/4 become inactive
    Ethernet11/3 is limited to 50G-2
  For speed 40G
    Ethernet11/2-4 become inactive
  For speed 25G
    Ethernet11/2-4 are limited to 25G
  For speed 10G*
    Ethernet11/2-4 are limited to 10G*
Ethernet11/3:
  For speed 50G-2
    Ethernet11/4 becomes inactive
    Primary interface Ethernet11/1 must be operating at 50G-2
  For speed 25G
    Primary interface Ethernet11/1 must be operating at 25G
  For speed 10G*
    Primary interface Ethernet11/1 must be operating at 10G*
```

Required Primary Interface Configuration

Some interface configurations require a particular primary interface configuration to function. These interactions are captured with messages similar to the ones bolded below.

```
switch#show interfaces et11/1,11/3 interactions
* = includes less than 10G speeds that the interface is capable of

Ethernet11/1:
  For speed 100G-4
    Ethernet11/2-4 become inactive
  For speed 50G-2
    Ethernet11/2,11/4 become inactive
    Ethernet11/3 is limited to 50G-2
  For speed 40G
    Ethernet11/2-4 become inactive
  For speed 25G
    Ethernet11/2-4 are limited to 25G
  For speed 10G*
    Ethernet11/2-4 are limited to 10G*
Ethernet11/3:
  For speed 50G-2
    Ethernet11/4 becomes inactive
    Primary interface Ethernet11/1 must be operating at 50G-2
  For speed 25G
    Primary interface Ethernet11/1 must be operating at 25G
  For speed 10G*
    Primary interface Ethernet11/1 must be operating at 10G*
```

Hardware Speed-Group Requirements

Some interface configurations require an additional speed-group configuration in order to operate correctly. If speed-group configurations are required, there will be a message displayed similar to the ones bolded below.

Note: A single speed-group may include more than one compatibility setting. Using the example below as reference, ***Ethernet1/1*** at 100G-4 and ***Ethernet2/1*** at 40G are compatible configurations as long as hardware speed-group 1 can include 50g and 10g rates simultaneously.

```
switch# show interfaces et1/1,2/1 interactions
* = includes less than 10G speeds that the interface is capable of

Ethernet1/1:
  For speed 100G-4
    Ethernet1/2-4 become inactive
    Hardware speed-group 1 must include 50g
  For speed 40G
    Ethernet1/2-4 become inactive
    Hardware speed-group 1 must include 10g
  For speed 25G
    Ethernet2/1,2/3 become inactive
    Ethernet1/3 is limited to 25G/10G
    Hardware speed-group 1 must include 25g
  For speed 10G
    Ethernet2/1,2/3 become inactive
    Ethernet1/3 is limited to 25G/10G
    Hardware speed-group 1 must include 10g
Ethernet2/1:
  For speed 100G-4
    Ethernet2/3 becomes inactive
    Ethernet1/1 must be operating at 100G-4/50G-2/40G
    Hardware speed-group 1 must include 50g
  For speed 40G
    Ethernet2/3 becomes inactive
    Ethernet1/1 must be operating at 100G-4/50G-2/40G
    Hardware speed-group 1 must include 10g
```

Compatible Parent Interface Configuration

The parent interface may be configured for any one of the list of compatible rates. Additionally, more than one interface may be required to be configured at a compatible rate. These interactions are captured with messages similar to the ones bolded below

Note: Using the example below as a reference, suppose that ***Ethernet1/1*** is configured for 100G-4 and that the goal is to configure ***Ethernet1/8*** to 50G-1. According to the display, ***Ethernet1/1*** is configured for a compatible rate. That said, in order for ***Ethernet1/1*** to operate at 100G-4, ***hardware speed-group 1*** must be configured to include the 25g compatibility setting. Additionally, hardware speed-group 1 must also include the 50g compatibility setting to enable ***Ethernet1/8*** to operate at 50G-1. Interactions on ***Ethernet1/5*** and ***Ethernet1/7*** must be considered as well (omitted from the example for brevity).

```
switch# show interfaces et1/1,1/8 interactions
* = includes less than 10G speeds that the interface is capable of

Ethernet1/1:
  Ethernet1/1-4/8 share 18 interface hardware resources
  For speed 400G-8
    Ethernet1/2-8 become inactive
    Hardware speed-group 1 must include 50g
  For speed 200G-4
    Ethernet1/2-4 become inactive
    Hardware speed-group 1 must include 50g
  For speed 100G-2
    Ethernet1/2 becomes inactive
    Hardware speed-group 1 must include 50g
  For speed 100G-4
```

```
    Ethernet1/2-4 become inactive
    Hardware speed-group 1 must include 25g
  For speed 50G-1
    Hardware speed-group 1 must include 50g
  For speed 50G-2
    Ethernet1/2 becomes inactive
    Hardware speed-group 1 must include 25g
  For speed 40G
    Ethernet1/2-4 become inactive
    Hardware speed-group 1 must include 10g
  For speed 25G
    Hardware speed-group 1 must include 25g
  For speed 10G
    Hardware speed-group 1 must include 10g
Ethernet1/8:
  For speed 50G-1
    Ethernet1/1 must be operating at 200G-4/100G-2/100G-4/50G-1/50G-2/40G/25
G/10G
    Ethernet1/5 must be operating at 100G-2/50G-1/50G-2/25G/10G
    Ethernet1/7 must be operating at 50G-1/25G/10G
    Hardware speed-group 1 must include 50g
```

Speed-limited Interfaces

Some interface configurations limit at what other interfaces can operate. If such limitations occur, a message displays similar to the ones bolded below.

```
switch# show interfaces et11/1,11/3 interactions
* = includes less than 10G speeds that the interface is capable of

Ethernet11/1:
  For speed 100G-4
    Ethernet11/2-4 become inactive
  For speed 50G-2
    Ethernet11/2,11/4 become inactive
    Ethernet11/3 is limited to 50G-2
  For speed 40G
    Ethernet11/2-4 become inactive
  For speed 25G
    Ethernet11/2-4 are limited to 25G
  For speed 10G*
    Ethernet11/2-4 are limited to 10G*
Ethernet11/3:
  For speed 50G-2
    Ethernet11/4 becomes inactive
    Primary interface Ethernet11/1 must be operating at 50G-2
  For speed 25G
    Primary interface Ethernet11/1 must be operating at 25G
  For speed 10G*
    Primary interface Ethernet11/1 must be operating at 10G*
```

Interfaces Sharing Logical Ports

Some interface ranges share logical port resources. If an interface shares logical ports with other interfaces, a message displays similar to the ones bolded below.



Note: There must be logical ports available for an interface to become operational. Not all interfaces sharing logical ports can be operational simultaneously. Inactive interfaces do not consume a resource.

```
switch# show interfaces et1/1,4/1 interactions
```

```
* = includes less than 10G speeds that the interface is capable of

Ethernet1/1:
  Ethernet1/1-4/8 share 18 interface hardware resources
  For speed 400G-8
    ...
Ethernet4/1:
  Ethernet1/1-4/8 share 18 interface hardware resources
  For speed 400G-8
    ...
```

11.1.7.40 show interfaces negotiation

The **show interfaces negotiation** command displays the speed, duplex, and flow control auto-negotiation status for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [**INTERFACE**] **negotiation** [**INFO_LEVEL**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.

  Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **INFO_LEVEL** Amount of information that is displayed. Options include:
  - ***no parameter*** Displays status and negotiated setting of local ports.
  - **detail** Displays status and negotiated settings of local ports and their peers.

Examples

- This command displays the negotiated status of ***management 1*** and ***management 2*** interfaces.

```
switch# show interface management 1-2 negotiation
Port       Autoneg            Negotiated Settings
           Status   Speed     Duplex    Rx Pause  Tx Pause
---------  -------  --------  --------  --------  --------
Ma1        success  100M      full      off       off
Ma2        success  auto      auto      off       off switch>
```

- This command displays the negotiated status of ***management 1*** interface and its peer interface.

```
switch# show interface management 1 negotiation detail
Management1 :

Auto-Negotiation Mode     10/100/1000 BASE-T (IEEE Clause 28)
Auto-Negotiation Status   Success

  Advertisements       Speed           Duplex     Pause
                       --------------- ---------- --------------------
      Local            10M/100M/1G     half/full  Disabled
      Link Partner     None            None       None

  Resolution           100Mb/s         full       Rx=off,Tx=off switch>
```

11.1.7.41 show interfaces phy

The **`show interfaces phy`** command displays physical layer characteristics for the specified interfaces.

Command Mode

EXEC

Command Syntax

`show interfaces` [**INTERFACE**] **`phy`** [**INFO_LEVEL**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** All interfaces.
- ***ethernet e_range*** Ethernet interfaces in specified range.

  Valid ***e_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **INFO_LEVEL** Amount of information that is displayed. Options include:
  - ***no parameter*** Command displays table that summarizes PHY data.
  - **detail** Command displays data block for each specified interface.

Examples

- This command summarizes PHY information for Ethernet interfaces ***1-5***.

```
switch# show interfaces ethernet 1-5 phy
Key:
   U    = Link up
   D    = Link down
   R    = RX Fault
   T    = TX Fault
   B    = High BER
   L    = No Block Lock
   A    = No XAUI Lane Alignment
   0123 = No XAUI lane sync in lane N

                                State    Reset
Port           PHY state       Changes    Count PMA/PMD PCS   XAUI
-------------- --------------- -------- -------- ------- ----- --------
Ethernet1      linkUp            14518     1750 U..     U.... U.......
Ethernet2      linkUp            13944     1704 U..     U.... U.......
Ethernet3      linkUp            13994     1694 U..     U.... U.......
Ethernet4      linkUp            13721     1604 U..     U.... U.......
Ethernet5      detectingXcvr         3        1               D..A0123 switch#
```

- This command displays detailed PHY information for ***interface ethernet 1***.

```
switch# show interfaces ethernet 1 phy detail
Current System Time: Mon Dec  5 11:32:57 2011
Ethernet1
                            Current State     Changes           Last Change
  PHY state                 linkUp              14523           0:02:01 ago
  HW resets                                      1751           0:02:07 ago
  Transceiver               10GBASE-SRL          1704           0:02:06 ago
  Transceiver SN            C743UCZUD
  Oper speed                10Gbps
  Interrupt Count                               71142
  Diags mode                normalOperation
  Model                     ael2005c
  Active uC image           microInit_mdio_SR_AEL2005C_28
  Loopback                  none
  PMA/PMD RX signal detect  ok                  11497           0:37:24 ago
  PMA/PMD RX link status    up                  11756           0:37:24 ago
  PMA/PMD RX fault          ok                  11756           0:37:24 ago
  PMA/PMD TX fault          ok                      0                 never
  PCS RX link status        up                   9859           0:02:03 ago
  PCS RX fault              ok                   9832           0:02:03 ago
  PCS TX fault              ok                    330           0:27:44 ago
  PCS block lock            ok                   9827           0:02:03 ago
  PCS high BER              ok                   8455           0:02:05 ago
  PCS err blocks            255                                 0:02:03 ago
```

```
  PCS BER                     16                   50092            0:02:05 ago
  XFI/XAUI TX link status     up                    1282             0:27:44 ago
  XFI/XAUI RX fault           ok                    585              0:27:44 ago
  XFI/XAUI TX fault           ok                    2142             0:02:05 ago
  XFI/XAUI alignment status   ok                    2929             0:02:05 ago
  XAUI lane 0-3 sync          (0123) = 1111         2932             0:02:05 ago
  XAUI sync w/o align HWM     0                                            never
  XAUI sync w/o align max OK  5
  XAUI excess sync w/o align  0                                            never
  Xcvr EEPROM read timeout                            46    4 days, 6:33:45 ago
  Spurious xcvr detection                             0                  never
  DOM control/status fail                             0
  I2C snoop reset             0
  I2C snoop reset (xcvr)      0
  Margin count                5                last > 0            0:00:00 ago
  EDC resets                  1                                    0:02:03 ago
  EDC FFE0 - FFE11            -4 -5 57 -6 -6 -2 1 0 -2 -1 1 -1
  EDC FBE1 - FBE4             6 -1 5 -1
  EDC TFBE1 - TFBE4           1 2 1 2
  EDC VGA1, VGA3              12 115
  TX path attenuation         3.0 dB
  TX preemphasis              (0,63,4) (pre,main,post)
switch#
```

11.1.7.42 show interfaces status errdisabled

The **`show interfaces status errdisabled`** command displays interfaces that are in errdisabled state, including their link status and errdisable cause.

Command Mode

EXEC

Command Syntax

`show interfaces` [**INTERFACE**] status errdisabled

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.

  Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the error-disabled ports.

```
switch# show interfaces status errdisabled

Port    Name            Status        Reason
------- --------------- ------------- ------------------
Et49/2                  errdisabled   multi-lane-intf
Et49/3                  errdisabled   multi-lane-intf
Et49/4                  errdisabled   multi-lane-intf switch>
```

11.1.7.43 show interfaces status

The **`show interfaces status`** command displays the interface name, link status, vlan, duplex, speed, and type of the specified interfaces. When the command includes a link status, the results are filtered to display only interfaces whose link status match the specified type.

Command Mode

EXEC

Command Syntax

`show interfaces` [**INTERFACE**]**`status`** [**STATUS_TYPE**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** All existing interfaces.
- **ethernet *e_range*** Ethernet interfaces in the specified range.
- **management *m_range*** Management interfaces in the specified range.
- **port-channel *p_range*** All existing port-channel interfaces in the specified range.

  Valid ***e_range***, ***m_range***, and ***p_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **STATUS_TYP** Einterface status upon which the command filters output. Options include:
  - ***no parameter*** Command does not filter on interface status.
  - **connected** Interfaces connected to another port.
  - **notconnect** Unconnected interfaces that are capable of connecting to another port.
  - **disabled** Interfaces that have been powered down or disabled.
  - **sub-interfaces** L3 subinterfaces configured on the switch.

    Command may include multiple status types (**connected, notconnect, disabled**), which can be placed in any order.

Examples

- This command displays the status of Ethernet interfaces ***1-5***.

```
switch# show interfaces ethernet 1-5 status
Port      Name               Status       Vlan         Duplex  Speed Type
Et1                          connected    1              full    10G 10GBASE-
SRL
Et2                          connected    1              full    10G 10GBASE-
SRL
Et3                          connected    1              full    10G 10GBASE-
SRL
Et4                          connected    1              full    10G 10GBASE-
SRL
Et5                          notconnect   1              full    10G Not
 Present switch>
```

- This command displays status information for all subinterfaces configured on the switch.

```
switch# show interfaces status sub-interfaces
Port       Name    Status       Vlan     Duplex Speed  Type
 Flags
Et1.1              connect       101      full   10G   dot1q-encapsulation
Et1.2              connect       102      full   10G   dot1q-encapsulation
Et1.3              connect       103      full   10G   dot1q-encapsulation
Et1.4              connect       103      full   10G   dot1q-encapsulation
 switch>
```

11.1.7.44 show interface transceiver dom

The **`show interface [<intf>] transceiver dom`** command displays the most important current performance data on the media (line) side.

Example

```
switch# show interface Ethernet11/1 transceiver dom
Ch: Channel, N/A: not applicable, TX: transmit, RX: receive
mA: milliamperes, dBm: decibels (milliwatts), C: Celsius, V: Volts

Port 11
Last update: 0:00:05 ago
                                          Value
                                        ---------------
   Case temperature                        66.59 C
   Voltage                                  3.26 V
   TX power                               -10.23 dBm
   RX total power                         -11.61 dBm
   RX channel power                       -11.94 dBm
   Pre-FEC BER                           1.82e-03
   Post-FEC errored frames ratio         0.00e+00
   Chromatic dispersion (short link)         0.00 ps/nm
   Chromatic dispersion (long link)          0.00 ps/nm
   Differential group delay                  9.31 ps
   SOPMD                                     0.00 ps^2
   Polarization dependent loss               0.40 dB
   Received OSNR estimate                   35.10 dB
   Received ESNR estimate                   17.50 dB
   Carrier frequency offset                  0.00 MHz
   Error vector magnitude                  100.00 %
   SOP rate of change                        0.00 krad/s
   Laser temperature                        59.54 C
   Laser frequency                         193100.00 GHz
```

- BER: Bit Error Rate
- FEC: Forward Error Correction
- OSNR: Optical Signal to Noise Ratio
- ESNR: Electrical Signal to Noise Ratio
- SOP: State of Polarization
- SOPMD: State of Polarization Mode Dispersion

11.1.7.45 show interface transceiver eeprom

The **show interface [<intf>] transceiver eeprom** command displays the parsed capabilities.

Example

For 400GBASE-ZR, parsing of frequency tuning, power tuning (page 04h) and VDM configuration pages (20h-23h) is added.

```
switch# show interface Ethernet15/1 transceiver eeprom
Ethernet15 EEPROM:
...
  Frequency tuning support (04h:128-129):
    Grid spacing capabilities (04h:128):
      100 GHz grid supported (04h:128): true
      12.5 GHz grid supported (04h:128): false
      25 GHz grid supported (04h:128): false
      3.125 GHz grid supported (04h:128): false
      33 GHz grid supported (04h:128): false
      50 GHz grid supported (04h:128): false
      6.25 GHz grid supported (04h:128): false
      75 GHz grid supported (04h:128): true
    Tunable wavelength (04h:128): true
    Fine tuning support (04h:129): false
```

```
  Supported channel boundaries (04h:130-161):
    100 GHz grid (04h:150-153):
      Lowest channel (04h:150-151): -18
      Lowest frequency (04h:150-151): 191300000 MHz
      Highest channel (04h:152-153): 30
      Highest frequency (04h:152-153): 196100000 MHz
    75 GHz grid (04h:158-161):
      Lowest channel (04h:158-159): -72
      Lowest frequency (04h:158-159): 191300000 MHz
      Highest channel (04h:160-161): 120
      Highest frequency (04h:160-161): 196100000 MHz
  Programmable output power advertisement (04h:196-201):
    Lane programmable output power supported (04h:196): false
  VDM configuration (20h:128-255;21h:128-255):
    VDM group 1 (20h:128-255):
      Parameter 1 (20h:128-129):
        Lane (20h:128): 0
        Threshold ID (20h:128): 0
        Parameter type (20h:129): Laser temperature
      Parameter 3 (20h:132-133):
        Lane (20h:132): 0
        Threshold ID (20h:132): 2
        Parameter type (20h:133): eSNR host input
      Parameter 4 (20h:134-135):
        Lane (20h:134): 1
        Threshold ID (20h:134): 2
        Parameter type (20h:135): eSNR host input
      Parameter 5 (20h:136-137):
        Lane (20h:136): 2
        Threshold ID (20h:136): 2
        Parameter type (20h:137): eSNR host input
...
       Parameter 84 (21h:166-167):
        Lane (21h:166): 0
        Threshold ID (21h:166): 14
        Parameter type (21h:167): MER
  Number of VDM groups supported (2Fh:128): 2
```

11.1.7.46 show interfaces transceiver channels

The **`show interfaces transceiver channels`** command displays current wavelength/frequency settings for the specified channels.

Command Mode

EXEC

Command Syntax

`show interfaces` [**INTERFACE e_range**] **`transceiver channels`**

Parameters

INTERFACE Interface type and port numbers.

- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.

Related Commands

- transceiver channel
- show interfaces transceiver hardware

Example

This command displays the supported wave lengths/frequencies and their corresponding channel numbers on ***Ethernet interface 4*** to ***slot 3*** through ***4***.

```
switch(config-as-if-Et4/1/3)#show interfaces ethernet 4/3/4 transceiver
channels
Name: Et4/3/4
100GHz- 50GHz-
Wavelength Frequency spacing spacing
(nm) (GHz) Channel Channel
---------- --------- ------- -------
1567.95 191,200 1 1
1567.54 191,250 2
1567.13 191,300 2 3
1566.72 191,350 4
....
1529.16 196,050 98
1528.77 196,100 50 99
1528.38 196,150 100
switch(config-as-if-Et4/1/3)#
```

11.1.7.47 show interfaces transceiver hardware

The **`show interfaces transceiver hardware`** command displays current wavelength/frequency settings for the specified transceiver interfaces.

Command Mode

EXEC

Command Syntax

`show interfaces` [**INTERFACE**] ethernet ***e_range*** **`transceiver hardware`**

Parameters

INTERFACE Interface type and port numbers.

- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.

Related Commands

- transceiver channel
- show interfaces transceiver channels

Example

This command displays the current wavelength/frequency settings on ***interface ethernet 4*** to ***slot 3*** through ***4***.

```
switch(config-as-if-Et4/1/3)# show interfaces ethernet 4 / 3 / 4 transceiver
 hardware
Name: Et4/3/4
Media Type: 10GBASE-DWDM
Configured Channel : 39
Configured Grid (GHz) : 50
Computed Frequency (GHz) : 193,100
Computed Wavelength (nm) : 1552.52
Operational Channel : 39 (Default)
Operational Grid (GHz) : 50 (Default)
Operational Frequency (GHz): 193,100
Operational Wavelength (nm): 1552.52
switch(config-as-if-Et4/1/3)#
```

11.1.7.48 show interfaces transceiver properties

The **show interfaces transceiver properties** command displays configuration information for the specified interfaces. Information provided by the command includes the media type, interface speed-duplex settings, speed-duplex operating state.

Command Mode

EXEC

Command Syntax

show interfaces [**INTERFACE**] **transceiver properties**

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.

  Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Related Commands

show interfaces transceiver

Example

This command displays the media type, speed, and duplex properties for Ethernet interfaces ***1-3***.

```
switch# show interfaces ethernet 1-3 transceiver properties
Name : Et1
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL

Name : Et2
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL

Name : Et3
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL switch>
```

11.1.7.49 show interfaces transceiver

The **show interfaces transceiver** command displays operational transceiver data for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [**INTERFACE**] **transceiver** [**DATA_FORMAT**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** All interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.

  Valid ***e_range***, and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **DATA_FORMAT** format used to display the data. Options include:
  - ***no parameter*** table entries separated by tabs.
  - **csv** table entries separated by commas.

Related Commands

show interfaces transceiver properties

Example

This command displays transceiver data on ***interface ethernet 1*** through ***interface ethernet 4***.

```
switch# show interfaces ethernet 1-4 transceiver
If device is externally calibrated, only calibrated values are printed.
N/A: not applicable, Tx: transmit, Rx: receive.
mA: milliamperes, dBm: decibels (milliwatts).
                              Bias      Optical   Optical
        Temp       Voltage    Current   Tx Power  Rx Power  Last Update
Port    (Celsius)  (Volts)    (mA)      (dBm)     (dBm)     (Date Time)
-----   ---------  --------   --------  --------  --------  -------------------
Et1      34.17      3.30       6.75     -2.41     -2.83     2011-12-02 16:18:48
Et2      35.08      3.30       6.75     -2.23     -2.06     2011-12-02 16:18:42
Et3      36.72      3.30       7.20     -2.02     -2.14     2011-12-02 16:18:49
Et4      35.91      3.30       6.92     -2.20     -2.23     2011-12-02 16:18:45 switch#
```

11.1.7.50 show monitor ethernet oam profile

The **show monitor ethernet oam profile** command displays configuration information for the specified ethernet OAM profile name or the summary information of all configured profile names.

Command Mode

EXEC

Command Syntax

show monitor ethernet oam profile [***name*** | summary]

Parameters

- ***name*** The EOAM profile name.
- **summary** The EOAM summary of all profiles that are configured.

Related Commands

- link-error
- monitor ethernet oam profile

Examples

- This command displays the OAM profile configuration information for the specific profile name.

```
switch# show monitor ethernet oam profile[name]

          Ethernet OAM Profile : p
```

```
         Error Type : symbol

                          Threshold  :    20  frames
                          Action     :          log
                          Period     :    20 seconds

         Error Type : fcs
                          Threshold  :    10  frames
                          Action     :     linkfault
                          Period     :   100  frame

         Recovery Timeout : 20
```

- This command displays the OAM profile configuration summary for all profiles configured.

```
switch# show monitor ethernet oam profile [name] summary

         Eoam Profile : p
            Configured on:  Et3/1-4,5
```

11.1.7.51 show platform fm6000 agileport map

The **show platform fm6000 agileport map** command displays the list of Ethernet interfaces that are combinable to form a higher speed port.

Command Mode

Privileged EXEC

Command Syntax

show platform fm6000 agileport map

Example

These commands displays the agile port map for the switch, then configures Ethernet interface ***13*** as a 40G port, subsuming Ethernet interfaces ***15***, ***17*** and ***19***.

```
switch# show platform fm6000 agileport map

-----------------------------------------------------------------
Agile Ports       |        Interfaces subsumed in 40G link
-----------------------------------------------------------------
Ethernet1         |  Ethernet3      Ethernet5      Ethernet7
Ethernet2         |  Ethernet4      Ethernet6      Ethernet8
Ethernet13        |  Ethernet15     Ethernet17     Ethernet19
Ethernet14        |  Ethernet16     Ethernet18     Ethernet20

switch# config
switch(config)# interface ethernet 13
switch(config-if-Et13)# speed 40gfull

  WARNING!  Executing this command will cause the forwarding agent
            to be restarted. All interfaces will briefly drop links
            and forwarding on all interfaces will momentarily stop.


            Do you wish to proceed with this command? [y/N]

Ethernet13 configured for 40G.
Ethernet15, Ethernet17 and Ethernet19 are now subsumed.
switch(config-if-Et13)#
```

11.1.7.52 show platform trident flexcounters

Use the **show platform trident flexcounters** command to display the L3 interfaces when the corresponding hardware counter feature is enabled.

Command Mode

Privileged EXEC

Command Syntax

show platform trident flexcounters [vni [egress summary | ingress summary]]|[[vtep [decap summary | encap summary]]

Parameters

- **vni** Displays the VNI feature state.
  - **egress summary** Displays the egress direction flexcounters summary.
  - **ingress summary** Displays the ingress direction flexcounters summary.
- **vtep** Displays the VTEP feature state
  - **decap summary** Displays the decap direction flexcounter summary.
  - **encap summary** Displays the encap direction flexcounter summary.

Examples

- The following example displays the headings for the **show platform tridant flexcounters vni egress summary** show command.

```
switch# show platform tridant flexcounters vni egress summary
VNI      EgrVfiIndex    PoolId    OffsetMode    BaseCounterIndex
-----------------------------------------------------------------
```

- The following example displays the headings for the **show platform trident flexcounters vni ingress summary** show command.

```
switch# show platform trident flexcounters vni ingress summary
VNI      VfiIndex    PoolId    OffsetMode    BaseCounterIndex
-----------------------------------------------------------
```

- The following example displays the headings for the **show platform trident flexcounters vtep decap summary** show command.

```
switch# show platform trident flexcounters vtep decap summary
VTEP      SvpIndex    PoolId    OffsetMode    BaseCounterIndex
-----------------------------------------------------------
```

- The following example displays the headings for the **show platform trident flexcounters vtep encap summary** show command.

```
switch# show platform trident flexcounters vtep encap summary
VTEP      DvpIndex    PoolId    OffsetMode    BaseCounterIndex
-----------------------------------------------------------
```

11.1.7.53 show platform trident flexcounters l3intf

Use the **show platform trident flexcounters l3intf** command to display the subinterface, SVI, and GRE tunnel interface features.

Command Mode

EXEC

Command Syntax

`show platform trident flexcounters l3intf` [[in | out] [summary | values]]

Parameters

- **in summary** Displays the details of the hardware resources allocated to the ingress subinterface, SVI, and GRE Tunnel Interface features.
- **out summary** Displays the details of the hardware resources allocated to the egress subinterface, SVI, and GRE Tunnel Interface features.
- **in values** Displays the counters in the counter and snapshot table for the ingress subinterface, SVI, and GRE Tunnel Inteface features.
- **out values** Displays the counters in the counter and snapshot table for the egress subinterface, SVI, and GRE Tunnel Interface features.



Note: The counter values in this show command output are not reset by the **`clear counters`** command.

Examples

- Use the **`show platform trident flexcounters l3intf in summary`** command to display the details of the hardware resources allocated to the ingress subinterface, SVI, and GRE Tunnel Interface features.

```
switch# show platform trident flexcounters l3intf in summary
L3intf     CounterIndex    PoolId    OffsetMode   BaseCounterIndex
-------------------------------------------------------------------
```

- Use the **`show platform trident flexcounters l3intf out summary`** command to display the details of the hardware resources allocated to the egress subinterface, SVI, and GRE Tunnel Interface features.

```
switch# show platform trident flexcounters l3intf out summary
L3intf     CounterIndex    PoolId    OffsetMode   BaseCounterIndex
-------------------------------------------------------------------
```

- Use the **`show platform trident flexcounters l3intf in values`** command to display the counters in the counter and snapshot table for the ingress subinterface, SVI, and GRE Tunnel Interface features.

```
switch# show platform trident flexcounters l3intf in values
L3intf   Offser Bytes(cntTbl)    Bytes(snapTbl)    Pkts(cntTbl)
 Pkts(snapTbl)
-------------------------------------------------------------------
-----------------
```

- Use the **`show platform trident flexcounters l3intf out values`** command to display the counters in the counter and snapshot table for the egress subinterface, SVI, and GRE Tunnel Interface features.

```
switch# show platform trident flexcounters l3intf out values

L3intf   Offser Bytes(cntTbl)    Bytes(snapTbl)    Pkts(cntTbl)
 Pkts(snapTbl)
-------------------------------------------------------------------
----------------
```

11.1.7.54 show poe

The **`show poe`** command displays PoE information for a specified port range or for all ports.

Command Mode

Privileged EXEC

Command Syntax

show poe [**INTERFACE**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.

Example

This command displays PoE information for ***interface ethernet 46***.

```
switch(config)# show poe interface ethernet 46

Port Enabled Enabled Limit  Power   State   Class  Power Current Voltage Temperature
---- ------- ------- ------ ------- ------- ------ ----- ------- ------- -----------
46      True    True 15.40W  15.40W powered class0 1.40W 27.00mA  55.04V      41.25C
switch(config-if-Et7)#
```

11.1.7.55 show qos scheduling

The **show qos scheduling** command displays the QoS configuration on one or more scheduling groups. Both group name and parent interface name are optional, in which case all groups will be displayed.

Command Mode

Privileged EXEC

Command Syntax

show qos scheduling Options

Parameter

Options include the Group, Interface or the Hierarchy.

Example

This command displays the QoS configuration for scheduling group G1 of interface ***Ethernet1***.

```
switch# show qos scheduling group G1 Ethernet1
Interface: Et1
Scheduling Group Name: G1
Bandwidth: 10.1 / 10.0 (Gbps)
Shape Rate: 75.2 / 75.0 (Gbps)

Member         Bandwidth        Shape Rate
                 (units)          (units)
------       -------------  ------------------
Et1.1           - /  - (-)  50.1 / 50.0 (Gbps)
Et1.2           - /  - (-)  30.1 / 30.0 (Gbps)
```

11.1.7.56 show route counters

Use the **show route [ipv4 | ipv6] counters** to display the IPv4 or IPv6 pack and byte counts.

Command Mode

EXEC

Command Syntax

show route [ipv4 | ipv6]

Parameters

- **ipv4** Displays IPv4 route counters.
- **ipv6** Displays IPv6 route counters.

Example

The following example displays the IPv4 packet and byte count.

```
switch# show route ipv4 counters
Vrf Name        Prefix       Packet Count    Byte Count
-------- ---------------- ------------------ ----------
 default    22.0.0.0/8                    0          0
 default    32.0.0.0/8                    0          0
```

11.1.7.57 show transceiver status interface

The **show transceiver status interface [<intf>]** command displays the most important alarms, faults and interface status.

Example

For 400GBASE-ZR modules, media and host side coherent alarms, host-side pre-FEC BER, defined in the Coherent CMIS and post-FEC BER have been added to the command's output:

```
switch# show transceiver status interface Ethernet14/1
                              Current State        Changes       Last Change
                              -------------        -------       -----------
Port 14
  Transceiver                 400GBASE-ZR                1       0:15:09 ago
  Transceiver SN              200554050
  Presence                    present
  Adapters                    none
  Bad EEPROM checksums                                   0       never
  Resets                                                 0       0:15:14 ago
  Interrupts                                             0       never
  Data path firmware fault    ok                         0       never
  Module firmware fault       ok                         0       never
  Temperature high alarm      ok                         0       never
  Temperature high warn       ok                         0       never
  Temperature low alarm       ok                         0       never
  Temperature low warn        ok                         0       never
  Voltage high alarm          ok                         0       never
  Voltage high warn           ok                         0       never
  Voltage low alarm           ok                         0       never
  Voltage low warn            ok                         0       never
  Module state                ready                      4       0:14:55 ago
  Data path 1 state           activated                  4       0:07:21 ago
  Data path 2 state           activated                  4       0:07:21 ago
  Data path 3 state           activated                  4       0:07:21 ago
  Data path 4 state           activated                  4       0:07:21 ago
  Data path 5 state           activated                  4       0:07:21 ago
  Data path 6 state           activated                  4       0:07:21 ago
  Data path 7 state           activated                  4       0:07:21 ago
  Data path 8 state           activated                  4       0:07:21 ago
  RX LOS                      ok                         2       0:05:54 ago
  TX fault                    ok                         0       never
  RX CDR LOL                  ok                         0       never
  TX power high alarm         ok                         0       never
  TX power high warn          ok                         2       0:07:39 ago
  TX power low alarm          ok                         2       0:07:50 ago
  TX power low warn           ok                         4       0:07:39 ago
  TX bias high alarm          ok                         0       never
```

```
  TX bias high warn              ok                          0        never
  TX bias low alarm              ok                          0        never
  TX bias low warn               ok                          0        never
  RX power high alarm            ok                          0        never
  RX power high warn             ok                          0        never
  RX power low alarm             ok                          0        never
  RX power low warn              ok                          0        never
  TX loss of alignment       ok                          0        never
  TX out of alignment            ok                          0        never
  TX clock monitor unit LOL      ok                          0        never
  TX reference clock LOL         ok                          0        never
  TX deskew LOL                  ok                          0        never
  TX FIFO error                  ok                          0        never
  RX demodulator LOL             ok                          0        never
  RX CD compensation LOL         ok                          0        never
  RX loss of alignment           ok                          0        never
  RX out of alignment            ok                          0        never
  RX deskew LOL                  ok                          0        never
  RX FIFO error                  ok                          0        never
  RX FEC excessive degrade       ok                          0        never
  RX FEC detected degrade        ok                          0        never
  Freq tuning in progress        idle                        0        never
  Freq tuning busy               ok                          0        never
  Freq tuning invalid channel    ok                          0        never
  Freq tuning completed          no                          2        0:07:34 ago
Ethernet14/1
  Operational speed              400Gbps
  Pre-FEC bit error rate         0.00e+00
  Post-FEC errored frames ratio  0.00e+00
  TX LOS
    Host lane 1                  ok                          0        never
    Host lane 2                  ok                          0        never
    Host lane 3                  ok                          0        never
    Host lane 4                  ok                          0        never
    Host lane 5                  ok                          0        never
    Host lane 6                  ok                          0        never
    Host lane 7                  ok                          0        never
    Host lane 8                  ok                          0        never
  TX CDR LOL
    Host lane 1                  ok                          0        never
    Host lane 2                  ok                          0        never
    Host lane 3                  ok                          0        never
    Host lane 4                  ok                          0        never
    Host lane 5                  ok                          0        never
    Host lane 6                  ok                          0        never
    Host lane 7                  ok                          0        never
    Host lane 8                  ok                          0        never
  TX adaptive input EQ fault
    Host lane 1                  ok                          0        never
    Host lane 2                  ok                          0        never
    Host lane 3                  ok                          0        never
    Host lane 4                  ok                          0        never
    Host lane 5                  ok                          0        never
    Host lane 6                  ok                          0        never
    Host lane 7                  ok                          0        never
    Host lane 8                  ok                          0        never
```

Note: In the operational state, module state is ‘Ready’ and datapath state for all 8 lanes is activated.

11.1.7.58 show vxlan counters vni

Use the **show vxlan counters vni** command to display the encapsulation and decapsulation counters at the VNI.

Command Mode

Privledged EXEC

Command Syntax

show vxlan counters vni [encap | decap]

Parameters

- **encap** Displays the encapsulation related counters at the VNI.
- **decap** Displays the decapsulation related counters at the VNI.

Examples

- The following **show vxlan counters vni encap** command displays the encapsulation related commands at the VNI.

```
switch# show vxlan counters vni encap
                                        Encap BUM   Encap Drop
VNI       Encap Bytes  Encap Packets  Packets     Packets
--------- ------------ -------------- ----------- -----------
2824963             0              0           0           0
7023745             0              0           0           0
```

  Encap Packets displays the total count of L2 packets that have been successfully encapsulated by the VNI towards the VTEP.

  Encap Drop Or Exception Packets display the total count of L2 packets that have been encapsulated and dropped for any reason.
- The following **show vxlan counters vni decap** command displays the decapsulation related counters at the VNI.

```
switch# show vxlan counters vni decap
                                                          Decap Drop Or

                            Decap Known   Decap BUM       Exception
VNI       Decap Bytes   Unicast Packets      Packets        Packets
--------  -----------   ---------------   ----------   -------------
2824963             0                 0            0               0
7023745             0                 0            0               0
```

  Decap Known UnicastPkts displays the total count of L2 known unicast packets coming into the VTI hitting the VTEP and getting decapsulated.

  Decap BUM Packets displays the total count of L2 BUM packets coming into the VTI hitting the VTEP and getting decapsulated.

  Decap Drop or Exception Packets displays the count of L2 packets coming into the VTI and getting dropped for any reason.

11.1.7.59 speed

The **speed** command configures the transmission speed and duplex setting for the configuration mode interface. The scope and effect of this command depends on the interface type. Interface types include:

- **40GBASE (QSFP+)**: Default is 4x10G-full. **Speed forced 40gfull** and **Speed auto 40gfull** configure interface as a 40G port.
- **10GBASE-T**: Default is 10G-full. **Speed** command affects interface.

- **10GBASE (SFP+)**: Default is 10G-full. **Speed** command does not affect interface.
- **1000BASE (copper)**: Default is 1G-full. **speed auto 100full** affects interface.
- **1000BASE (fiber)**: Default is 1G-full. **Speed** command does not affect interface.
- **10/100/1000**: Default is auto-negotiation. **Speed** command (10/100/1000 options) affects interface.

The **`speed 40gfull`** and **`auto 40gfull`** commands configure a QSFP+ Ethernet interface as a 40G port. The **`no speed`** and **`no auto 40gfull`** commands configure a QSFP+ Ethernet interface as four 10G ports. These commands must be applied to the /1 port. These commands are hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, these commands restart the forwarding agent, which will result in traffic disruption.

The **`no speed`** and **`default speed`** commands restore the default setting for the configuration mode interface by removing the corresponding **`speed`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

`speed` **MODE**

`no speed`

`default speed`

Parameters

MODE Transmission speed and duplex setting. Options include:

- **speed auto** auto negotiation mode. (For SFP-1G-T, auto-negotiates 1Gbps, this is because no speed is specified, and we are defaulting to advertise 1G).
- **speed auto 40gfull** auto negotiation mode with clause 73 auto negotiation.
- **speed auto 1G full/ speed 1G** auto-negotiated 1Gbps (note that per BASE-T standard, 1G must be negotiated).
- **speed auto 100full** auto-negotiated 100Mbps.
- **speed 100full** non-negotiated and true-forced 100Mbps.

  Note: Interfaces using clause 73 auto negotiation must connect to a device that runs clause 73 auto negotiation.
- **sfp-1000baset auto** auto-negotiation mode (fu).
- **10000full** 10G full duplex.
- **1000full** 1G full duplex.
- **1000half** 1G half duplex.
- **100full** 100M full duplex.
- **100gfull** 100G full duplex.
- **100half** 100M half duplex.
- **10full** 10M full duplex.
- **10half** 10M half duplex.
- **40gfull** 40G full duplex.

On 40GBASE and 100GBASE interfaces, options that change the SFP+ and MXP interfaces (the **auto 40gfull**, the ***40gfull***, and the **no speed** options) may restart the forwarding agent on some switch platforms, disrupting traffic on all ports for more than a minute.

Guidelines

Note: The **SFP-1G-T** transceivers advertise one speed at a time only. Hence, the desired speed and negotiation must be configured explicitly using the **speed auto, speed auto 1G full/ speed 1G, speed auto 100full,** and **speed 100full** commands.

Examples

- This command configures a 40GBASE interface as a 40G port.

```
switch(config)# interface ethernet 49/1
switch(config-if-Et49/1)# speed 40gfull
switch(config-if-Et49/1)# show interface ethernet 49/1 - 49/4 status
Port      Name          Status       Vlan        Duplex  Speed Type
Et49/1                  connected    in Po999      full    40G 40GBASE-CR4
Et49/2                  errdisabled  inactive    unconf unconf 40GBASE-CR4
Et49/3                  errdisabled  inactive    unconf unconf 40GBASE-CR4
Et49/4                  errdisabled  inactive    unconf unconf 40GBASE-CR4
switch(config-if-Et49/1)#
```

- This command configures a 40GBASE interface as four 10G ports (default configuration).

```
switch(config-if-Et49/1)# no speed
switch(config-if-Et49/1)# show interface ethernet 49/1 - 49/4 status
Port      Name          Status       Vlan        Duplex  Speed Type
Et49/1                  connected    routed        full    10G 40GBASE-SR4
Et49/2                  connected    routed        full    10G 40GBASE-SR4
Et49/3                  connected    routed        full    10G 40GBASE-SR4
Et49/4                  notconnect   inactive      full    10G 40GBASE-SR4
switch(config-if-Et49/1)#
```

11.1.7.60 system

The **system** command allows the user to configure the system-wide TCAM profiles such as default, mirroring-acl, mpls-evpn, pbr-match-nexthop-group, qos, tap-aggregation-default, tap-aggregation-extended, tc-counters, test, and vxlan-routing.

The **exit** command returns the switch to global configuration mode.

Command Mode

Hardware TCAM

Command Syntax

system profile *name*

Parameter

name TCAM profile name.

Reference

hardware tcam

Example

This commands allow the switch to configure the TCAM profile qos.

```
switch(config-hw-tcam)# system profile qos
```

11.1.7.61 threshold

The **threshold** command configures the link monitoring threshold value that is specified for a link error.

The **no threshold** and the **default threshold** commands remove the threshold value specified for the chosen link error.

Command Mode

Link-error Configuration

Command Syntax

[**fcs** | **symbol**] **`threshold`** ***threshold_value***

no [**fcs** | **symbol**] **`threshold`** ***threshold_value***

default [**fcs** | **symbol**] **`threshold`** ***threshold_value***

Parameters

- **fcs** Inbound packets with Frame check sequence (FCS) error.
- **symbol** Inbound packets with symbol error.
- ***threshold_value*** Specifies the threshold value in number of errors. The value ranges from ***1*** to ***100***.

Related Commands

- action
- period

Example

These commands set the threshold value of ***20*** for the ***profile profile1*** in the ***link-error*** configuration mode.

```
switch(config)# monitor ethernet oam
switch(config-eoam)# profile profile1
switch(config-eoam-profile-profile1)# link-error
switch(config-eoam-profile-profile1-link-error)# symbol threshold 20
```

11.1.7.62 transceiver channel

The **`transceiver channel`** command displays transceiver wavelength/frequency by channel number. The channel numbering depends on the selected grid-spacing mode. The default grid-spacing mode is 50GHz-spacing.

- If the startup configuration does not specify the channel number for the interface, the transceiver will automatically tune to the default channel (i.e. ***channel-39*** of 50GHz-spacing grid) when it is inserted.
- If the configured wavelength/frequency is not supported by the transceiver, the transceiver is tuned to the default channel (i.e. ***channel-39*** of 50GHz-spacing grid).

The interface is shutdown before the channel number is configured.

Command Mode

Global Configuration

Command Syntax

`transceiver channel` **CHANNEL_NUMBER** **`grid-spacing`** ***SPACING_GRID***

`no transceiver channel` **CHANNEL_NUMBER** **`grid-spacing`** ***SPACING_GRID***

`default transceiver channel` **CHANNEL_NUMBER** **`grid-spacing`** ***SPACING_GRID***

Parameters

- **CHANNEL-NUMBER** The default channel is ***39*** (50GHz-spacing grid) which corresponds to a frequency of 193,100 GHz and a wavelength of 1552.52 nm.
- **GRID_SPACING** Grid-spacing mode (optional) depends on the selected grid-spacing mode. The default grid-spacing mode is 50GHz-spacing. For example, channel ***39*** of 50GHz-spacing grid is equivalent to channel ***20*** of 100GHz-spacing grid, which corresponds to a frequency of 193,100 GHz and a wavelength of 1552.52 nm.
  - ***SPACING_GRID*** default grid-spacing mode in GHz.

Related Commands

- show interfaces transceiver channels
- show interfaces transceiver hardware

Example

This command tunes the transceiver on slot number ***4*** to slot ***1*** through ***3*** of 50GHz-spacing grid.

```
switch(config-as)# interface ethernet 4/1/3
switch(config-if-Et4/1/3)# transceiver channel 1 grid-spacing 50
switch(config-if-Et4/1/3)#
```

11.1.7.63 transceiver qsfp default-mode

The **`transceiver qsfp default-mode`** command specifies the transmission mode of all QSFP transceiver modules that are not explicitly configured.

Each QSFP+ module Ethernet interface is configurable as a single 40G port or as four 10G ports. The switch displays four ports for each interface. Each ports status depends on the interface configuration:

- The ***/1*** port is active (**connected** or not connected), regardless of the interface configuration.
- The ***/2***, ***/3***, and ***/4*** ports are **error-disabled** when the interface is configured as a single 40G port.
- All ports are active (**connected** or not connected), when the interface is configured as four 10G ports.

QSFP modules that are not configured through a **`speed`** command are operated as four 10G ports.

The **`no transceiver qsfp default-mode`** and **`default transceiver qsfp default-mode`** commands restore the default-mode transceiver setting from 40G to 4x10G.

Command Mode

Global Configuration

Command Syntax

`transceiver qsfp default-mode 4x10G`

`no transceiver qsfp default-mode`

`default transceiver qsfp default-mode`



Note: QSFP100 ports with external PHY always have 40G as the default QSFP mode.

Guidelines

The **`transceiver qsfp default-mode 4x10g`** statement is always in ***running-config*** and cannot be modified or removed in the current release.

Example

- When default QSFP mode is configured as 4x10G, the **`show running-config`** contains **`transceiver qsfp default-mode 4x10G`**. **`show interfaces hardware`** shows 10G as the default speed.

```
switch(config)# transceiver qsfp default-mode 4x10G
switch(config)# show running-config | grep 4x10G
transceiver qsfp default-mode 4x10G
switch(config)# show interfaces ethernet 35/1 hardware
*  = Requires speed group setting change
Ethernet35/1
  Model:          DCS-7280CR3-32P4
  Type:           40GBASE-CR4
  Speed/Duplex:   10G/full(default),40G/full,auto
  Flowcontrol:    rx-(off,on,desired),tx-(off)
```

- When QSFP mode configuration is reverted to the default, **`show running-config`**does not contain **`transceiver qsfp default-mode 4x10G`**. **`show interfaces hardware`** shows 40G as the default speed.

```
switch(config)# no transceiver qsfp default-mode
switch(config)# show running-config | grep 4x10G
switch(config)# show interfaces ethernet 35/1 hardware
*  = Requires speed group setting change
Ethernet35/1
  Model:          DCS-7280CR3-32P4
  Type:           40GBASE-CR4
  Speed/Duplex:   10G/full,40G/full(default),auto
  Flowcontrol:    rx-(off,on,desired),tx-(off)
```

11.2 Port Channels and LACP

This chapter describes channel groups, port channels, port channel interfaces, and the Link Aggregation Control Protocol (LACP). This chapter contains the following sections:

- Port Channel Introduction
- Port Channel Conceptual Overview
- Port Channel Configuration Procedures
- Load Balancing Hash Algorithms
- Port Channel and LACP Configuration Commands

11.2.1 Port Channel Introduction

Arista's switching platforms support industry-standard link aggregation protocols. Arista switches optimize traffic throughput by using MAC addressing, IP addressing, and services fields to effectively load share traffic across aggregated links. Managers can configure multiple ports into a logical port channel, either statically or dynamically through the IEEE Link Aggregation Control Protocol (LACP). Various negotiation modes are supported to accommodate different configurations and peripheral requirements, including LACP fallback to support devices that need simple network connectivity to retrieve images or configurations prior to engaging port channel aggregation modes.

Arista's Multi-chassis Link Aggregation protocol (MLAG) supports LAGs across paired Arista switches to provide both link aggregation and active/active redundancy.

11.2.2 Port Channel Conceptual Overview

11.2.2.1 Channel Groups and Port Channels

A port channel is a communication link between two switches supported by matching channel group interfaces on each switch. A port channel is also referred to as a Link Aggregation Group (LAG). Port channels combine the bandwidth of multiple Ethernet ports into a single logical link.

A channel group is a collection of Ethernet interfaces on a single switch. A port channel interface is a virtual interface that serves a corresponding channel group and connects to a compatible interface on another switch to form a port channel. Port channel interfaces can be configured and used in a manner similar to Ethernet interfaces. Port channel interfaces are configurable as Layer 2 interfaces, Layer 3 (routable) interfaces, and VLAN members. Most Ethernet interface configuration options are also available to port channel interfaces.

11.2.2.2 Port Channel Subinterfaces

Port channel subinterfaces divide a single port channel interface into multiple logical L3 interfaces based on the 802.1q tag (VLAN ID) of incoming traffic. Subinterfaces are commonly used in the L2/L3 boundary device,

but they can also be used to isolate traffic with 802.1q tags between L3 peers by assigning each subinterface to a different VRF.

For further details about subinterfaces, see Subinterfaces.

11.2.2.3 Link Aggregation Control Protocol (LACP)

The Link Aggregation Control Protocol (LACP), described by IEEE 802.3ad, defines a method for two switches to automatically establish and maintain link aggregation groups (LAGs, also called channel groups or port channels). Using LACP, a switch can configure LACP-compatible ports into a dynamic LAG. The ports try to complete LACP negotiation automatically with the linked ports (also configured as a dynamic LAG) on the partner switch. The maximum number of ports per LAG varies by platform; numbers for each platform in the latest EOS release are available here: https://www.arista.com/en/support/product-documentation/supported-features.

Static LAGs

In static mode (with the channel-group mode configured as **on** on the member interfaces), the switch aggregates links without an awareness of LAGs on the partner switch and without LACP negotiation. The member ports do not send LACP packets or process inbound LACP packets on static LAGs. Packets may drop when static LAG configurations differ between switches.

Dynamic LAGs

Dynamic LAGs are aware of their partners' port-channel states. Interfaces configured as dynamic LAGs are designated as **active** or **passive**.

- **Active interfaces** send LACP Protocol Data Units (LACP PDUs) at a rate of one per second when forming a channel with an interface on the peer switch. An aggregate forms if the peer runs LACP in active or passive mode.
- **Passive interfaces** only send LACP PDUs in response to PDUs received from the partner. The partner switch must be in active mode and initiates negotiation by sending a LACP packet. The passive mode switch receives and responds to the packet to form a LAG.

An active interface can form port channels with passive or active partner interfaces, but port channels are not formed when the interface on each switch is passive.

Table 63: LACP Mode Combinations summarizes the effect of different LACP mode combinations:

Table 63: LACP Mode Combinations

Switch 1	Switch 2	Comments
active	active	Links aggregate when LACP negotiation is successful.
active	passive	Links aggregate when LACP negotiation is successful.
passive	passive	Links do not aggregate because LACP negotiation is not initiated.
on (static)	on (static)	Links aggregate without LACP.
on (static)	active or passive	Links aggregate on the static switch without LACP; links do not aggregate on the other switch, and no port-channel connection is established with the partner.

During synchronization, interfaces in dynamic LAGs transmit one LACP PDU per second. After synchronization is complete, interfaces exchange one PDU every thirty seconds, facilitated by a default timeout of 30 seconds and a failure tolerance of three. Under these parameters, when the switch does not receive a LACP PDU for an interface during a ninety-second period, it records the partner interface as **failed** and removes the interface from the port channel.

Fallback Mode

An active interface that is not in fallback mode does not form a LAG until it receives PDUs from, and negotiates with its peer. Fallback mode allows an active LACP interface to maintain a LAG without receiving PDUs from its peer. The fallback timer specifies the period the LAG waits to receive a peer PDU. Upon timer expiry, the port channel reverts to its configured fallback mode if one is configured.

Static fallback: the port channel maintains one active port while in fallback mode; all its other member ports are in standby mode until a LACP PDU is received by the port channel. All member ports send (and can receive) LACP PDUs, but only the active port sends or receives data.

Individual fallback: all member ports act as individual switch ports while in fallback mode. Individual port configuration (rather than port channel configuration) is active while the port channel is in fallback mode, with the exception of ACLs. This includes VLAN membership. All member ports send and receive data, and continue to send LACP PDUs. As soon as a LACP PDU is received by a member of the port channel, all ports revert to normal port-channel operation.

The switch uses a link aggregation hash algorithm to determine the forwarding path within a link aggregation group. The IP and MAC header fields can be selected as components of the hash algorithm.

11.2.3 Port Channel Configuration Procedures

These sections describe channel group and port channel configuration procedures:

- Configuring a Channel Group
- Configuring a Port Channel Interface
- Maximum Port Channel ID Increase
- Configuring Port Channel Subinterfaces
- Configuring LACP
- Displaying Port Channel Information

11.2.3.1 Configuring a Channel Group

Creating a Channel Group

The channel-group command assigns the configuration-mode Ethernet interfaces to a channel group, creates the channel group if it does not already exist, and specifies LACP attributes for the channel.

Channel groups are associated with a port channel interface immediately upon their creation. A command that creates a new channel group also creates a port channel with a matching ID. The port channel is configured in port-channel configuration mode. Configuration changes to a port channel interface propagate to all Ethernet interfaces in the corresponding channel group.

LACP is enabled on the member interfaces by setting the channel-group mode to **active** or **passive**. Setting the mode to **on** disabled LACP on the member interfaces and creates a static channel group.

Example:

These commands assign Ethernet interfaces ***1*** and ***2*** to channel group ***10*** (creating the channel group if it does not already exist), enable LACP on those interfaces, and place the channel group in a negotiating state.

```
switch(config)# interface ethernet 1-2
switch(config-if-Et1-2)# channel-group 10 mode active
```

```
switch(config-if-Et1-2)#
```

Adding an Interface to a Channel Group

The channel-group command is also used to add the configuration mode interface to an existing channel group. When adding channels to a previously created channel group, the channel-group mode for the new channel must match the mode for the existing group.

Example

These commands add Ethernet interfaces ***7*** through ***10*** to previously created channel group ***10***, using the channel-group mode (**active**) under which it was created.

```
switch(config)# interface ethernet 7-10
switch(config-if-Et7-10)# channel-group 10 mode active
switch(config-if-Et7-10)#
```

Removing an Interface from a Channel Group

The **`no channel-group`** command removes the configuration mode interface from the specified channel group. Deleting all members of a channel group does not remove the associated port channel interface from ***running-config***.

Example

These commands removes ***interface ethernet 8*** from previously created channel group ***10***.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# no channel-group
switch(config-if-Et8)#
```

Configuring a Port-Channel as Mixed-Speed

By default, only configured members of the same speed become active. The port-channel speed mixed command configures a port channel with the ability to have active members of multiple speeds.

Note: Available on the 7020, 7280, 7500, and 7800 platforms. Minimum links is not available on mixed-speed port channels.

Example

```
switch(config)# interface port-channel 1
switch(config-if-Po1)# port-channel speed mixed
```

Configuring Minimum Links

Note: Minimum links is not available on Mixed-Speed Port-Channels. If a minimum requirement is desired for a Mixed-Speed Port-Channel, consider Minimum Speed instead. On Port-Channels that are not mixed-speed, if both Minimum Links and Minimum Speed are configured, then Minimum Speed will take precedence.

Configuring Minimum Speed

The port-channel speed minimum command specifies the cumulative minimum speed of all active members in order for a port channel to become active. If there is less than the specified by this command, the port channel interface does not become active.

Note: If both minimum speed and minimum links are configured, minimum speed will take precedence.

Example

These command sets 100 Gbps as the minimum speed needed for ***port channel 1*** to become active.

```
switch(config)# interface port-channel 1
switch(config-if-Po1)# port-channel speed minimum 100 gbps
```

Deleting a Channel Group

A channel group is deleted by removing all Ethernet interfaces from the channel group. A channel group's LACP mode can be changed only by deleting the channel group and then creating an equivalent group with a different LACP mode. Deleting a channel group by removing all Ethernet interfaces from the group preserves the port channel interface and its configuration settings.

View ***running-config*** to verify the deletion of all Ethernet interfaces from a channel group.

11.2.3.2 Configuring a Port Channel Interface

Creating a Port Channel Interface

The switch provides two methods for creating port channel interfaces:

- creating a channel group simultaneously creates an associated port channel.
- the **`interface port-channel`** command creates a port channel without assigning Ethernet channels to the new interface.

The interface port-channel command places the switch in ***interface-port channel*** configuration mode.

Example

This command creates ***interface port-channel 8*** and places the switch in ***port channel interface*** configuration mode.

```
switch(config)# interface port-channel 8
switch(config-if-Po8)#
```

Deleting a Port Channel Interface

The **`no interface port-channel`** command deletes the configuration mode port channel interface and removes the channel group assignment for each Ethernet interface assigned to the group associated with the port channel interface. Removing all Ethernet interfaces from a channel group does not remove the associated port channel interface from ***running-config***.

11.2.3.3 Maximum Port Channel ID Increase

Previously, the maximum valid port channel ID was equal to the maximum number of port channels configurable on the system, ***2000***, and this feature increases the maximum ID to ***999,999*** while maintaining the same limit of ***2000*** port channels on the system.

11.2.3.3.1 Configuration

This feature does not involve any specific configuration procedure, but it does include visible changes to port channel configuration commands. In the following examples, suppose port channels ***1-2000*** have already been configured, so creating Port-Channel ***2001*** would exceed the configuration limit.

```
switch(config)# interface create port-channel 2001
```

```
Port channel config limit 2000 reached. No interfaces were created.
```

```
switch(config-ifEtX)# channel-group 2001 mode
Port channel config limit 2000 reached. No interfaces were created.
```

11.2.3.3.2 Show Commands

Changes to existing show commands simply involve displaying when a port channel is inactive. In the following examples, suppose port channels ***2001-4001*** are configured, Port-Channel ***4001*** is inactive, and Ethernet1 is a member of Port-Channel ***4001***.

```
switch(config)# show lacp 1-$ aggregates
Port channel 4001 is inactive. The number of configured port channels exceeds the config limit
 2000.
Port-Channels1-2000,4002-999999 not configured as LAG
Port Channel Port-Channel2001:
Aggregate ID: [(8000,00-1c-73-04-36-d7,0001,0000,0000),(8000,00-1c-73-09-a0-f3,0001,0000,0000)]
  Bundled Ports: Ethernet43 Ethernet44 Ethernet45 Ethernet46
Port Channel Port-Channel2002:
Aggregate ID: [(8000,00-1c-73-01-02-1e,0002,0000,0000),(8000,00-1c-73-04-36-d7,0002,0000,0000)]
  Bundled Ports: Ethernet47 Ethernet48
Port Channel Port-Channel2003:
Aggregate ID: [(8000,00-1c-73-04-36-d7,0003,0000,0000),(8000,00-1c-73-0c-02-7d,0001,0000,0000)]
  Bundled Ports: Ethernet3 Ethernet4
Port Channel Port-Channel2004:
Aggregate ID: [(0001,00-22-b0-57-23-be,0031,0000,0000),(8000,00-1c-73-04-36-d7,0004,0000,0000)]
  Bundled Ports: Ethernet42
Port Channel Port-Channel2005:
Aggregate ID: [(0001,00-22-b0-5a-0c-51,0033,0000,0000),(8000,00-1c-73-04-36-d7,0005,0000,0000)]
  Bundled Ports: Ethernet41



switch(config)# show lacp 1-$ counters
Port channel 4001 is inactive. The number of configured port channels exceeds the config limit
 2000.
Port-Channels1-2000,4002-999999 not configured as LAG


switch(config)# show lacp 1-$ internal
Port channel 4001 is inactive. The number of configured port channels exceeds the config limit
 2000.
Port-Channels1-2000,4002-999999 not configured as LAG
LACP System-identifier: 8000,00-1c-73-04-36-d7
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
            |Partner                              Actor
Port Status  | Sys-id                Port#  State    OperKey  PortPriority
-------------------------------------------------------------------------
Port Channel Port-Channel2001:
Et43 Bundled | 8000,00-1c-73-09-a0-f3    43  ALGs+CD   0x0001         32768
Et44 Bundled | 8000,00-1c-73-09-a0-f3    44  ALGs+CD   0x0001         32768
Et45 Bundled | 8000,00-1c-73-09-a0-f3    45  ALGs+CD   0x0001         32768
Et46 Bundled | 8000,00-1c-73-09-a0-f3    46  ALGs+CD   0x0001         32768

switch(config)# show lacp 1-$ peer
Port channel 4001 is inactive. The number of configured port channels exceeds the config limit
 2000.
Port-Channels1-2000,4002-999999 not configured as LAG
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
              |                          Partner
Port   Status  | Sys-id                  Port#   State     OperKey  PortPri
-------------------------------------------------------------------------
Port Channel Port-Channel2001:
Et1    Bundled | 8000,00-1c-73-00-13-19      1   ALGs+CD    0x0001    32768
Et2    Bundled | 8000,00-1c-73-00-13-19      2   ALGs+CD    0x0001    32768
Port Channel Port-Channel2002:
Et23   Bundled | 8000,00-1c-73-04-36-d7     47   ALGs+CD    0x0002    32768
Et24   Bundled | 8000,00-1c-73-04-36-d7     48   ALGs+CD    0x0002    32768
```

```
Port Channel Port-Channel2004*:
Et3    Bundled | 8000,00-1c-73-0b-a8-0e     45   ALGs+CD    0x0001    32768
Et4    Bundled | 8000,00-1c-73-0b-a8-0e     46   ALGs+CD    0x0001    32768
Port Channel Port-Channel2005*:
Et19   Bundled | 8000,00-1c-73-0c-30-09     49   ALGs+CD    0x0005    32768
Et20   Bundled | 8000,00-1c-73-0c-30-09     50   ALGs+CD    0x0005    32768
Port Channel Port-Channel2006*:
Et6    Bundled | 8000,00-1c-73-01-07-b9     49   ALGs+CD    0x0001    32768
Port Channel Port-Channel2007*:
Et5    Bundled | 8000,00-1c-73-0f-6b-22     51   ALGs+CD    0x0001    32768
Port Channel Port-Channel2008*:
Et10   Bundled | 8000,00-1c-73-10-40-fa     51   ALGs+CD    0x0001    32768

* - Only local interfaces for MLAGs are displayed. Connect to the peer to
    see the state for peer interfaces.

switch(config)# show lacp interface Ethernet1 [(internal|neighbor|peer)]
Interface Ethernet1 is a member of an inactive LACP port channel. The number of configured port
 channels exceeds the config limit 2000.

switch(config)# show port-channel 1-$
Port Channel Port-Channel2001:
  No Active Ports
...
Port Channel Port-Channel4000:
  No Active Ports
Port Channel Port-Channel4001:
  Inactive, The number of configured port channels exceeds the config limit 2000.

switch(config)# show port-channel (dense|summary)

                  Flags
-------------------------- ----------------------------- -------------------------
   a - LACP Active            p - LACP Passive           * - static fallback
   F - Fallback enabled       f - Fallback configured    ^ - individual fallback
   U - In Use                 D - Down
   + - In-Sync                - - Out-of-Sync            i - incompatible with agg
   P - bundled in Po          s - suspended              G - Aggregable
   I - Individual             S - ShortTimeout           w - wait for agg
   E - Inactive. The number of configured port channels exceeds the config limit

Number of channels in use: ...
Number of aggregators: ...

   Port-Channel       Protocol    Ports
------------------ -------------- --------------------
   Po2001(U)          LACP(a)     Et47(PG+) Et48(PG+)
   Po2002(U)          LACP(a)     Et39(PG+) Et40(PG+)
   Po4001(E)          Static      Et7(P)
```

11.2.3.3.3 Limitations

- The number of configured port channels can exceed the configurable limit if two configuration sessions simultaneously create two different port channels. In this scenario, port channels that exceed the limit are inactive. This is uncommon and does not impact traffic in any way. If an inactive port channel exists and an active port channel is deleted, then the inactive port channel is activated.
- Only port channels with ID from ***1 to 2000*** are configured as MLAG port channels.

11.2.3.4 Configuring Port Channel Subinterfaces

When configuring subinterfaces on a port channel interface (the virtual interface associated with a port channel), the following restrictions apply:

An L3 interface with subinterfaces configured on it should not be made a member of a port channel.

- An interface that is a member of a port channel should not have subinterfaces configured on it.
- A subinterface cannot be made a member of a port channel.

Port channel subinterfaces are otherwise configured similarly to Ethernet subinterfaces. For additional information, see Subinterfaces.

11.2.3.5 Configuring LACP

Configuring the Channel-group Mode

The channel-group mode is configured when a channel group is created using the channel-group command. A channel group's mode cannot be modified without deleting the entire channel group, but it can be modified without deleting the port channel interface associated with the channel group. The mode setting defines whether the port channel is static or dynamic, and whether a dynamic port channel is active or passive.

Examples

- These commands create a dynamic channel group and place it in LACP **active** mode.

```
switch(config)# interface ethernet 1-2
switch(config-if-Et1-2)# channel-group 10 mode active
switch(config-if-Et1-2)#
```

- These commands create a static channel group.

```
switch(config)# interface ethernet 4-5
switch(config-if-Et4-5)# channel-group 11 mode on
switch(config-if-Et4-5)#
```

Configuring the System Priority

Each switch is assigned a globally unique system identifier by concatenating the system priority (16 bits) to the MAC address of one of its physical ports (48 bits). The system identifier is used by peer devices when forming an aggregation to verify that all links are from the same switch. The system identifier is also used when dynamically changing aggregation capabilities in response to LACP information; the system with the numerically lower system identifier is permitted to dynamically change advertised aggregation capabilities.

The lacp system-priority command configures the switch's LACP system priority.

Example

This command assigns the system priority of ***8192*** to the switch.

```
switch(config)# lacp system-priority 8192
switch(config)#
```

Configuring Port Priority

LACP port priority determines the port that is active in a LAG in fallback mode. Numerically lower values have higher priority. Port priority is supported on port channels that are enabled with LACP physical interfaces.

The lacp port-priority command sets the aggregating port priority for the configuration mode interface.

Example

This command assigns the port priority of ***4096*** to ***Ethernet interface 1***.

```
switch(config-if-Et1)# lacp port-priority 4096
switch(config-if-Et1)#
```

Configuring the LACP Packet Reception Rate

The lacp timer command sets the reception rate of LACP packets on the local device for the interface being configured. This command supports the following reception rates:

- **normal:** LACP packets are received at the following rates:
  - 30 seconds for synchronized interfaces.
  - One second for interfaces that are being synchronized.
- **fast:** LACP packets are received every second.
-

Example

This command sets the LACP reception rate to one second on the ***Ethernet interface 4***.

```
switch(config-if-Et4)# lacp timer fast
switch(config-if-Et4)#
```

Configuring LACP Fallback

Fallback mode (static or individual) is configured on a port channel interface with the port-channel lacp fallback command. The fallback timeout interval is configured with the port-channel lacp fallback timeout command. Fallback timeout settings persist in ***running-config*** without taking effect for interfaces that are not configured into fallback mode. The default fallback timeout period is ***90*** seconds.

Examples

- These commands enable LACP static fallback mode, then configure an LACP fallback timeout of ***100*** seconds on ***port channel interface 13***. If LACP negotiation fails, only the member port with the lowest LACP priority will remain active until an LACP PDU is received by one of the member ports.

```
switch(config)# interface port-channel 13
switch(config-if-Po13)# port-channel lacp fallback static
switch(config-if-Po13)# port-channel lacp fallback timeout 100
switch(config-if-Po13)# show active
interface Port-Channel13
   port-channel lacp fallback static
   port-channel lacp fallback timeout 100
switch(config-if-Po13)#
```

- These commands enable LACP individual fallback mode, then configure an LACP fallback timeout of ***50*** seconds on ***port channel interface 17***. If LACP negotiation fails, all member ports will act as individual switch ports, using port-specific configuration, until a LACP PDU is received by one of the member ports.

```
switch(config)# interface port-channel 17
switch(config-if-Po17)# port-channel lacp fallback individual
switch(config-if-Po17)# port-channel lacp fallback timeout 50
switch(config-if-Po17)# show active
interface Port-Channel17
   port-channel lacp fallback individual
   port-channel lacp fallback timeout 50
switch(config-if-Po17)#
```

Configuring Minimum Links

The port-channel min-links command specifies the minimum number of interfaces that the configuration mode LAG requires to be active. If there are fewer ports than specified by this command, the port channel interface does not become active.

Note: In static LAGs, the min-links value must be met for the LAG to be active. The LAG will not become active until it has at least the min-links number of functioning links in the channel group. If failed links cause the number to drop below the minimum, the LAG will go down and administrator action will be required to bring it back up. In dynamic LAGs, the LACP protocol must determine that at least min-links physical ports are aggregable (they are physically compatible and have the same keys both remotely and locally) before it begins negotiating to make any ports active members of the port-channel. However once negotiation begins, an error on the partner's side or an error in programming of member interfaces can cause the LAG to become active with fewer than the minimum number of links. EOS evaluates min-links after min-links-review-timeout (linearly proportional to configured min-links) when LACP protocol collecting and/or distributing state changes. If the number of active member interfaces in a port-channel is less than configured min-links, it brings the corresponding port-channel Link Down and syslogs **LAG-4-MINLINK_INTF_INSUFFICIENT** message. If additional interfaces get programmed as collecting and distributing, EOS re-evaluates min-links on the port-channel. If sufficient number of interfaces are available to be a part of port-channel, then all interfaces of the corresponding port-channel are re-enabled for LACP negotiation and the port-channel becomes Link Up. **LAG-4-MINLINK_INTF_NORMAL** is syslogged after min-links-review-timeout if the min-links condition is satisfied; otherwise **LAG-4-MINLINK_INTF_INSUFFICIENT** is syslogged and the port-channel goes Link Down. If an interface remains in collecting state but not in distributing state for min-links-review-timeout, it is moved out of collecting state. It is periodically re-enabled after min-links-retry-timeout (which is 360 seconds) till it progresses to collecting and distributing. Meanwhile, if a port-channel becomes Link Up because sufficient number of interfaces progressed to collecting and distributing states, then this interface is enabled for LACP negotiation.

Example

This command sets four as the minimum number of ports required for ***port channel 5*** to become active.

```
switch(config-if-Po5)# port-channel min-links 4
switch(config-if-Po5)#
```

Configuring Minimum Links Review Interval

The port-channel min-links review interval command enables or disables timer based min-links review feature for all port-channels. The timer based min-links feature is enabled when all of the following conditions are true. It is disabled otherwise:

- The min-links configured is greater than ***1***.
- LACP fallback is disabled.
- The number of interfaces configured in the port-channel is more than min-links.
- The number of active member interfaces in the port-channel is less than min-links.
- The default timer values are:
  - min-links-review-timeout = min-links-timeout-base + f (configured min-links)
  - min-links-timeout-base = ***180*** seconds
  - min-links-retry-timeout = ***360*** seconds

11.2.3.6 Displaying Port Channel Information

Port channel information is accessed using some of the **show** commands listed under Interface Display Commands. Ensure that while using the show interfaces counters rates command to view the rate information of a port channel, rate values for the individual member ports are less inaccurate than rate values of the port channel.

Both the port channel rate and the individual port rates are calculated approximations; the rate value of a port channel might vary from the total of the rates for the member ports. The discrepancy is likely to be larger for port channels with fewer ports, and will be most obvious in single-port port channels.

11.2.4 Load Balancing Hash Algorithms

The switch balances packet load across multiple links in a port channel by calculating a hash value based on packet header fields. The hash value determines the active member link through which the packet is transmitted. This method, in addition to balancing the load in the LAG, ensures that all packets in a data stream follow the same network path.

In network topologies that include MLAGs or Multiple Paths with Equal Cost (ECMP), programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links. This uneven distribution is avoided by performing different hash calculations on each switch routing the paths.

The port-channel load-balance command specifies the seed for hashing algorithms that balance the load across ports comprising a port channel. Available seed values vary by switch platform.

Example

This command configures the hash seed of ***10*** on 7150 Series (FM6000 platform) switches.

```
switch(config)# port-channel load-balance fm6000 10
switch(config)#
```

Hashing algorithm inputs varies by switch platform. These sections describe hashing algorithm inputs for each platform.

- Load Balance Hash Algorithms on 7048 and 7500 Series Switches
- Load Balance Hash Algorithms on 7500E Series Switches
- Load Balance Hash Algorithms on 7050 Series Switches
- Load Balance Hash Algorithms on 7150 Series Switches

11.2.4.1 Load Balance Hash Algorithms on 7048 and 7500 Series Switches

One command configures the load balance hash algorithm on 7048 and 7500 Series switches:

- port-channel load-balance petraA fields ip: controls the hash algorithm for IP packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets), and the entire MAC address header.

The hash algorithm for non-IP packets is not configurable and always includes the entire MAC header.

Example

These commands configure the load balance algorithm for IP packets by using the entire MAC header.

```
switch(config)# port-channel load-balance petraA fields ip mac-header
```

```
switch(config)#
```

11.2.4.2 Load Balance Hash Algorithms on 7500E Series Switches

One command configures the load balance hash algorithm on 7500E Series switches:

port-channel load-balance arad fields ip: controls the hash algorithm for IP packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets), and the entire MAC address header.

The hash algorithm for non-IP packets is not configurable and always includes the entire MAC header.

Example

These commands configure the load balance algorithm for IP packets by using the entire MAC header.

```
switch(config)# port-channel load-balance arad fields ip mac-header
switch(config)#
```

11.2.4.2.1 Dynamic and Symmetric LAG Hashing

Dynamic LAG hashing enables high link utilization and highly even distribution among LAG members by employing a randomized hashing algorithm. Symmetric LAG hashing allows the two flows of a bidirectional communication link, even when the two flows enter the switch on different ingress ports, to be hashed to the same member of a LAG on egress.

Dynamic and symmetric LAG hashing policies are enabled via named port-channel load-balancing profiles. LAG load-balancing policies can be provisioned on per line-card basis using these profiles. Load-balancing profiles can be used to provision all LAG load-balance attributes, including hash polynomials, hash seeds, and hash fields.

When no specific LAG hashing profile is assigned to a line card, then a global LAG hashing profile can be defined and applied to all the line cards with no LAG hashing defined on them.

Note, if no profile is selected as global profile then the default profile takes the precedence and set as a global profile. The default profile is reserved and if it is set as a global profile it cannot be deleted, if the profile is deleted then the following warning message is displayed.

Note: When a global profile is already set and if some other profile is tried to configured as a default profile the following warning message is displayed "! A global load balancing profile myProfile is currently active. This setting will not take effect."

Examples

- These commands configure a load balance profile for symmetric hashing.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance arad profile
switch(config-sand-load-balance-profile-symmetric-profile-1)# hash
 symmetric
switch(config-sand-load-balance-profile-symmetric-profile-1)# show
 active
load-balance policies
   load-balance arad profile symmetric-profile-1
```

```
      hash symmetric
```

- These commands configure a load balance profile for dynamic hashing.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance arad profile
switch(config-sand-load-balance-profile-dynamic-hash-
profile-1)# distribution
clock
switch(config-sand-load-balance-profile-dynamic-hash-
profile-1)# show active
load-balance policies
   load-balance arad profile dynamic-hash-profile-1
      distribution clock
```

- This command assigns a named load-balancing profile to a linecard.

```
switch(config)# port-channel load-balance module 3-7 sand profile
 Linecard5
switch(config)#
```

- This command unassigns a named load-balancing profile to a linecard.

```
switch(config)# no port-channel load-balance module 3-7 sand profile
 Linecard5
switch(config)#
```

- This command configures a global profile on all line cards on which LAG hashing is not defined.

```
switch(config)# port-channel load-balance sand profile myGlobalProfi
le
```

- These commands designates a default profile as a global profile, if no other profile is set as a global profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance sand profile
 default
```

- These commands configure a hash seed in a profile and assigns it as a global profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance sand profile
 myGlobalProfile
switch(config-sand-load-balance-profile-myGlobalProfile)# hash seed
 20
switch(config)# port-channel load-balance sand profile myGlobalProfi
le
```

- This command assigns a named load-balancing profile to a linecard.

```
switch(config)# port-channel load-balance module 3-7 sand profile
 Linecard5
switch(config)#
```

- This command unassigns a named load-balancing profile to a linecard.

```
switch(config)# no port-channel load-balance module 3-7 sand profile
 Linecard5
```

```
switch(config)#
```

11.2.4.3 Load Balance Hash Algorithms on 7050 Series Switches

Three commands configure the load balance hash algorithm on 7050 Series switches:

- port-channel load-balance trident fields ip controls the hash algorithm for IP packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets), and fields specified by the **`port-channel load-balance trident fields mac`** command.
- port-channel load-balance trident fields ipv6 controls the hash algorithm for IPv6 packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets), and fields specified by the **`port-channel load-balance trident fields mac`** command.
- port-channel load-balance trident fields mac controls the hash algorithm for non-IP packets b specifying the algorithm's use of MAC header fields. Fields that the command can specify include the MAC source address, MAC destination address, and Ethernet type fields.

Example

These commands configure the switch's port channel load balance for non IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
switch(config)# port-channel load-balance trident fields mac dst-mac
 eth-type
switch(config)#
```

11.2.4.4 Load Balance Hash Algorithms on 7150 Series Switches

Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among component ECMP routes. The switch supports ***16*** load balance profiles, including the default profile. The default load balance profile is configured through port-channel load-balance fm6000 fields ip and port-channel load-balance fm6000 fields mac commands.

11.2.4.4.1 Load Balance Profiles

Load balance profiles are managed in ***load-balance-policies*** configuration mode. The ***load-balance-policies*** configuration mode provides commands that display the contents of all configured profiles and place the switch in **`load-balance-profile`** command. Load balance profiles are created by entering the ***load-balance-profile*** mode and edited while in that mode.

The load-balance policies command places the switch in ***load-balance-policies*** configuration mode. Load balance profiles specify the inputs used by the hashing algorithms that distribute traffic across ports comprising a port channel or among ECMP routes.

Examples

- This command places the switch in ***load-balance-policies*** configuration mode.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)#
```

- This command displays the contents of the four load balance profiles configured on the switch.

```
switch(config-load-balance-policies)# show active

load-balance policies
   load-balance fm6000 profile F-01
      port-channel hash-seed 22
      fields ip dscp
      distribution random port-channel
   !
   load-balance fm6000 profile F-02
      fields ip protocol dst-ip
      distribution random port-channel
   !
   load-balance fm6000 profile F-03
      fields ip protocol dst-ip
      fields mac dst-mac eth-type
      distribution random ecmp port-channel
   !
   load-balance fm6000 profile F-04

switch(config-load-balance-policies)#
```

Creating a Load Balance Profile

The load-balance fm6000 profile command places the switch in load-balance-profile configuration mode to configure a specified load balance profile. The command specifies the name of the profile that subsequent commands modify. It creates a profile if the profile it references does not exist.

Example

These commands enter ***load-balance-profile*** configuration mode, creates the ***LB-5*** profile, and lists the default settings for the profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance fm6000 profile LB-5
switch(config-load-balance-profile-LB-5)# show active all

load-balance policies
   load-balance fm6000 profile LB-5
      port-channel hash-seed 0
      fields mac dst-mac src-mac eth-type vlan-priority vlan-id
      fields ip protocol dst-ip dst-port src-ip src-port dscp
      no distribution symmetric-hash
      no distribution random

switch(config-load-balance-profile-LB-5)#
```

Configuring a Load Balance Profile

These commands are available in load-balance-profile configuration mode to specify the parameters that comprise a profile.

- The fields ip command specifies the L3/L4 data fields used by the hash algorithm defined by the configuration mode load balance profile.
- The fields mac command specifies the L2 data fields used by the hash algorithm defined by the configuration mode load balance profile.

- The distribution symmetric-hash command enforces traffic symmetry on data distributed by the hash algorithm defined by the configuration mode load balance profile. Symmetric traffic is the flow of both directions of a data stream across the same physical link.
- The distribution random command specifies the random distribution of data packets handled by the hash algorithm defined by the configuration mode load balance profile.

Example

These commands configure the following components of the hash algorithm defined by the LB-7 load balance profile:

- L2 header fields: MAC destination address, VLAN priority.
- L3/L4 header fields: Source IP address, protocol field.
- Symmetric hash distribution of IP and non-IP packets.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance fm6000 profile
 LB-7
switch(config-load-balance-profile-LB-7)# fields ip src-ip protocol
switch(config-load-balance-profile-LB-7)# fields mac dst-mac vlan-
priority
switch(config-load-balance-profile-LB-7)# distribution symmetric-has
h mac-ip
switch(config-load-balance-profile-LB-7)# show active
load-balance policies
   load-balance fm6000 profile LB-7
      fields mac dst-mac vlan-priority
      fields ip protocol src-ip
      distribution symmetric-hash mac-ip
switch(config-load-balance-profile-LB-7)# exit
switch(config-load-balance-policies)# exit
switch(config)# exit
```

Assigning a Load Balance Profile to an Interface

The ingress load-balance profile command applies a specified load-balance profile to the configuration mode interface. Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among ECMP routes. The switch supports 16 load balance profiles, including the default profile.

Example

This command applies the ***LB-1*** load balance profile to ***interface port-channel 100***.

```
switch(config)# interface port-channel 100
switch(config-if-Po100)# ingress load-balance profile LB-1
switch(config-if-Po100)# show active
interface Port-Channel100
   ingress load-balance profile LB-1
switch(config-if-Po100)#
```

11.2.4.4.2 Default Load Balance Profile

Two commands configure the load balance default profile on 7150 Series switches:

- port-channel load-balance fm6000 fields ip controls the hash algorithm for IP packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets).
- port-channel load-balance fm6000 fields mac controls the hash algorithm for non-IP packets by specifying the algorithm's use of MAC header fields. Fields that the command can specify include include the MAC source address, MAC destination address, and Ethernet type, VLAN-ID, and VLAN-priority fields.

Examples

- These commands configure the load balance default profile for IP packets by using source and destination IP address fields, along with source and destination port fields for TCP, and UDP packets.

```
switch(config)# port-channel load-balance fm6000 fields ip ip-tcp-
udp-header
switch(config)#
```

- This command applies the default load balance profile to ***interface port-channel 100***.

```
switch(config)# interface port-channel 100
switch(config-if-Po100)# no ingress load-balance profile
switch(config-if-Po100)# show active
interface Port-Channel100
switch(config-if-Po100)#
```

11.2.5 Port Channel and LACP Configuration Commands

Global Port Channel and LACP Configuration Commands

- interface port-channel
- lacp system-priority

Interface Configuration Commands – Ethernet Interface

- channel-group
- lacp port-priority
- lacp timer
- port-channel lacp fallback
- port-channel lacp fallback timeout
- port-channel min-links
- port-channel min-links review interval
- port-channel speed minimum
- port-channel speed mixed

Load Balance (Default) Commands

- port-channel load-balance
- port-channel load-balance arad fields ip
- port-channel load-balance fm6000 fields ip
- port-channel load-balance fm6000 fields mac
- port-channel load-balance module
- port-channel load-balance petraA fields ip
- port-channel load-balance sand profile (7500E/7500R)
- port-channel load-balance trident fields ip

- port-channel load-balance trident fields ipv6
- port-channel load-balance trident fields mac

Load Balance Policies Commands

- distribution random
- distribution symmetric-hash
- fields ip
- fields mac
- hash-seed
- ingress load-balance profile
- load-balance fm6000 profile
- load-balance policies
- load-balance sand profile (7500E/7500R)
- port-channel hash-seed

EXEC Commands

- show lacp aggregates
- show lacp counters
- show lacp interface
- show lacp internal
- show lacp peer
- show lacp sys-id
- show load-balance profile
- show port-channel
- show port-channel dense
- show port-channel limits
- show port-channel load-balance
- show port-channel load-balance fields

11.2.5.1 channel-group

The **`channel-group`** command assigns the configuration mode Ethernet interfaces to a channel group, creates the group if it does not already exist, and sets the port-channel mode for the group. When adding interfaces to a previously created channel group, the port-channel mode for the newly added interfaces must match the mode for the existing group.

Channel groups are associated with a port channel interface immediately upon their creation. A command that creates a new channel group also creates a port channel with a matching ID. The port channel is configured in Port-channel Configuration Mode. Configuration changes to a port channel interface propagate to all Ethernet interfaces in the corresponding channel group. The interface port-channel command places the switch in the ***interface-port-channel*** configuration mode.

The **`no channel-group`** and **`default channel group`** commands remove the configuration-mode interface from the specified channel group.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`channel-group` ***number*** mode ***group_mode***

`no channel-group`

`default channel-group`

Parameters

- ***number*** Specifies a channel group ID. Values range from 1 through 2000.
- ***group_mode*** Specifies the channel-group mode for the channel group. Values include:
  - **on** Port channel is static and LACP is disabled on member interfaces. Port neither verifies nor negotiates port channel membership.
    - **active** Port channel is dynamic and member interfaces are active LACP ports that transmit and receive LACP negotiation packets.
    - **passive** Port channel is dynamic and member interfaces are passive LACP ports that respond to LACP negotiation packets but do not generate them.

Guidelines: Port Channels

You can configure a port channel to contain many ports, but only a subset may be active at a time. All active ports in a port channel must be compatible. Compatibility includes many factors and is platform-specific. For example, compatibility may require identical operating parameters such as speed and Maximum Transmission Unit (MTU). Compatibility may only be possible between specific ports because of the internal organization of the switch.

Guidelines: MLAG Configurations

Static LAG is not recommended in MLAG configurations. However, these considerations apply when the channel group mode is **on** while configuring static MLAG:

- When configuring multiple interfaces on the same static port channel:
  - all interfaces must physically connect to the same neighboring switch.
  - the neighboring switch must configure all interfaces into the same port channel.

The switches are misconfigured when these conditions are not met.

Disable the static port channel membership before moving any cables connected to these interfaces or changing a static port channel membership on the remote switch.

Examples

- These commands assign Ethernet interfaces ***8*** and ***9*** to channel group ***10***, and enable LACP in negotiating mode.

```
switch(config)# interface ethernet 8-9
switch(config-if-Et8-9)# channel-group 10 mode active
switch(config-if-Et8-9)# show active
interface Ethernet8
   channel-group 10 mode active
interface Ethernet9
   channel-group 10 mode active
switch(config-if-Et8-9)#
```

- These commands assign Ethernet interfaces ***12*** and ***13*** to static channel group ***11***. LACP is disabled on these interfaces.

```
switch(config)# interface ethernet 12-13
switch(config-if-Et12-13)# channel-group 11 mode on
switch(config-if-Et12-13)# show active
interface Ethernet12
   channel-group 11 mode on
interface Ethernet13
   channel-group 11 mode on
switch(config-if-Et12-13)#
```

11.2.5.2 distribution random

The **`distribution random`** command specifies the random distribution of data packets handled by the hash algorithm defined by the configuration mode load balance profile. All data fields and hash seeds that are configured for the profile are used as seeds for the random number generator that defines the distribution of individual packets.

Command options allow for the random distribution of traffic across port channel links and ECMP routes. Random distribution can be enabled for either, both, or neither.

The **`no distribution random`** and **`default distribution random`** commands remove random distribution on the configuration mode load balance profile by deleting the corresponding **`distribution random`** command from the configuration.

Command Mode

Load-balance-profile Configuration

Command Syntax

`distribution random` **BALANCE_TYPE**

`no distribution random`

`default distribution random`

Parameters

SCOPE Specifies use of random distribution for port channels and ECMP routes. Options include:

- ***no parameter*** Random distribution is enabled for ECMP routes and port channel links.
- **ecmp** Random distribution is enabled for ECMP routes.
- **port-channel** Random distribution is enabled for port channel links.
- **port-channel ecmp** Random distribution is enabled for ECMP routes and port channel links.
- **ecmp port-channel** Random distribution is enabled for ECMP routes and port channel links.

Guidelines

The **`distribution random`** command takes precedence over the distribution symmetric-hash command when both methods are simultaneously enabled.

Related Commands

load-balance fm6000 profile places the switch in the ***load-balance-profile*** configuration mode.

Example

These commands configure symmetric hashing on all traffic distributed through the algorithm defined by the ***LB-1*** load balance profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)# distribution random ecmp
 port-channel
switch(config-load-balance-profile-LB-1)# show active
load-balance policies
   load-balance fm6000 profile LB-1
      distribution random ecmp port-channel
switch(config-load-balance-profile-LB-1)#
```

11.2.5.3 distribution symmetric-hash

The **distribution symmetric-hash** command enforces traffic symmetry on data distributed by the hash algorithm defined by the configuration mode load balance profile. Symmetric traffic is the flow of both directions of a data stream across the same physical link.

Two symmetric-hash options specify the traffic upon which symmetry is enforced:

- **distribution symmetric-hash mac** specifies that only non-IP traffic is hashed symmetrically. IP traffic is hashed normally without regard to symmetry.
- **distribution symmetric-hash mac-ip** specifies that all traffic is hashed symmetrically.

The **no distribution symmetric-hash** and **default distribution symmetric-hash** commands remove the specified hashing symmetry restriction on the configuration mode load balance profile by deleting the corresponding **distribution symmetric-hash** command from ***running-config***.

Command Mode

Load-balance-profile

Command Syntax

distribution symmetric-hash FIELD_TYPE

no distribution symmetric-hash

default distribution symmetric-hash

Parameters

FIELD_TYPE Fields the hashing algorithm uses for Layer 3 routing. Options include:

- **mac** Non-IP traffic is hashed symmetrically.
- **mac-ip** All traffic is hashed symmetrically.

Guidelines

The distribution random command takes precedence over the **distribution symmetric-hash** command when both methods are simultaneously enabled.

Related Commands

load-balance fm6000 profile places the switch in the ***load-balance-profile*** configuration mode.

Example

These commands configure symmetric hashing on all traffic distributed through the algorithm defined by the ***LB-1*** load balance profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)# distribution symmetric-hash
 mac-ip
switch(config-load-balance-profile-LB-1)# show active
load-balance policies
   load-balance fm6000 profile LB-1
      distribution symmetric-hash mac-ip
switch(config-load-balance-profile-LB-1)#
```

11.2.5.4 fields ip

The **fields ip** command specifies the L3/L4 data fields used by the hash algorithm defined by the configuration mode load balance profile. When a load balance profile is assigned to a port channel or

Ethernet interface, its associated hash algorithm determines the distribution of packets that ingress the interface. Profile algorithms can load balance packets across port channel links or ECMP routes.

The switch calculates a hash value by using the packet header fields to balance packets across links. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **`no fields ip`** configures the algorithm not to use L3/L4 data fields. The **`default fields ip`** command restores the default data L3/L4 fields to the load balancing algorithm defined by the configuration mode profile by removing the corresponding **`fields ip`** or **`no fields ip`** command from ***running-config***.

Command Mode

Load-balance-profile Configuration

Command Syntax

`fields ip` **IP_FIELD**

`no fields ip`

`default fields ip`

Parameters

IP_FIELD Specifies the L3/L4 fields the hashing algorithm uses. Options include:

- **dscp** Algorithm uses dscp field.
- **dst-ip** Algorithm uses destination IP address field.
- **dst-port** Algorithm uses destination TCP/UDP port field.
- **protocol** Algorithm uses protocol field.
- **src-ip** Algorithm uses source IP address field.
- **src-port** Algorithm uses source TCP/UDP port field.

Command may include from one to six fields, in any combination and listed in any order. The default setting is the selection of all fields.

Related Commands

load-balance fm6000 profile places the switch in the ***load-balance-profile*** configuration mode.

Example

These commands specify the IP source and protocol fields as components of the hash algorithm defined by the ***LB-1*** load balance profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)# fields ip src-ip protocol
switch(config-load-balance-profile-LB-1)# show active
load-balance policies
   load-balance fm6000 profile LB-1
      fields ip protocol src-ip
switch(config-load-balance-profile-LB-1)#
```

11.2.5.5 fields mac

The **fields mac** command specifies the L2 data fields used by the hash algorithm defined by the configuration mode load balance profile. When a load balance profile is assigned to a port channel or Ethernet interface, its associated hash algorithm determines the distribution of packets that ingress the interface. Profile algorithms can load balance packets across port channel links or ECMP routes.

The switch calculates a hash value using the packet header fields to balance packets across links. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no fields mac** configures the algorithm not to use L2 data fields. The **default fields mac** command restores the default data L2 fields to the load balancing algorithm defined by the configuration mode profile by removing the corresponding **fields mac** or **no fields mac** command from ***running-config***.

Command Mode

Load-balance-profile Configuration

Command Syntax

fields mac **MAC_FIELD**

no fields mac

default fields mac

Parameters

MAC_FIELD Specifies the L2 fields the hashing algorithm uses. Options include:

- **dst-mac** Algorithm uses the MAC destination field.
- **eth-type** Algorithm uses the Ethernet port type field.
- **src-mac** Algorithm uses MAC source field.
- **vlan-id** Algorithm uses VLAN ID field.
- **vlan-priority** Algorithm uses VLAN priority field.

Related Commands

The load-balance fm6000 profile command places the switch in to the ***load-balance-profile*** configuration mode.

Example

These commands specify the MAC destination and VLAN priority fields as components of the hash algorithm defined by the ***LB-1*** load balance profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)# fields mac dst-mac vlan-
priority
switch(config-load-balance-profile-LB-1)# show active
load-balance policies
   load-balance fm6000 profile LB-1
      fields mac dst-mac vlan-priority
```

```
switch(config-load-balance-profile-LB-1)#
```

11.2.5.6 hash-seed

The **hash-seed** command specifies the seed used by the hash algorithm defined by the configuration mode load balance profile. Profile algorithms can load balance packets across port channel links or ECMP routes.

The **no hash-seed** and **default hash-seed** commands restore the default hash seed value of 0 to the load balancing algorithm defined by the configuration mode profile by removing the corresponding **hash-seed** command from ***running-config***.

Command Mode

Load-balance-profile Configuration

Command Syntax

hash-seed ***number***

no hash-seed ***number***

default hash-seed ***number***

Parameters

number Specifies the value of the hash seed. Value ranges from ***0*** to ***39***.

Example

These commands configure the ***hash seed 20*** in a profile and assign it as the global profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance sand profile
 myGlobalProfile
switch(config-sand-load-balance-profile-myGlobalProfile)# hash-seed 20
switch(config)# port-channel load-balance sand profile myGlobalProfile
```

11.2.5.7 ingress load-balance profile

The **ingress load-balance profile** command applies the specified load-balance profile to the configuration mode interface. Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among ECMP routes. The switch supports 16 load balance profiles, including the default profile.

Load balance profiles can be assigned to Ethernet and port channel interfaces. Profiles define the distribution method of traffic that ingresses the interface among the ports comprising a port channel or routes comprising an ECMP.

The default load balance profile is configured through port-channel load-balance fm6000 fields ip and port-channel load-balance fm6000 fields mac commands.

The **no ingress load-balance profile** and **default ingress load-balance profile** commands restore the default load balance profile for the configuration mode interface by removing the corresponding **ingress load-balance profile** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`ingress load-balance profile` ***profile_name***

`no ingress load-balance profile`

`default ingress load-balance profile`

Parameters

profile_name Name of profile assigned to interface.

Example

This command applies the ***LB-1*** load balance profile to port channel interface ***100***.

```
switch(config)# interface port-channel 100
switch(config-if-Po100)# show active
interface Port-Channel100

switch(config-if-Po100)# ingress load-balance profile LB-1
switch(config-if-Po100)#
interface Port-Channel100
   ingress load-balance profile LB-1

switch(config-if-Po100)#
```

11.2.5.8 interface port-channel

The **`interface port-channel`** command places the switch in port-channel interface configuration mode for modifying parameters of specified link aggregation (LAG) interfaces. When entering configuration mode to modify existing port channel interfaces, the command can specify multiple interfaces.

The command creates a port channel interface if the specified interface does not exist prior to issuing the command. When creating an interface, the command can only specify a single interface.

The **`no interface port-channel`** and **`default interface port-channel`** commands delete the specified LAG interfaces from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`interface port-channel` ***p_range***

`no interface port-channel` ***p_range***

`default interface port-channel` ***p_range***

Parameter

p_range Port channel interfaces (number, range, or comma-delimited list of numbers and ranges).

Port channel numbers range from ***1*** to ***2000***.

Guidelines

When configuring a port channel, you do not need to issue the **`interface port-channel`** command before assigning a port to the port channel (see the channel-group command). The port channel number is implicitly created when a port is added to the specified port channel with the **`channel-group number`** command.

To display ports that are members of a port channel, enter show port-channel. To view information about hardware limitations for a port channel, enter show port-channel limits.

All active ports in a port channel must be compatible. Compatibility comprises many factors and is specific to a given platform. For example, compatibility may require identical operating parameters such as speed and/or Maximum Transmission Unit (MTU). Compatibility may only be possible between specific ports because of internal organization of the switch.

You can configure a port channel with a set of ports such that more than one subset of the member ports are mutually compatible. Port channels in EOS are designed to activate the compatible subset of ports with the largest aggregate capacity. A subset with two 40 Gbps ports (aggregate capacity 80 Gbps) has preference to a subset with five active 10 Gbps ports (aggregate capacity 50 Gbps).

Example

This example creates ***interface port-channel 3***:

```
switch(config)# interface port-channel 3
switch(config-if-Po3)#
```

11.2.5.9 lacp port-priority

The **`lacp port-priority`** command sets the aggregating port priority for the configuration mode interface. Priority is supported on port channels with LACP-enabled physical interfaces. LACP port priority determines the port that is active in a LAG in fallback mode. Numerically lower values have higher priority.

Each port in an aggregation is assigned a 32-bit port identifier by prepending the port priority (16 bits) to the port number (16 bits). Port priority determines the ports that are placed in standby mode when hardware limitations prevent a single aggregation of all compatible ports.

Priority numbers range from ***0*** to ***65535***. The default is ***32768***. Interfaces with higher priority numbers are placed in standby mode before interfaces with lower priority numbers.

The **`no lacp port-priority`** and **`default lacp port-priority`** commands restore the default port-priority to the configuration mode interface by removing the corresponding **`lacp port-priority`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`lacp port-priority` ***priority_value***

`no lacp port-priority`

`default lacp port-priority`

Parameters

priority_level Port priority. Values range from ***0*** to ***65535***. Default is ***32768***

Example

These commands assign the port priority of ***4096*** to ***interface ethernet 8***.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# lacp port-priority 4096
switch(config-if-Et8)# show active
interface Ethernet8
```

```
   lacp port-priority 4096
switch(config-if-Et8)#
```

11.2.5.10 lacp system-priority

The **lacp system-priority** command configures the switch's LACP system priority. Values range between ***0*** and ***65535***. Default value is ***32768***.

Each switch is assigned a globally unique 64-bit system identifier by prepending the system priority (16 bits) to the MAC address of one of its physical ports (48 bits). Peer devices use the system identifier when forming an aggregation to verify that all links are from the same switch. The system identifier is also used when dynamically changing aggregation capabilities resulting from LACP data; the system with the numerically lower system identifier can dynamically change advertised aggregation parameters.

The **no lacp system-priority** and **default lacp system-priority** commands restore the default system priority by removing the **lacp system-priority** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

lacp system-priority ***priority_value***

no lacp system-priority

default lacp system-priority

Parameters

priority_value System priority number. Values range from ***0*** to ***65535***. Default is ***32768***.

Example

This command assigns the system priority of ***8192*** to the switch.

```
switch(config)# lacp system-priority 8192
switch(config)#
```

11.2.5.11 lacp timer

The **lacp timer** command configures the LACP reception interval on the configuration mode interface. The LACP timeout specifies the reception rate of LACP packets at interfaces supporting LACP. Supported rates include:

- **normal**: 30 seconds with synchronized interfaces; one second while interfaces are synchronizing.
- **fast**: one second.

This command is supported on LACP-enabled interfaces. The default value is **normal**.

The **no lacp timer** and **default lacp timer** commands restore the default value of **normal** on the configuration mode interface by deleting the corresponding **lacp timer** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

lacp timer **RATE_LEVEL**

no lacp timer

default lacp timer

Parameters

RATE_LEVEL LACP reception interval. Options include:

- **fast** One second.
- **normal *30*** seconds for synchronized interfaces; ***1*** second while interfaces synchronize.

Example

This command sets the LACP timer to ***1*** second on ***ethernet interface 4***.

```
switch(config-if-Et4)# lacp timer fast
switch(config-if-Et4)#
```

11.2.5.12 load-balance fm6000 profile

The **load-balance fm6000 profile** command places the switch in load-balance-profile configuration mode to configure a specified load balance profile. The command specifies the name of the profile that subsequent commands modify. It creates a profile if the profile it references does not exist.

Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among component ECMP routes. The switch supports 16 load balance profiles, including the default profile. The default load balance profile is configured through port-channel load-balance fm6000 fields ip and port-channel load-balance fm6000 fields mac commands.

The load balance profile name is referenced when it is applied to an interface. The default profile is not associated with a name and is applied to an interface in the absence of a named profile assignment.

The **no load-balance fm6000 profile** and **default load-balance fm6000 profile** commands delete the specified load balance profile from ***running-config***. Profiles that are assigned to an interface cannot be deleted. Attempts to delete an assigned profile generate a **profile in use** error messages.

The **load-balance fm6000 profile** command is accessible from ***load-balance-policies*** configuration mode. The ***load-balance-profile*** configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***load-balance-policies*** configuration mode does not affect the configuration. The **exit** command returns the switch to the ***load-balance-policies*** configuration mode.

Command Mode

Load-balance-policies Configuration

Command Syntax

load-balance fm6000 profile ***profile_name***

no load-balance fm6000 profile ***profile_name***

default load-balance fm6000 profile ***profile_name***

Parameters

profile_name Name of the load-balance profile.

Commands Available in Load-balance-profile Configuration Mode

- fields ip
- fields mac
- distribution random

- distribution symmetric-hash
- port-channel hash-seed
- **show active** displays the contents of the configuration mode profile.

Related Commands

- The load-balance policies command places the switch in to the ***load-balance-policies*** configuration mode.
- The ingress load-balance profile command applies a load-balance profile to an Ethernet or port channel interface.
- The show load-balance profile command displays the contents of load balance profiles.

Example

These commands enters the ***load-balance-profile*** configuration mode, creates the **LB-1** profile, and lists the default settings for the profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)# show active all
load-balance policies
   load-balance fm6000 profile LB-1
      port-channel hash-seed 0
      fields mac dst-mac src-mac eth-type vlan-priority vlan-id
      fields ip protocol dst-ip dst-port src-ip src-port dscp
      no distribution symmetric-hash
      no distribution random
switch(config-load-balance-profile-LB-1)#
```

11.2.5.13 load-balance policies

The **load-balance policies** command places the switch in load-balance-policies configuration mode. Load-balance-policies configuration mode provides commands for managing load-balance profiles. Load balance profiles specify the inputs used by the hashing algorithms that distribute traffic across ports comprising a port channel or among ECMP routes.

The **no load-balance policies** and **default load-balance policies** commands delete all load balance profiles from ***running-config***. The command generates an error message when at least one profile is assigned to an interface.

Load-balance-policies configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***load-balance-policies*** configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

load-balance policies

no load-balance policies

default load-balance policies

Commands Available in Load-balance-policies Configuration Mode

- load-balance fm6000 profile places the switch in ***load-balance-profile*** configuration mode.
- **show active** displays contents of all load balance profiles.

Related Commands

- The ingress load-balance profile command applies a load-balance profile to an Ethernet or port channel interface.
- The show load-balance profile command displays the contents of load balance profiles.

Examples

- This command places the switch in the ***load-balance-policies*** configuration mode.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)#
```

- This command displays the contents of the three configured load balance profiles.

```
switch(config-load-balance-policies)# show active

load-balance policies
   load-balance fm6000 profile F-01
      port-channel hash-seed 22
      fields ip dscp
      distribution random port-channel
   !
   load-balance fm6000 profile F-02
      fields ip protocol dst-ip
      fields mac dst-mac eth-type
      distribution random ecmp port-channel
   !
   load-balance fm6000 profile F-03

switch(config-load-balance-policies)#
```

11.2.5.14 load-balance sand profile (7500E/7500R)

The **`load-balance sand profile`** command configures a load-balance profile on a sand module switch. A default profile is designated as a global profile when no other profile is set as global profile. Note, a warning message is displayed when a profile is entered or deleted.

If **`no load-balance sand profile`** command is executed when the profile set is default then the following warning message is displayed:

```
! profile default is a reserved profile and cannot be deleted
```

Command Mode

Global Configuration

Command Syntax

`load-balance sand profile` ***profile_name***

`no load-balance sand profile` ***profile_name***

Parameter

profile_name Name of the profile assigned to the selected module.

Examples

- These commands designate a default profile as a global profile on sand module platform switch. Note, a warning message is displayed when a profile is entered or deleted.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance sand profile
 default
  ! profile default is a reserved profile
! profile default is the current global profile
```

- When no form of the command is executed it displays the following warning message.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# no load-balance sand profile
 default
! profile default is a reserved profile and cannot be deleted
```

11.2.5.15 port-channel hash-seed

The **`port-channel hash-seed`** command specifies the seed used by the hash algorithm defined by the configuration mode load balance profile when distributing the load across ports comprising a port channel. When a load balance profile is assigned to a port channel or Ethernet interface, its associated hash algorithm determines the distribution of packets that ingress the interface. Profile algorithms can load balance packets across port channel links or ECMP routes.

The hash seed that the algorithm uses to select port channel links or ECMP routes is configured by the ip load-sharing command.

The **`no port-channel hash-seed`** and **`default port-channel hash-seed`** commands restore the default hash seed value of ***0*** to the load balancing algorithm defined by the configuration mode profile by removing the corresponding **`port-channel hash-seed`** command from ***running-config***.

Command Mode

Load-balance-profile Configuration

Command Syntax

`port-channel hash-seed` ***number***

`no port-channel hash-seed`

`default port-channel hash-seed`

Parameters

number The hash seed. Value ranges from ***0*** to ***39***.

Related Commands

The load-balance fm6000 profile command places the switch in to the ***load-balance-profile*** configuration mode.

Example

Thes commands configure the port-channel hash seed of ***22*** for the hash algorithm defined by the ***LB-1*** load balance profile.

```
switch(config)# load-balance policies
switch(config-load-balance-policies)# load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)# port-channel hash-seed 22
switch(config-load-balance-profile-LB-1)# show active
load-balance policies
```

```
   load-balance fm6000 profile LB-1
      port-channel hash-seed 22
switch(config-load-balance-profile-LB-1)#
```

11.2.5.16 port-channel lacp fallback

The **`port-channel lacp fallback`** command enables the LACP fallback mode on the interface.

LACP fallback is unconfigured and disabled by default. An LACP interface without fallback enabled does not form a LAG until it receives PDUs from its peer.

LACP fallback can be configured on an interface in static or individual mode:

- **static mode** The port channel member with the lowest LACP port priority is active and maintains contact with the peer (sending and receiving data) while other port channel members remain in standby mode until a LACP PDU is received. All members continue to send (and can receive) LACP PDUs.
- **individual mode** All port channel members act as individual ports, reverting to their port-specific configuration while the channel is in fallback mode, and continue to send and receive data. All members continue to send LACP PDUs until a LACP PDU is received by one of the member ports.

The **`no port-channel lacp fallback`** and **`default port-channel lacp fallback`** commands disable LACP fallback mode on the configuration mode interface by removing the corresponding **`port-channel lacp fallback`** command from ***running-config***.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

`port-channel lacp fallback` [**MODE**]

`no port-channel lacp fallback`

`default port-channel lacp fallback`

Parameters

MODE LACP fallback mode. Options include:

- ***no parameter*** Enables static LACP fallback mode.
  - **static** Enables static LACP fallback mode.
  - **individual** Eenables individual LACP fallback mode.

Related Commands

- The port-channel lacp fallback timeout command configures the fallback timeout period for a port channel interface. The default LACP fallback timeout period is ***90*** seconds.
- The lacp port-priority command configures the port priority for an individual interface.

Examples

- These commands enable LACP static fallback mode, then configure an LACP fallback timeout of 100 seconds on ***interface port-channel 13***. If LACP negotiation fails, only the member port with the lowest LACP priority will remain active until an LACP PDU is received by one of the member ports.

```
switch(config)# interface port-channel 13
switch(config-if-Po13)# port-channel lacp fallback static
switch(config-if-Po13)# port-channel lacp fallback timeout 100
switch(config-if-Po13)# show active
interface Port-Channel13
```

```
   port-channel lacp fallback static
   port-channel lacp fallback timeout 100
switch(config-if-Po13)#
```

- These commands enable LACP individual fallback mode, then configure an LACP fallback timeout of ***50*** seconds on ***interface port-channel 17***. If LACP negotiation fails, all member ports will act as individual switch ports, using port-specific configuration, until a LACP PDU is received by one of the member ports.

```
switch(config)# interface port-channel 17
switch(config-if-Po17)# port-channel lacp fallback individual
switch(config-if-Po17)# port-channel lacp fallback timeout 50
switch(config-if-Po17)# show active
interface Port-Channel17
   port-channel lacp fallback individual
   port-channel lacp fallback timeout 50
switch(config-if-Po17)#
```

11.2.5.17 port-channel lacp fallback timeout

The **`port-channel lacp fallback timeout`** command specifies the fallback timeout period for the configuration mode interface.

Fallback timeout settings persist in ***running-config*** without taking effect for interfaces that are not configured into fallback mode. The default fallback timeout period is ***90*** seconds.

The **`no port-channel lacp fallback timeout`** and **`default port-channel lacp fallback timeout`** commands restore the default fallback timeout of ***90*** seconds for the configuration mode interface by removing the corresponding **`port-channel lacp fallback timeout`** command from ***running-config***.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

`port-channel lacp fallback timeout` ***period***

`no port-channel lacp fallback timeout`

`default port-channel lacp fallback timeout`

Parameters

period Maximum interval between receipt of LACP PDU packets (seconds). Value ranges from ***1*** to ***300*** seconds. Default value is ***90***.

Related Commands

The port-channel lacp fallback command configures fallback mode for a port channel interface.

Guidelines

The fallback timeout period should not be shorter than the LACP reception interval (lacp timer). The default LACP reception interval is ***30*** seconds.

Example

This command enables LACP fallback mode, then configures an LACP fallback timeout of ***100*** seconds on ***interface port-channel 13***.

```
switch(config)# interface port-channel 13
```

```
switch(config-if-Po13)# port-channel lacp fallback
switch(config-if-Po13)# port-channel lacp fallback timeout 100
switch(config-if-Po13)# show active
interface Port-Channel13
   port-channel lacp fallback
   port-channel lacp fallback timeout 100
switch(config-if-Po13)#
```

11.2.5.18 port-channel load-balance

The **`port-channel load-balance`** command specifies the seed in the hashing algorithm that balances the load across ports comprising a port channel. Available seed values vary by switch platform.

The **`no port-channel load-balance`** and **`default port-channel load-balance`** commands remove the **`port-channel load-balance`** command from ***running-config***, restoring the default hash seed value of ***0***.

Command Mode

Global Configuration

Command Syntax

`port-channel load-balance` ***platform*** { ***hash_seed*** | fields ip ***fields*** | hash ***hash_function*** }

`no port-channel load-balance` ***platform*** [***hash_seed***]

`default port-channel load-balance` ***platform*** [***hash_seed***]

Parameters

Note: Parameter options vary by switch model. Verify available options with the **`?`** command.

- ***platform*** ASIC switching device. Value depends on the switch model.
- ***hash_seed*** The numerical seed for the hash function. Value range varies by switch platform:
  - **arad** ***0*** to ***65535***.
  - **fm6000** ***0*** to ***39***.
  - **petraA** Uses field inputs only.
  - **trident** ***0*** to ***47***.

  For trident platform switches, algorithms using hash seeds between ***0*** and *15* typically result in more effective distribution of data streams across the port channels.
- ***fields*** Which fields will be used as inputs to the port channel hash.
  - **gre** Configure which GRE fields are inputs to the hash.
  - **ip** Configure which fields are inputs to the hash for IPv4 packets.
  - **ipv6** Configure which fields are inputs to the hash for IPv6 packets.
  - **mac** Configure which MAC fields are inputs to the hash.
  - **mac-in-mac** Configure which MAC-in-MAC fields are inputs to the hash.
  - **mpls** Configure which MPLS fields are inputs to the hash.
  - **destination-ip** Use the Layer 3 IP destination address in the hash.
  - **destination-port** Use the Layer 4 TCP/UDP destination port in the hash.
  - **dst-ip** Use the destination IP address in the hash.
  - **dst-mac** Use the destination Payload MAC in the hash (or the destination MAC address in the MAC hash).
  - **eth-type** Use the Ethernet type in the MAC hash.
  - **ip-in-ip** Use the outer IP header in the hash for IPv4 over IPv4 GRE tunnel.
  - **ip-in-ipv6** Use the outer IP header in the hash for IPv4 over IPv6 GRE tunnel.

- **ipv6-in-ip** Use the outer IP header in the hash for IPv6 over IPv4 GRE tunnel.
- **ipv6-in-ipv6** Use the outer IP header in the hash for IPv6 over IPv6 GRE tunnel.
- **ip-tcp-udp-header** Use the Layer 3 and Layer 4 hashes.
- **isid** Use the MAC-in-MAC ISID in the hash.
- **label** Use the MPLS label in the hash.
- **mac-header** Use the MAC hash.
- **outer-mac** Use the outer MAC of source and destination in the hash.
- **source-ip** Use the Layer 3 IP source address in the hash.
- **src-ip** Use the source IP address in the hash.
- **source-port** Use l\Layer 4 TCP/UDP source port in the hash.
- **src-mac** Use the source payload MAC in the hash (or the source MAC address in the MAC hash).

- ***hash_function*** Specifies the hash polynomial function. Values range from ***0***-***2***.

Example

This command configures a hash seed of ***10*** on an FM6000 platform switch.

```
switch(config)# port-channel load-balance fm6000 10
switch(config)#
```

11.2.5.19 port-channel load-balance arad fields ip

The **port-channel load-balance arad fields ip** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IP packets on Arad platform switches. The hashing algorithm fields used for IP packets differ from the fields used for non-IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance arad fields ip** and **default port-channel load-balance arad fields ip** commands restore the default data fields for the IP packet load balancing algorithm by removing the **port-channel load-balance arad A fields ip** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance arad fields ip **IP_FIELD_NAME**

no port-channel load-balance arad fields ip

default port-channel load-balance arad fields ip

Parameters

IP_FIELD_NAME Fields the hashing algorithm uses for Layer 3 routing. Options include:

- **ip-tcp-udp-header** Algorithm uses source and destination IP address fields. Source and destination port fields are included for TCP and UDP packets.

- **mac-header** Algorithm uses entire MAC header.

  A command can only specify one option. The default setting is **ip-tcp-udp-header**.

Guidelines

The port channel hash algorithm for non-IP packets is not configurable and always includes the entire MAC header.

Related Command

The port-channel load-balance command configures the hash seed for the algorithm.

Example

These commands configure the switch's port channel load balance hash algorithm for IP packets to use source and destination IP address (and port) fields.

```
switch(config)# port-channel load-balance fm6000 fields ip ip-tcp-udp-
header
switch(config)#
```

11.2.5.20 port-channel load-balance fm6000 fields ip

The **port-channel load-balance fm6000 fields ip** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IP packets on FM6000 platform switches. The hashing algorithm fields used for IP packets differ from the fields used for non-IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance fm6000 fields ip** and **default port-channel load-balance fm6000 fields ip** commands restore the default data fields for the IP packet load balancing algorithm by removing the **port-channel load-balance fm6000 fields ip** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance fm6000 fields ip **IP_FIELD_NAME**

no port-channel load-balance fm6000 fields ip

default port-channel load-balance fm6000 fields ip

Parameters

IP_FIELD_NAME Specifies fields the hashing algorithm uses for layer 3 routing. Options include:

- **ip-tcp-udp-header** Algorithm uses source and destination IP address fields. Source and destination port fields are included for TCP and UDP packets.

A command can only specify one option. The default setting is **ip-tcp-udp-header**.

Related Commands

- The port-channel load-balance command configures the hash seed for the algorithm.
- The port-channel load-balance fm6000 fields mac command controls the hash algorithm for non-IP packets.

Example

These commands configure the switch's port channel load balance for IP packets by source and destination IP address and port fields.

```
switch(config)# port-channel load-balance fm6000 fields ip ip-tcp-udp-header
switch(config)#
```

11.2.5.21 port-channel load-balance fm6000 fields mac

The **port-channel load-balance fm6000 fields mac** command specifies data fields that configure the port channel load balance hash algorithm for non-IP packets on FM6000 platform switches. The hashing algorithm fields used for balancing non-IP packets differ from the fields used for IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance fm6000 fields mac** and **default port-channel load-balance fm6000 fields mac** commands restore the default data fields for the non-IP packet load balancing algorithm by removing the **port-channel load-balance fm6000 fields mac** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance fm6000 fields mac **MAC_FIELD_NAME**

no port-channel load-balance fm6000 fields mac

default port-channel load-balance fm6000 fields mac

Parameters

MAC_FIELD_NAME Fields the hashing algorithm uses for Layer 2 routing. Options include:

- **dst-mac** MAC destination field.
- **eth-type** EtherType field.
- **src-mac** MAC source field.
- **vlan-id** VLAN ID field.
- **vlan-priority** VLAN priority field.

Command may include from one to five fields, in any combination and listed in any order. The default setting is the selection of all fields.

Related Commands

- The port-channel load-balance command configures the hash seed for the algorithm.
- The port-channel load-balance fm6000 fields ip command controls the hash algorithm for IP packets.

Example

These commands configure the switch's port channel load balance for non-IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
switch(config)# port-channel load-balance fm6000 fields mac dst-mac
 eth-type
switch(config)#
```

11.2.5.22 port-channel load-balance module

The **port-channel load-balance module** command assigns a named load-balancing profile to a linecard.

Note: Available on the 7500E platform.

The **no port-channel load-balance module** and **default port-channel load-balance module** commands unassigns the load balancing module, or restores the default data fields for the load balancing module.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance module **LINECARD_RANGE** sand profile **PROFILE_NAME**

no port-channel load-balance module **LINECARD_RANGE** sand profile **PROFILE_NAME**

default port-channel load-balance module **LINECARD_RANGE** sand profile **PROFILE_NAME**

Parameters

- **LINECARD_RANGE** Linecard number range includes:
  - ***3-10*** Linecard number range.
- **PROFILE_NAME** Load-balance profile name.

Examples

- This command assigns a named load-balancing profile to a linecard.

```
switch(config)# port-channel load-balance module 3-7 sand profile
 Linecard5
switch(config)#
```

- This command unassigns a named load-balancing profile to a linecard.

```
switch(config)# no port-channel load-balance module 3-7 sand profile
 Linecard5
switch(config)#
```

11.2.5.23 port-channel load-balance petraA fields ip

The **port-channel load-balance petraA fields ip** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IP packets on Petra platform switches. The hashing algorithm fields used for IP packets differ from the fields used for non-IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance petraA fields ip** and **default port-channel load-balance petraA fields ip** commands restore the default data fields for the IP packet load balancing algorithm by removing the **port-channel load-balance petraA fields ip** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance petraA fields ip **IP_FIELD_NAME**

no port-channel load-balance petraA fields ip

default port-channel load-balance petraA fields ip

Parameters

IP_FIELD_NAME Fields the hashing algorithm uses for Layer 3 routing. Options include:

- **ip-tcp-udp-header** Algorithm uses source and destination IP address fields. Source and destination port fields are included for TCP and UDP packets.
- **mac-header** Algorithm uses entire MAC header.

A command can only specify one option. The default setting is **ip-tcp-udp-header**.

Guidelines

The port channel hash algorithm for non-IP packets is not configurable and always includes the entire MAC header.

Related Command

The port-channel load-balance command configures the hash seed for the algorithm.

Example

These commands configure the switch's port channel load balance hash algorithm for IP packets to use source and destination IP address (and port) fields.

```
switch(config)# port-channel load-balance fm6000 fields ip ip-tcp-udp-
header
switch(config)#
```

11.2.5.24 port-channel load-balance sand profile (7500E/7500R)

The **port-channel load-balance sand profile** command configures a global LAG hashing profile on the port channel interface. A default profile is set as a global profile when no other profile is set as global.

The **no port-channel load-balance sand profile** command removes the active profile from the **port-channel load-balance** command from ***running-config***, restoring the default profile.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance sand profile ***profile_name***

no port-channel load-balance sand profile ***profile_name***

Parameter

profile_name Name of the profile assigned to the selected module.

Example

This command configures a global LAG hashing profile on 7500 series platform switch.

```
switch(config)# port-channel load-balance sand profile myGlobalProfile
switch(config)#
```

11.2.5.25 port-channel load-balance trident fields ip

The **port-channel load-balance trident fields ip** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IP packets on Trident platform switches. The hashing algorithm fields used for IP packets differ from the fields used for non-IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance trident fields ip** and **default port-channel load-balance trident fields ip** commands restore the default data fields for the IP packet load balancing algorithm by removing the **port-channel load-balance trident fields ip** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance trident fields ip **IP_FIELD_NAME**

no port-channel load-balance trident fields ip

default port-channel load-balance trident fields ip

default port-channel load-balance trident fields ip ingress-interface ***disabled***

Parameters

- **IP_FIELD_NAME** Specifies fields the hashing algorithm uses for Layer 3 routing. Command may include from one to four of the following four options, in any combination and listed in any order.
  - **destination-ip** Algorithm uses destination IP address field.
  - **source-ip** Algorithm uses source IP address field.
  - **destination-port** Agorithm uses destination TCP/UDP port field.
  - **source-port** Algorithm uses source TCP/UDP port field.
    - **ip-tcp-udp-header** Algorithm uses source and destination IP address fields. Source and destination port fields are included for TCP and UDP packets.

      Note: This option cannot be used in combination with any other option.
    - **mac-header** Algorithm uses fields specified by port-channel load-balance trident fields mac.

      Note: This option cannot be used in combination with any other option.
  - **ingress-interface** Disable from LAG hashing.

Default setting is **ip-tcp-udp-header**

Related Commands

- The port-channel load-balance command configures the hash seed for the algorithm.
- The port-channel load-balance trident fields ipv6 command controls the hash algorithm for IPv6 packets.
- The port-channel load-balance trident fields mac command controls the hash algorithm for non-IP/IPv6 packets.

Examples

- These commands configure the switch's port channel load balance for IP packets by using the IPv6 destination field in the hashing algorithm.

```
switch(config)# port-channel load-balance trident fields ip
 destination-ip
switch(config)#
```

- This command disables the ingress interface for IPv4 traffic.

```
switch(config)# port-channel load-balance trident fields ip ingress-
interface disabled
switch(config)#
```

11.2.5.26 port-channel load-balance trident fields ipv6

The **`port-channel load-balance trident fields ipv6`** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IPv6 packets on Trident platform switches. The hashing algorithm fields used for IPv6 packets differ from the fields used for non-IPv6 packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance trident fields ipv6** and **default port-channel load-balance trident fields ipv6** commands restore the default data fields for the IPv6 packet load balancing algorithm by removing the **port-channel load-balance trident fields ipv6** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance trident fields ipv6 **IP_FIELD_NAME**

no port-channel load-balance trident fields ipv6

default port-channel load-balance trident fields ipv6

Parameters

- **IP_FIELD_NAME** Specifies fields the hashing algorithm uses for Layer 3 routing. Command may include from one to four of the following four options, in any combination and listed in any order.
  - **destination-ip** Algorithm uses destination IPv6 address field.
  - **source-ip** Algorithm uses source IPv6 address field.
  - **destination-port** Algorithm uses destination TCP/UDP port field.
  - **source-port** Algorithm uses source TCP/UDP port field.
    - **ip-tcp-udp-header** Algorithm uses source and destination IPv6 address fields. Source and destination port fields are included for TCP and UDP packets.

      Note: This option can't be used in combination with any other option.
    - **mac-header** Algorithm uses fields specified by port-channel load-balance trident fields mac.

      Note: This option can't be used in combination with any other option.
    - **ingress-interface** Disable from LAG hashing.

    Default setting is **ip-tcp-udp-header**

Related Commands

- The port-channel load-balance command configures the hash seed for the algorithm.
- The port-channel load-balance trident fields ipv6 commands controls the hash algorithm for non-IP packets.
- The port-channel load-balance trident fields mac command controls the hash algorithm for non-IP packets.

Examples

- These commands configure the switch's port channel load balance for IP packets by using the IPv6 source field in the hashing algorithm.

```
switch(config)# port-channel load-balance trident fields ipv6
 source-ip
switch(config)#
```

- This command disables the ingress interface for IPv6 traffic.

```
switch(config)# port-channel load-balance trident fields ipv6
 ingress-interface disabled
switch(config)#
```

11.2.5.27 port-channel load-balance trident fields mac

The **port-channel load-balance trident fields mac** command specifies data fields that the port channel load balance hash algorithm uses for distributing non-IP packets on Trident platform switches. The hashing algorithm fields used for non-IP packets differ from the fields used for IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance trident fields mac** and **default port-channel load-balance trident fields mac** commands restore the default data fields for the non-IP packet load balancing algorithm by removing the **port-channel load-balance trident fields mac** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

port-channel load-balance trident fields mac **MAC_FIELD_NAME**

no port-channel load-balance trident fields mac

default port-channel load-balance trident fields mac

default port-channel load-balance trident fields mac ingress-interface ***disabled***

Parameters

- **MAC_FIELD_NAME** Fields the hashing algorithm uses for Layer 2 routing. Options include:
  - **dst-mac** MAC destination field.
  - **eth-type** EtherType field.
  - **src-mac** MAC source field.
  - **ingress-interface** Disable from LAG hashing.

  Command may include from one to three fields, in any combination and listed in any order. The default setting is the selection of all fields.

Related Commands

- port-channel load-balance configures the hash seed for the algorithm.
- port-channel load-balance trident fields ip controls the hash algorithm for IP packets.
- port-channel load-balance trident fields ipv6 controls the hash algorithm for IP packets.

Examples

- These commands configure the switch's port channel load balance for non-IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
switch(config)# port-channel load-balance trident fields mac dst-mac
 eth-type
switch(config)#
```

- This command disables the ingress interface for IPv4 traffic.

```
switch(config)# port-channel load-balance trident fields mac
 ingress-interface disabled
switch(config)#
```

11.2.5.28 port-channel min-links

The **port-channel min-links** command specifies the minimum number of interfaces that the configuration mode LAG requires to become active. If there are fewer ports than specified by this command, the port channel interface does not become active. The default min-links value is ***0***.

The **no port-channel min-links** and **default port-channel min-links** commands restore the default min-links setting for the configuration mode LAG by removing the corresponding **port-channel min-links** command from the configuration.

Note: In static LAGs, the min-links value must be met for the LAG to be active. The LAG will not become active until it has at least the min-links number of functioning links in the channel group. If failed links cause the number to drop below the minimum, the LAG will go down and administrator action will be required to bring it back up. In dynamic LAGs, the LACP protocol must determine that at least min-links physical ports are aggregable (they are physically compatible and have the same keys both remotely and locally) before it begins negotiating to make any ports active members of the port-channel. However once negotiation begins, an error on the partner's side or an error in programming of member interfaces can cause the LAG to become active with fewer than the minimum number of links. EOS evaluates min-links after min-links-review-timeout (linearly proportional to configured min-links) when LACP protocol collecting and/or distributing state changes. If the number of active member interfaces in a port-channel is less than configured min-links, it brings the corresponding port-channel Link Down and syslogs **LAG-4-MINLINK_INTF_INSUFFICIENT** message. If additional interfaces get programmed as collecting and distributing, EOS re-evaluates min-links on the port-channel. If sufficient number of interfaces are available to be a part of port-channel, then all interfaces of the corresponding port-channel are re-enabled for LACP negotiation and the port-channel becomes Link Up. **LAG-4-MINLINK_INTF_NORMAL** is syslogged after min-links-review-timeout if the min-links condition is satisfied; otherwise **LAG-4-MINLINK_INTF_INSUFFICIENT** is syslogged and the port-channel goes Link Down. If an interface remains in collecting state but not in distributing state for min-links-review-timeout, it is moved out of collecting state. It is periodically re-enabled after min-links-retry-timeout (which is ***360s*** seconds) till it progresses to collecting and distributing. Meanwhile, if a port-channel becomes Link Up because sufficient number of interfaces progressed to collecting and distributing states, then this interface is enabled for LACP negotiation.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

port-channel min-links ***quantity***

no port-channel min-links

default port-channel min-links

Parameters

quantity Minimum number of interfaces. Value range varies by platform. Default value is ***0***.

Example

These commands set ***4*** as the minimum number of ports required for ***port channel 13*** to become active.

```
switch(config)# interface port-channel 13
switch(config-if-Po13)# port-channel min-links 4
switch(config-if-Po13)# show active
interface Port-Channel13
   port-channel min-links 4
switch(config-if-Po13)#
```

11.2.5.29 port-channel min-links review interval

The **`port-channel min-links review interval`** command enables or disables timer based min-links review feature for all port-channels.

The **`no port-channel min-links review interval`** and **`default port-channel min-links review interval`**commands restore the default min-links-timeout-base to 180 seconds by removing the corresponding **`port-channel min-links review interval`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`port-channel min-links review interval` ***timeout (seconds)***

`no port-channel min-links review interval`

`default port-channel min-links review interval`

Guidelines

The min-links-timeout-base interval for port-channels can be set within the range of ***0*** to ***600*** seconds. When setting the review interval to zero, the command has the following effect:

- Disables the timer-based min-links review feature for all port-channels.
  - For LACP port-channels, it prevents the port-channel from bringing link up (even after one or more member ports were negotiated to collect or distribute (rx or tx)) until there are sufficient member interfaces ready to join the port-channel. Meanwhile, the partner can enable the port-channel link with fewer than required member interfaces. This configuration does not impact port-channels without min-links configuration.

Related Command

port-channel min-links

Example

This command sets the port-channel min-links interval to ***200*** seconds.

```
switch(config)# port-channel min-links review interval 200
```

11.2.5.30 port-channel speed mixed

The **`port-channel speed mixed`** command configures a port channel with the ability to have active members of multiple speeds.

Note: Available on the 7020, 7280, 7500, and 7800 platforms. Minimum links is not available on mixed-speed port channels.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

`port-channel speed mixed`

Related Commands

The interface port-channel command places the switch in the ***interface-port-channel*** configuration mode.

Example

These commands place the switch in the ***interface port-channel*** mode and configure the mixed speed port-channel.

```
switch(config)# interface port-channel 1
switch(config-if-Po1)# port-channel speed mixed
```

11.2.5.31 port-channel speed minimum

The **`port-channel speed minimum`** command specifies the cumulative minimum speed of all active members in order for a port channel to become active. If there is less than the specified by this command, the port channel interface does not become active.

Note: If both minimum speed and minimum links are configured, minimum speed will take precedence.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

`port-channel speed minimum` ***speed-value***

Parameter

speed-value Minimum speed value. The value ranges from ***1*** to ***65535***.

Related Command

The interface port-channel command places the switch in interface-port-channel configuration mode.

Example

These command sets ***100*** Gbps as the minimum speed needed for port channel ***1*** to become active.

```
switch(config)# interface port-channel 1
switch(config-if-Po1)# port-channel speed minimum 100 gbps
```

11.2.5.32 show lacp aggregates

The **`show lacp aggregates`** command displays aggregate IDs and the list of bundled ports for all specified port channels.

Command Mode

EXEC

Command Syntax

`show lacp` [**PORT_LIST**] aggregates [**PORT_LEVEL**] [**INFO_LEVEL**]

Note: **PORT_LEVEL** and **INFO_LEVEL** parameters can be placed in any order.

Parameters

- **PORT_LIST** Port channels for which aggregate information is displayed. Options include:
  - ***<no parameter>*** All configured port channels.
  - ***c_range*** Channel list (number, range, or comma-delimited list of numbers and ranges).
- **PORT_LEVEL** Ports displayed, in terms of aggregation status. Options include:
  - ***no parameter*** Ports bundled by LACP into the port channel.
  - **all-ports** All channel group ports, including channel group members not bundled into the port channel interface.
- **INFO_LEVEL** Amount of information that is displayed. Options include:
  - ***no parameter*** Aggregate ID and bundled ports for each channel.
  - **brief** Aggregate ID and bundled ports for each channel.
  - **detailed** Aggregate ID and bundled ports for each channel.

Example

This command lists aggregate information for all configured port channels.

```
switch> show lacp aggregates

Port Channel Port-Channel1:
 Aggregate ID:
[(8000,00-1c-73-04-36-d7,0001,0000,0000),(8000,00-1c-73-09-a0-f3,0001,0000,0000)]
  Bundled Ports: Ethernet43 Ethernet44 Ethernet45 Ethernet46
Port Channel Port-Channel2:
 Aggregate ID:
[(8000,00-1c-73-01-02-1e,0002,0000,0000),(8000,00-1c-73-04-36-d7,0002,0000,0000)]
  Bundled Ports: Ethernet47 Ethernet48
Port Channel Port-Channel3:
 Aggregate ID:
[(8000,00-1c-73-04-36-d7,0003,0000,0000),(8000,00-1c-73-0c-02-7d,0001,0000,0000)]
  Bundled Ports: Ethernet3 Ethernet4
Port Channel Port-Channel4:
 Aggregate ID:
[(0001,00-22-b0-57-23-be,0031,0000,0000),(8000,00-1c-73-04-36-d7,0004,0000,0000)]
  Bundled Ports: Ethernet1 Ethernet2
Port Channel Port-Channel5:
 Aggregate ID:
[(0001,00-22-b0-5a-0c-51,0033,0000,0000),(8000,00-1c-73-04-36-d7,0005,0000,0000)]
  Bundled Ports: Ethernet41 switch>
```

11.2.5.33 show lacp counters

The **`show lacp counters`** command displays LACP traffic statistics.

Command Mode

EXEC

Command Syntax

`show lacp` [**PORT_LIST**] counters [**PORT_LEVEL**] [**INFO_LEVEL**]

Note: **PORT_LEVEL** and **INFO_LEVEL** parameters can be interchanged while running the command.

Parameters

- **PORT_LIST** Ports for which port information is displayed. Options include:
  - ***no parameter*** All configured port channels.

- ***c_range*** Ports in specified channel list (number, number range, or list of numbers and ranges).
- **interface** Ports on all interfaces.
- **interface ethernet *e_num*** Port on Ethernet interface specified by ***e_num***.
- **interface port-channel *p_num*** Port on port channel interface specified by ***p_num***.

- **PORT_LEVEL** Ports displayed, in terms of aggregation status. Options include:
  - ***no parameter*** Only ports bundled by LACP into an aggregate.
  - **all-ports** All ports, including LACP candidates that are not bundled.
- **INFO_LEVEL** Amount of information that is displayed. Options include:
  - ***no parameter*** Displays packet transmission (TX and RX) statistics.
  - **brief** Displays packet transmission (TX and RX) statistics.
  - **detailed** Displays packet transmission (TX and RX) statistics and actor-partner statistics.

Example

This command displays transmission statistics for all configured port channels.

```
switch> show lacp counters brief

                      LACPDUs         Markers   Marker Response
Port   Status        RX         TX   RX    TX   RX    TX   Illegal
---------------------------------------------------------------------
Port Channel Port-Channel1:
Et43   Bundled   396979     396959    0     0    0     0    0
Et44   Bundled   396979     396959    0     0    0     0    0
Et45   Bundled   396979     396959    0     0    0     0    0
Et46   Bundled   396979     396959    0     0    0     0    0

Port Channel Port-Channel2:
Et47   Bundled   396836     396883    0     0    0     0    0
Et48   Bundled   396838     396883    0     0    0     0    0 switch>
```

11.2.5.34 show lacp interface

The **`show lacp interface`** command displays port status for all port channels that include the specified interfaces. Within the displays for each listed port channel, the output displays sys-id, partner port, state, actor port, and port priority for each interface in the channel.

Command Mode

EXEC

Command Syntax

`show lacp interface` [**INTERFACE_PORT**] [**PORT_LEVEL**] [**INFO_LEVEL**]

Note: INTERFACE_PORT is listed first when present. Other parameters can be listed in any order.

Parameters

- **INTERFACE_PORT** Interfaces for which information is displayed. Options include:
  - ***no parameter*** All interfaces in channel groups.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **port-channel *p_num*** Port channel interface specified by ***p_num***.
- **PORT_LEVEL** Ports displayed, in terms of aggregation status. Options include:
  - ***no parameter*** Command lists data for ports bundled by LACP into the aggregate.
  - **all-ports** Command lists data for all ports, including LACP candidates that are not bundled.
- **INFO_LEVEL** Amount of information that is displayed. Options include:

- ***no parameter*** Displays same information as **brief** option.
- **brief** Displays LACP configuration data, including sys-id, actor, priorities, and keys.
- **detailed** Includes **brief** option information plus state machine data.

Example

This command displays LACP configuration information for all ethernet interfaces.

```
switch> show lacp interface
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state

             |                        Partner                          Actor
Port Status  | Sys-id                  Port# State   OperKey PortPri  Port#
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled | 8000,00-1c-73-09-a0-f3    43 ALGs+CD  0x0001   32768   43
Et44 Bundled | 8000,00-1c-73-09-a0-f3    44 ALGs+CD  0x0001   32768   44
Et45 Bundled | 8000,00-1c-73-09-a0-f3    45 ALGs+CD  0x0001   32768   45
Et46 Bundled | 8000,00-1c-73-09-a0-f3    46 ALGs+CD  0x0001   32768   46
Port Channel Port-Channel2:
Et47 Bundled | 8000,00-1c-73-01-02-1e    23 ALGs+CD  0x0002   32768   47
Et48 Bundled | 8000,00-1c-73-01-02-1e    24 ALGs+CD  0x0002   32768   48

             |                 Actor
Port Status  |   State         OperKey    PortPriority
-----------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled |   ALGs+CD        0x0001           32768
Et44 Bundled |   ALGs+CD        0x0001           32768
Et45 Bundled |   ALGs+CD        0x0001           32768
Et46 Bundled |   ALGs+CD        0x0001           32768
Port Channel Port-Channel2:
Et47 Bundled |   ALGs+CD        0x0002           32768
Et48 Bundled |   ALGs+CD        0x0002           32768 switch>
```

11.2.5.35 show lacp internal

The **show lacp internal** command displays the local LACP state for all specified channels. Local state data includes the state machines and LACP protocol information.

Command Mode

EXEC

Command Syntax

show lacp [**PORT_LIST**] internal [**PORT_LEVEL**] [**INFO_LEVEL**]

Parameters

- **PORT_LIST** Interface for which port information is displayed. Options include:
  - ***no parameter*** All configured port channels.
  - ***c_range*** Ports in specified channel list (number, number range, or list of numbers and ranges).
  - **interface** Ports on all interfaces.
  - **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **interface port-channel *p_num*** Port channel interface specified by ***p_num***.
- **PORT_LEVEL** Ports displayed, in terms of aggregation status. Options include:
  - ***no parameter*** Command lists data for ports bundled by LACP into an aggregate.
  - **all-ports** Command lists data for all ports, including LACP candidates that are not bundled.
- **INFO_LEVEL** Amount of information that is displayed. Options include:

- ***no parameter*** Displays same information as **brief** option.
- **brief** Displays LACP configuration data, including sys-id, actor, priorities, and keys.
- **detailed** Includes **brief** option information plus state machine data.

Note: PORT_LEVEL and **INFO_LEVEL** parameters can be placed in any order.

Example

This command displays internal data for all configured port channels.

```
switch> show lacp internal

LACP System-identifier: 8000,00-1c-73-04-36-d7
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
             |Partner                              Actor
Port Status  | Sys-id                 Port#  State    OperKey  PortPriority
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled | 8000,00-1c-73-09-a0-f3    43  ALGs+CD   0x0001   32768
Et44 Bundled | 8000,00-1c-73-09-a0-f3    44  ALGs+CD   0x0001   32768
Et45 Bundled | 8000,00-1c-73-09-a0-f3    45  ALGs+CD   0x0001   32768
Et46 Bundled | 8000,00-1c-73-09-a0-f3    46  ALGs+CD   0x0001   32768
```

11.2.5.36 show lacp peer

The **`show lacp peer`** command displays the LACP protocol state of the remote neighbor for all specified port channels.

Command Mode

EXEC

Command Syntax

`show lacp` [**PORT_LIST**] peer [**PORT_LEVEL**] [**INFO_LEVEL**]

Note: PORT_LEVEL and **INFO_LEVEL** parameters can be placed in any order.

Parameters

- **PORT_LIST** Interface for which port information is displayed. Options include:
  - ***no parameter*** Displays information for all configured port channels.
  - ***c_range*** Ports in specified channel list (number, number range, or list of numbers and ranges).
  - **interface** Ports on all interfaces.
  - **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **interface port-channel *p_num*** Port channel interface specified by ***p_num***.
- **PORT_LEVEL** Ports displayed, in terms of aggregation status. Options include:
  - ***no parameter*** Command lists data for ports bundled by LACP into an aggregate.
  - **all-ports** Command lists data for all ports, including LACP candidates that are not bundled.
- **INFO_LEVEL** Amount of information that is displayed. Options include:
  - ***no parameter*** Displays same information as **brief** option.
  - **brief** Displays LACP configuration data, including sys-id, actor, priorities, and keys.
  - **detailed** Includes **brief** option information plus state machine data.

Example

This command displays the LACP protocol state of the remote neighbor for all port channels.

```
switch> show lacp peer

State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
                |                          Partner
Port    Status  | Sys-id                  Port#   State     OperKey  PortPri
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et1     Bundled | 8000,00-1c-73-00-13-19      1   ALGs+CD    0x0001    32768
Et2     Bundled | 8000,00-1c-73-00-13-19      2   ALGs+CD    0x0001    32768
Port Channel Port-Channel2:
Et23    Bundled | 8000,00-1c-73-04-36-d7     47   ALGs+CD    0x0002    32768
Et24    Bundled | 8000,00-1c-73-04-36-d7     48   ALGs+CD    0x0002    32768
Port Channel Port-Channel4*:
Et3     Bundled | 8000,00-1c-73-0b-a8-0e     45   ALGs+CD    0x0001    32768
Et4     Bundled | 8000,00-1c-73-0b-a8-0e     46   ALGs+CD    0x0001    32768
Port Channel Port-Channel5*:
Et19    Bundled | 8000,00-1c-73-0c-30-09     49   ALGs+CD    0x0005    32768
Et20    Bundled | 8000,00-1c-73-0c-30-09     50   ALGs+CD    0x0005    32768
Port Channel Port-Channel6*:
Et6     Bundled | 8000,00-1c-73-01-07-b9     49   ALGs+CD    0x0001    32768
Port Channel Port-Channel7*:
Et5     Bundled | 8000,00-1c-73-0f-6b-22     51   ALGs+CD    0x0001    32768
Port Channel Port-Channel8*:
Et10    Bundled | 8000,00-1c-73-10-40-fa     51   ALGs+CD    0x0001    32768

* - Only local interfaces for MLAGs are displayed. Connect to the peer to
    see the state for peer interfaces.
switch>
```

11.2.5.37 show lacp sys-id

The **show lacp sys-id** command displays the System Identifier the switch uses when negotiating remote LACP implementations.

Command Mode

EXEC

Command Syntax

show lacp sys-id [**INFO_LEVEL**]

Parameters

INFO_LEVEL Amount of information that is displayed. Options include:

- ***no parameter*** Displays system identifier.
- **brief** Displays system identifier.
- **detailed** Displays system identifier and system priority, including the MAC address.

Examples

- This command displays the system identifier.

```
switch> show lacp sys-id brief
8000,00-1c-73-04-36-d7
```

- This command displays the system identifier and system priority.

```
switch> show lacp sys-id detailed
```

```
System Identifier used by LACP:
System priority: 32768  Switch MAC Address: 00:1c:73:04:36:d7
  802.11.43 representation: 8000,00-1c-73-04-36-d7
```

11.2.5.38 show load-balance profile

The **`show load-balance profile`** command displays the contents of the specified load balance profiles. Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among component ECMP routes.

Command Mode

EXEC

Command Syntax

`show load-balance profile` [**PROFILES**]

Parameters

PROFILES Load balance profiles for which command displays contents. Options include:

- ***no parameter*** Displays all load balance profiles.
- ***profile_name*** Displays specified profile.

Related Commands

- load-balance policies places the switch in ***load-balance-policies*** configuration mode.
- ingress load-balance profile applies a load-balance profile to an Ethernet or port channel interface.

Example

This command displays the contents of the ***LB-1*** load balance profile.

```
switch> show load-balance profile LB-1

---------- LB-1 ----------

Source MAC address hashing               ON
Destination MAC address hashing          ON
Ethernet type hashing                    ON
VLAN ID hashing                          ON
IP protocol field hashing                ON
DSCP field hashing is                    ON
Symmetric hashing for non-IP packets     OFF
Symmetric hashing for IP packets         OFF
Random distribution for port-channel     ON
Random distribution for ecmp             ON

Profile LB-1 is applied on the following
    Port-Channel100

---------- myGlobalProfile (global) ----------
L3 hashing is ON
Symmetric hashing is OFF
Hashing mode is flow-based
Hash polynomial is 3
Hash seed is 0
Profile myGlobalProfile (global) is applied on the following
Linecard3
Linecard4
Linecard5
```

```
Linecard6 switch>
```

11.2.5.39 show port-channel

The **show port-channel** command displays information about members the specified port channels.

Command Mode

EXEC

Command Syntax

show port-channel [**MEMBERS**] [**PORT_LIST**] [**INFO_LEVEL**]

Parameters

- **MEMBERS** List of port channels for which information is displayed. Options include:
  - ***no parameter*** All configured port channels.
  - ***p_range*** Ports in specified channel list (number, number range, or list of numbers and ranges).
- **PORT_LEVEL** Ports displayed, in terms of aggregation status. Options include:
  - ***no parameter*** Displays information on ports that are active members of the LAG.
  - **active-ports** Displays information on ports that are active members of the LAG.
  - **all-ports** Displays information on all ports (active or inactive) configured for LAG.
- **INFO_LEVEL** Amount of information that is displayed. Options include:
  - ***no parameter*** Displays information at the brief level.
  - **brief** Displays information at the brief level.
  - **detailed** Displays information at the detail level.

Display Values

- **Port Channel** Type and name of the port channel.
- **Time became active** Time when the port channel came up.
- **Protocol** Protocol operating on the port channel.
- **Mode** Status of the Ethernet interface on the port. The status value is Active or Inactive.
- **No active ports** Number of active ports on the port channel.
- **Configured but inactive ports** Ports configured but that are not actively up.
- **Reason unconfigured** Reason why the port is not part of the LAG.

Guidelines

You can configure a port channel to contain many ports, but only a subset may be active at a time. All active ports in a port channel must be compatible. Compatibility includes many factors and is platform specific. For example, compatibility may require identical operating parameters such as speed and Maximum Transmission Unit (MTU). Compatibility may only be possible between specific ports because of the internal organization of the switch.

Examples

- This command displays output from the **show port-channel** command.

```
switch> show port-channel 3
Port Channel Port-Channel3:
  Active Ports:
       Port               Time became active       Protocol    Mode
     ------------------------------------------------------------------------
       Ethernet3          15:33:41                 LACP        Active
       PeerEthernet3      15:33:41                 LACP        Active
```

- This command displays output from the **show port-channel active-ports** command.

```
switch> show port-channel active-ports
Port Channel Port-Channel3:
  No Active Ports
Port Channel Port-Channel11:
  No Active Ports switch>
```

- This command displays output from the **show port-channel all-ports** command.

```
switch> show port-channel all-ports
Port Channel Port-Channel3:
  No Active Ports
  Configured, but inactive ports:
      Port            Time became inactive    Reason unconfigured
    ---------------------------------------------------------------------------
      Ethernet3       Always                  not compatible with aggregate

Port Channel Port-Channel11:
  No Active Ports
  Configured, but inactive ports:
      Port            Time became inactive    Reason unconfigured
    ---------------------------------------------------------------------------
      Ethernet25      Always                  not compatible with aggregate
      Ethernet26      Always                  not compatible with aggregate
```

11.2.5.40 show port-channel dense

The **show port-channel dense** command displays the port-channels on the switch and lists their component interfaces, LACP status, and set flags.

Command Mode

EXEC

Command Syntax

show port-channel dense

Example

This command displays **show port-channel dense** output:

```
switch> show port-channel dense

    Flags
--------------------------------------------------------------------
a - LACP Active        p - LACP Passive
U - In Use             D - Down
+ - In-Sync            - - Out-of-Sync      i - incompatible with agg
P - bundled in Po      s - suspended        G - Aggregable
I - Individual         S - ShortTimeout     w - wait for agg

Number of channels in use: 2
Number of aggregators:2

   Port-Channel       Protocol    Ports
------------------------------------------------------
   Po1(U)             LACP(a)      Et47(PG+) Et48(PG+)
   Po2(U)             LACP(a)      Et39(PG+) Et40(PG+)
```

11.2.5.41 show port-channel limits

The **show port-channel limits** command displays groups of ports that are compatible and may be joined into port channels. Each group of compatible ports is called a LAG group. For each LAG group, the command also displays **Max interfaces** and **Max ports per interface**.

- **Max interfaces** defines the maximum number of active port channels that may be formed out of these ports.
- **Max ports per interface** defines the maximum number of active ports allowed in a port channel from the compatibility group.

All active ports in a port channel must be compatible. Compatibility comprises many factors and is specific to a given platform. For example, compatibility may require identical operating parameters such as speed and/or ZMaximum Transmission Unit (MTU). Compatibility may only be possible between specific ports because of internal organization of the switch.

Command Mode

EXEC

Command Syntax

show port-channel limits

Example

This command displays **show port-channel list** output:

```
switch> show port-channel limits

LAG Group: focalpoint
------------------------------------------------------------------------
  Max port-channels per group: 24, Max ports per port-channel: 16
  24 compatible ports: Ethernet1  Ethernet2  Ethernet3  Ethernet4
                       Ethernet5  Ethernet6  Ethernet7  Ethernet8
                       Ethernet9  Ethernet10 Ethernet11 Ethernet12
                       Ethernet13 Ethernet14 Ethernet15 Ethernet16
                       Ethernet17 Ethernet18 Ethernet19 Ethernet20
                       Ethernet21 Ethernet22 Ethernet23 Ethernet24
------------------------------------------------------------------------
```

11.2.5.42 show port-channel load-balance fields

The **show port-channel load-balance fields** command displays the fields that the hashing algorithm uses to distribute traffic across the interfaces that comprise the port channels.

Command Mode

EXEC

Command Syntax

show port-channel load-balance **HARDWARE** fields

Parameters

HARDWARE ASIC switching device. Selection options depend on the switch model and include:

- **arad**
- **fm6000**
- **petraA**
- **trident**

Example

This command displays the hashing fields used for balancing port channel traffic.

```
switch> show port-channel load-balance fm6000 fields

Source MAC address hashing for non-IP packets is ON
```

```
Destination MAC address hashing for non-IP packets is ON
Ethernet type hashing for non-IP packets is ON
VLAN ID hashing for non-IP packets is ON
VLAN priority hashing for non-IP packets is ON
Source MAC address hashing for IP packets is ON
Destination MAC address hashing for IP packets is ON
Ethernet type hashing for IP packets is ON
VLAN ID hashing for IP packets is ON
VLAN priority hashing for IP packets is ON
IP source address hashing is ON
IP destination address hashing is ON
IP protocol field hashing is ON
TCP/UDP source port hashing is ON
TCP/UDP destination port hashing is ON switch>
```

11.2.5.43 show port-channel load-balance

The **`show port-channel load-balance`** command displays the traffic distribution between the member ports of the specified port channels. The command displays distribution for unicast, multicast, and broadcast streams.

The distribution values displayed are based on the total interface counters which start from ***0*** at boot time or when the counters are cleared. For more current traffic distribution values, clear the interface counters of the member interfaces using the clear counters command.

Command Mode

EXEC

Command Syntax

`show port-channel load-balance` [**MEMBERS**]

Parameters

MEMBERS list of port channels for which information is displayed. Options include:

- ***no parameter*** all configured port channels.
- ***c_range*** ports in specified channel list (number, number range, or list of numbers and ranges).

Example

This command displays traffic distribution for all configured port channels.

```
switch> show port-channel load-balance
ChanId      Port Rx-Ucst Tx-Ucst Rx-Mcst Tx-Mcst Rx-Bcst Tx-Bcst
------ --------- ------- ------- ------- ------- ------- -------
     8      Et10 100.00% 100.00% 100.00% 100.00%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
     1       Et1  13.97%  42.37%  47.71%  30.94%   0.43%  99.84%
     1       Et2  86.03%  57.63%  52.29%  69.06%  99.57%   0.16%
------ --------- ------- ------- ------- ------- ------- -------
     2      Et23  48.27%  50.71%  26.79%  73.22%   0.00% 100.00%
     2      Et24  51.73%  49.29%  73.21%  26.78%   0.00%   0.00%
------ --------- ------- ------- ------- ------- ------- -------
     4       Et3  55.97%  63.29%  51.32%  73.49%   0.00%   0.00%
     4       Et4  44.03%  36.71%  48.68%  26.51%   0.00%   0.00%
------ --------- ------- ------- ------- ------- ------- -------
     5      Et19  39.64%  37.71%  50.00%  90.71%   0.00%   0.00%
     5      Et20  60.36%  62.29%  50.00%   9.29%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
     6       Et6 100.00% 100.00% 100.00% 100.00%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
```

```
     7       Et5 100.00%   0.00% 100.00% 100.00%   0.00%   0.00%
switch>
```

11.3 Multi-Chassis Link Aggregation

Arista switches support Multi-Chassis Link Aggregation (MLAG) to logically aggregate ports across two switches. For example, two 10-gigabit Ethernet ports, one each from two MLAG configured switches, can connect to two 10-gigabit ports on a host, switch, or network device to create a link that appears as a single 20-gigabit port. MLAG-configured ports provide Layer 2 multipathing, increased bandwidth, higher availability, and other improvements on traditional active-passive or Spanning Tree governed infrastructures.

The Multi-Chassis Link Aggregation section contains these topics:

- MLAG Introduction
- MLAG Conceptual Overview
- MLAG Maintenance
- MLAG Dual Primary Detection and Release
- Configuring MLAG
- EVPN - MLAG Single-homed Hosts
- MLAG Implementation Example
- MLAG Commands

11.3.1 MLAG Introduction

High availability data center topologies typically provide redundancy protection at the expense of over-subscription by connecting Top-Of-Rack (TOR) switches and servers to dual aggregation switches. In these topologies, Spanning Tree Protocol prevents network loops by blocking half of the links to the aggregation switches. This reduces the available bandwidth by 50%.

Deploying MLAG removes over-subscription by configuring an MLAG link between two aggregation switches to create a single logical switching instance that utilizes all connections to the switches. Interfaces on both devices participate in a distributed port channel, enabling all active paths to carry data traffic while maintaining the integrity of the Spanning Tree topology.

MLAG provides these benefits:

- Aggregates multiple Ethernet ports across two switches.
- Provides higher bandwidth links as network traffic increases.
- Utilizes bandwidth more efficiently with fewer links blocked by STP.
- Connects to other switches and servers by static LAG or LACP without other proprietary protocols.
- Supports normal STP operation to prevent loops.
- Supports active-active Layer-2 redundancy.

Note: For information on enabling Precision Timing Protocol (PTP) on an MLAG interface, see the Timing Protocols chapter.

Note: The global STP configuration is derived from the primary peer device while the secondary device parameters are ignored. When STP is disabled on the primary device, the secondary device will not contain any STP configuration information from the primary device. As a result, the secondary device will not be able to decide on the port roles or states, and will remain in the default state which is the discarding state. This is an expected behavior.

Note: It is highly recommended that both MLAG peer switches are identical platforms and run identical EOS images. Running different images/platform may result in a failure to form an association with the MLAG peer or see discrepancy in behavior.

11.3.2 MLAG Conceptual Overview

11.3.2.1 MLAG Operation Process

A Multi-chassis Link Aggregation Group (MLAG) is a pair of links that terminate on two cooperating switches and appear as an ordinary Link Aggregation Group (LAG). The cooperating switches are MLAG peer switches and communicate through an interface called a peer link. While the peer link's primary purpose is exchanging MLAG control information between peer switches, it also carries data traffic from devices that are attached to only one MLAG peer and have no alternative path. An MLAG domain consists of the peer switches and the control links that connect the switches.

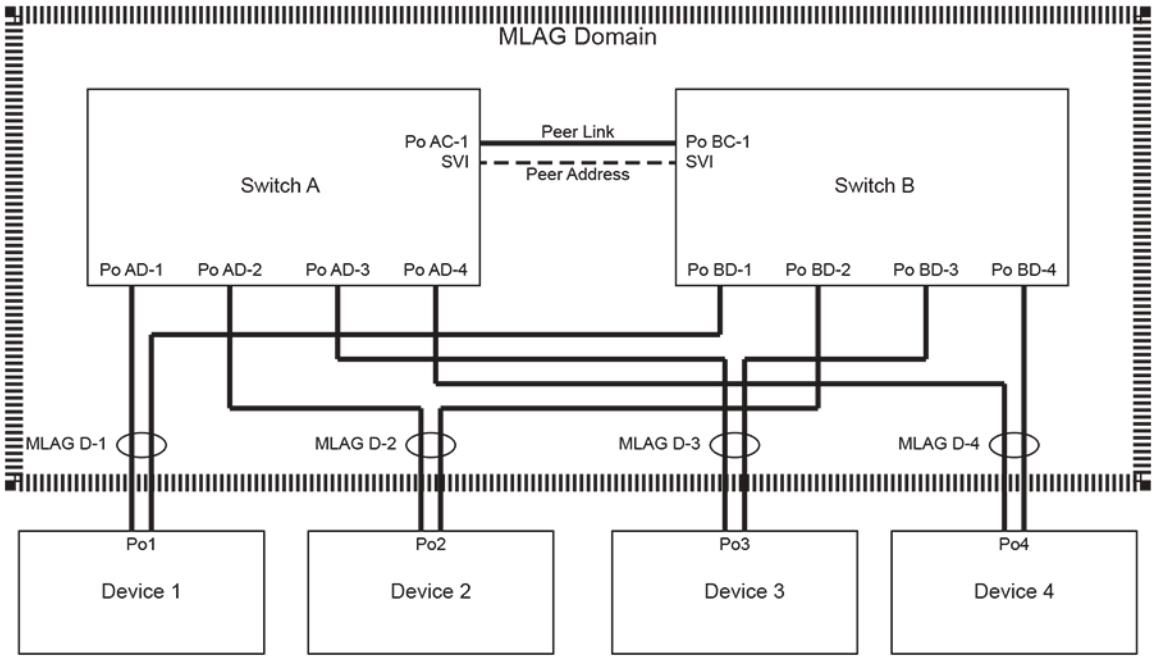
In the figure below, Switch A and Switch B are peer switches in the MLAG domain and connect to each other through the peer link. Each peer switch uses the peer address to form and maintain the peer link.

The MLAG domain ID is a text string configured in each peer switch. MLAG switches use this string to identify their peers. The MLAG System ID (MSI) is the MLAG domain's MAC address. The MSI is automatically derived when the MLAG forms and does not match the bridge MAC address of either peer. Each peer uses the MSI in STP and LACP PDUs.

The topology shown below contains four MLAGs: one MLAG connects each device to the MLAG domain. Each peer switch connects to the four servers through MLAG link interfaces.

In a conventional topology, with dually-attaching devices to multiple switches for redundancy, Spanning Tree Protocol (STP) blocks half of the switch-device links. In the MLAG topology, STP does not block any portion because it views the MLAG Domain as a single switch and each MLAG as a single port. The MLAG protocol facilitates the balancing of device traffic between the peer switches.

Figure 11-1: MLAG Domain Topology



When MLAG is disabled, peer switches revert to their independent state. MLAG is disabled by any of the following:

- MLAG configuration changes.
- The TCP connection breaks.
- The peer-link or local-interface goes down.
- A switch does not receive a response to a keep alive message from its peer within a specified period.

11.3.2.2 MLAG Interoperability with Other Features

The following sections describe MLAG interaction with other switch features.

11.3.2.2.1 VLANs

VLAN parameters must be configured identically on each peer for the LAGs comprising the peer link and MLAGs. These parameters include the switchport access VLAN, switchport mode, trunk-allowed VLANs, the trunk native VLAN, and switchport trunk groups.

Configuration discrepancies may result in traffic loss in certain failure scenarios. Port-specific bridging configuration originates on the switch where the port is physically located.

11.3.2.2.2 LACP

Link Aggregation Control Protocol (LACP) should be used on all MLAG interfaces, including the peer-link. LACP control packets reference the MLAG system ID.

11.3.2.2.3 Static MAC Addresses

A static MAC address configured on an MLAG interface is automatically configured on the peer's corresponding interface. Configuring static MAC addresses on both peers prevents undesired flooding if an MLAG peer relationship fails.

If the MLAG peering relationship is disabled, the static MAC previously learned from peer is removed.

11.3.2.2.4 Spanning Tree Protocol (STP)

When implementing MLAG in a spanning tree network, spanning tree must be configured globally and on port-channels configured with an MLAG ID. Port specific spanning tree configuration comes from the switch where the port physically resides. This includes spanning-tree PortFast BPDU Guard and BPDU filter.

11.3.2.2.5 Port Mirroring

A port channel which is a member of an MLAG *must not* be used as the destination port for a port mirroring (port monitoring) session.

11.3.2.3 IPv6 Flow Label Hashing

Arista switches use the hashing algorithm to load-balance traffic among LAG members and Layer 3 ECMP (equal cost multipath) paths. For IP and IPv6 traffic, the hashing algorithm includes (if so configured for LAG) the IP packet fields such as source and destination IP addresses as well as source and destination ports for UDP and TCP traffic.

To improve traffic distribution for IPv6 traffic, IPv6 Flow Label field has been added to the hashing algorithm for both LAG and ECMP.

Note: IPv6 Flow Label is included in the LAG hashing algorithm only when the MAC header hashing is not enabled for IPv6 traffic.

11.3.3 MLAG Maintenance

These sections describe tasks required for MLAG to operate on the switch:

- Ensuring Control Plane ACL Compatibility
- MLAG Availability through a Single Functional Peer
- Upgrading MLAG Peers

11.3.3.1 Ensuring Control Plane ACL Compatibility

The control plane Access Control List (ACL) on any interface participating in the MLAG must be configured to allow only the peer link neighbor to generate MLAG control traffic. The required rules are included in the default control plane ACL for Ethernet ports.

Any custom control plane ACL applied to a participating port must include these three rules:

```
permit tcp any any eq mlag ttl eq 255
permit udp any any eq mlag ttl eq 255
permit tcp any eq mlag any ttl eq 255
```

MLAG peers that function as routers must each have routing enabled.

11.3.3.2 MLAG Availability through a Single Functional Peer

MLAG high availability advantages are fully realized when all devices that connect to one MLAG switch also connect to the peer switch. A switch can continue supporting MLAG when its peer is offline if the STP agent is restartable. When one peer is offline, data traffic flows from the devices through the MLAG component link that connects to the functioning switch. When a switch is offline, its interfaces and ports do not appear in show mlag and **`show spanning tree protocol`** commands of the functioning peer.

To view the restartability status of the STP agent, use the **detail** option of the **`show spanning-tree instance`** command:

```
switch-1# show spanning-tree instance detail | grep agent
Stp agent restartable                    :            True
```

STP agent restartability requires consistent configuration between the peers of STP, LACP, MLAG, and switchport parameters. Events triggering an STP state machine change may also briefly prevent the STP agent from being restartable.

11.3.3.2.1 Reload Delay

If an MLAG peer reboots, all ports except those in the peer-link port-channel remain in ***errdisabled*** state for a specified time, called the reload-delay period. This period allows all topology states to stabilize before the switch begins forwarding traffic. Each Arista switch defaults to the recommended reload-delay value, which varies by switch platform:

- **Fixed configuration switches**: 300 seconds
- **Trident II modular switches**: 1200 seconds
  - 7304
  - 7308
  - 7316
  - 7300X series
- **Sand platform fixed configuration switches**: 600 seconds
  - 7280 series (except 7280CR2 and 7280SR2)
  - 7020 series
- **Sand platform modular switches**: 1800 seconds
  - 7504
  - 7508
  - 7500E series
  - 7548S
- **Sand Jericho+ fixed configuration switches**: 900 seconds
  - 7280CR2 series

- 7280SR2 series

In those cases where network topology requires additional time to stabilize or where a shorter delay can be tolerated, the reload-delay period can be configured using the reload-delay mlag command.

Severing the physical connection (cable) that establishes the peer-link between MLAG peers may result in a ***split brain*** state where each peer independently enters spanning tree state to prevent topology loops. Sessions established through one interface of a dual attached device may fail if its path is disrupted by the STP reconvergence, possibly resulting in temporarily lost connectivity. Sessions can be reestablished if permitted by the resulting topology.

11.3.3.3 Upgrading MLAG Peers

MLAG ISSU (In-Service Software Upgrade) upgrades EOS software on one MLAG peer with minimal traffic disruptions on active MLAG interfaces and without changing the network topology.

11.3.3.3.1 Verifying Configuration Compatibility

A seamless EOS upgrade on an MLAG peer requires that the following features are configured consistently on each switch:

- VLANs.
- Switchport configuration on port channel interfaces that are configured with an MLAG ID.
- STP configuration (global).

11.3.3.3.2 Version Compatibility

A switch running MLAG can be upgraded without disrupting MLAG traffic when the upgrade EOS version is compatible with the version on the peer switch. Refer to the Release Notes for a list of compatible EOS versions.

11.3.3.3.3 Reload Warning Conditions

Entering an EOS reload command while MLAG is active generates warning messages if conditions that can result in packet loss during the upgrade are present. All warnings should be resolved before confirming the reload request. The following table displays the reload conditions and a common resolution method for each condition.

Table 64: Reload Warning Resolutions

Reload Condition	Resolution Method
Compatibility check	Refer to the Release Notes to verify that the new version is compatible with the currently installed version.
Active-partial MLAG warning	Bring up the remote port-channel. If the MLAG is not actively used, then this warning can be ignored.
STP is not restartable	Wait for STP to be restartable: typically 30 seconds, up to 120 seconds for a newly started STP agent. Refer to MLAG Availability through a Single Functional Peer for information on checking restartability.
Reload delay too low	Configure a reload delay value greater than or equal to the default. Recommend delay is 300 seconds for TOR switches, 900 seconds for modular switches, and 600 seconds for Sand platform fixed configuration switches (7020 and 7280 series (except 7280CR2 and 7280SR2)).
Peer has error-disabled interfaces	Wait for reload-delay to expire on the peer.

Example

The following **reload** command generates MLAG warning conditions that should be addressed before confirming the ***proceed with reload*** prompt.

```
switch(config)# reload
If you are performing an upgrade, and the Release Notes for the
new version of EOS indicate that MLAG is not backwards-compatible
with the currently installed version (4.9.2), the upgrade will
result in packet loss.

The following MLAGs are not in Active mode. Traffic to or from these ports will
be lost during the upgrade process.
                                                                  local/remote
  mlag      desc               state      local      remote          status
--------- --------- ------------------- ---------- ---------- ------------
    14               active-partial       Po14       Po14         up/down
    15               active-partial       Po15       Po15         up/down

Stp is not restartable. Topology changes will occur during the upgrade process.

The configured reload delay of 100 seconds is below the default
value of 300 seconds. A longer reload delay allows more time to
rollback an unsuccessful upgrade due to incompatibility.

The other MLAG peer has errdisabled interfaces. Traffic loss will occur during
the upgrade process.

Proceed with reload? [confirm]
```

11.3.3.3.4 Performing an MLAG ISSU Upgrade

The following procedure performs an MLAG ISSU upgrade:

1. Verify configuration consistency on each peer (Verifying Configuration Compatibility).
2. Verify version compatibility between the new and existing images (Version Compatibility).
3. Configure reload-delay mlag (MLAG Availability through a Single Functional Peer) if needed. Recommended delay period varies by switch type, and each switch defaults to its recommended delay period.
4. Install the new image onto one of the peers:
   a. Upload the new image to the switch.
   b. Set the boot path to the new image.
   c. Enter the **reload** command.
5. Resolve all reload warnings.
6. Confirm the reload.
7. Wait for MLAG peers to renegotiate to the active state and reload-delay expiry on rebooted peer; until reload-delay period has expired, ports on the rebooted peer (except the peer-link) will be in ***errdisabled*** state with err-disabled reason being ***mlag-issu***.

   Avoid configuration changes on both peers until after this step.
8. Repeat the upgrade process for the other peer.

   When upgrading modular switches with dual supervisors, upgrade the standby supervisors first, then upgrade the active supervisors.

11.3.4 MLAG Dual Primary Detection and Release

When the MLAG peer-link goes down, the secondary peer assumes the primary peer is down or dead, and takes over the primary role. It is possible that when peer-link is down, the primary is not actually down or dead, and both MLAG peers think they are the primary. This is called **dual-primary** condition or state. In this state, each peer runs Layer-2 protocols such as Spanning Tree Protocol independently. Depending on the topology, this can cause loops in the network. This can also impact the IGMP snooping feature.

The Dual Primary Detection feature uses the management interface as an out-of-band connection between the two peers along with the peer-link. Once configured, the MLAG peers sends UDP heartbeat packets on the management interface. When the peer-link goes down and the heartbeat is still active on out-of-band connection, both MLAG peers detects **dual-primary** condition. The switch can take optional action to shutdown all links except the peer-link on the original secondary MLAG peer.

Beginning with the ***EOS Release 4.23.1F***, there is support for dual primary recovery delay.

Recovery delay is an improvement to the current dual-primary detection feature. Use only when MLAG pairs recover from dual-primary detection with configured errdisable all-interfaces. When recovery delay is configured, the old secondary MLAG peer expects to un-errdisable MLAG interfaces later than non-MLAG to allow more time for L3 interfaces to converge.

Beginning with the ***EOS Release 4.25.0F***, when dual-primary-detection is enabled, there will be heartbeats transferred on both the peer-link and the management link at the same time. This improves dual-primary-detection because both the management link and the peer-link need to fail before the detection fails.

11.3.4.1 Supported EOS Versions

MLAG dual primary detection is supported beginning with ***EOS Release 4.20.0F*** and onwards.

11.3.4.2 Limitations

The following limitaions apply to the MLAG Dual Primary Detection feature.

1. UDP port 4432 is used to exchange control information between the peers, and must not be blocked on the management interfaces on both of the MLAG peers.
2. At least one MLAG port-channel must be configured on the MLAG peers to perform dual primary detection and the corresponding errdisable action.
3. Errdisable action brings down all physical Ethernet interfaces excluding the MLAG peer-link and management port. However, Layer-2 port-channels go down and VLAN SVIs may go down (i.e. autostate taking down VLAN SVIs).
4. When the errdisable action is configured with the non-MLAG interfaces recovery delay the errdisable action must always be less than the MLAG interfaces recovery delay. Otherwise, MLAG interfaces will come up sooner and cause northbound traffic loss.
5. Do not configure this feature if there can be consecutive peer-link flaps. There is an issue that if there are consecutive peer-link flaps during which MLAG peers also happen to switch roles, you can have unexpired recovery timers on the new primary and the secondary can not go dual primary detected until the MLAG recovery timer expires. This can open a small window for L3 traffic blackholing on the primary depending on the delta between recovery delay for MLAG and non-MLAG interfaces.
6. Changing the recovery delay configurations during active recovery delay may cause interfaces to stay errdisabled.

11.3.4.3 MLAG Dual Primary Detection and Release Configuration

To enable MLAG dual primary detection feature, you must configure the following commands on both MLAG peers in the ***MLAG*** configuration mode.

```
switch(config-mlag)# peer-address heartbeat Peer-IP [vrf VRF-NAME ]
```

Parameters:

- ***Peer-IP*** The Management IP address of the MLAG peer reachable in the targeted VRF.
- ***VRF-NAME*** The MLAG peer reachable in VRF (or default VRF if no VRF is configured).

The **`peer-address heartbeat`** command causes the MLAG agent to start using ***Peer-IP*** address in the given VRF for UDP-based heartbeat control messages.

The **dual-primary detection delay** command configures dual primary detection delay with an optional action to errdisable all interfaces on secondary MLAG peer after detecting the dual primary condition.

```
switch(config-mlag)# dual-primary detection delay SECONDS [action errdisable
 all-interfaces]
```

Parameters:

- **delay** Specifies the number of seconds to wait after MLAG failover to listen for MLAG heartbeats to determine dual primary condition.
- **action errdisable all-interfaces** Errdisables all local physical interfaces on the secondary MLAG peer after detecting the dual primary condition.

Note: This feature must be configured on both MLAG peers.

Use both the **peer-address heartbeat** command and the **dual-primary detection** command on each peer to disable the Dual Primary Detection feature.

The **peer-address heartbeat** command triggers MLAG to revert using the peer-link for heartbeat control messages instead of the management IP address of the MLAG peer.

```
switch(config-mlag)# (no|default) peer-address heartbeat
```

The **dual-primary detection** command removes the detection delay time as well as detection action.

Examples

- This example specifies the MLAG Peer's Management IP address ***172.30.118.190*** in default VRF for heartbeats.

  ```
  switch(config)# mlag
  switch(config-mlag)# peer-address heartbeat 172.30.118.190
  ```

- In this example, the command errdisables all physical interfaces on secondary MLAG peer when dual primary condition is detected.

  ```
  switch(config-mlag)# dual-primary detection delay 5 action errdisable all-
  interfaces
  ```

  Both MLAG peers must have equivalent configurations.

The following commands remove the Dual Primary Detection feature. You must unconfigure both MLAG peers.

```
switch(config-mlag)# no peer-address heartbeat
switch(config-mlag)# no dual-primary detection
```

The **dual-primary recovery delay** command configures the dual-primary detection recovery-delay for MOLG interfaces and non-MPLG interfaces. Negating the configurations or configuring default values makes both recovery delay values reset back to 0. The non-MLAG delay must always be less than the MLAG delay so that you have more time for L3 convergence before you enable the MLAG interfaces. A suggested value for MLAG is 60 seconds, and non-MLAG is 0 seconds. These values can be adjusted depending on the network scale.

```
switch(config-mlag)# dual-primary recovery delay mlag <0-1000> non-
mlag <0-1000>
switch(config-mlag)# [no|default] dual-primary recovery delay
```

11.3.4.3.1 ISSU Implications

When upgrading from an EOS release that does not support the Dual Primary Detection feature, you must configure this feature only after both MLAG peers are upgraded to the release that has support. When downgrading to a release without Dual Primary Detection support, the configuration must be removed from both the peers before the downgrade process is initiated on either peer. Otherwise, MLAG fails configuration sanity.

11.3.4.4 Viewing the MLAG Dual Primary Detection and Release Status

Use the **show running-config** command to display the MLAG dual primary detection configuration.

Example

```
switch# show running-config
...
mlag configuration
 domain-id test_domain_1
 local-interface Vlan4094
 peer-address 10.0.0.2
 peer-address heartbeat 172.30.118.190
 primary-priority 11
 peer-link Port-Channel1
dual-primary detection delay 5 action errdisable all-interfaces
 ...
```

Use the **show mlag detail** command to display MLAG dual primary interface errdisabled status, recovery delay status, and the heartbeat counts sent and received. You can infer that the status of recovery delay is active if the dual-primary detection is Configured (dual primary resolved) and dual-primary interface errdisabled is True.

Example

```
switch(config-mlag)# show mlag detail
 MLAG Configuration:
 domain-id : test_domain_1
 local-interface : Vlan4094
 peer-address : 10.0.0.1
 peer-link : Port-Channel1
 hb-peer-address : 172.30.117.28
 peer-config : consistent

MLAG Status:
 state : Active
 negotiation status : Connecting
 peer-link status : Down
 local-int status : Up
 system-id : 46:4c:a8:c6:42:dd
 dual-primary detection : Detected

MLAG Ports:
 Disabled : 0
 Configured : 0
 Inactive : 0
 Active-partial : 0
 Active-full : 0

MLAG Detailed Status:
 State : primary
 State changes : 5
 Last state change time : 1:06:19 ago
 Hardware ready : True
```

```
 Failover : True
 Last failover change time : 1:06:19 ago
 Secondary from failover : False
 primary-priority : 12
 Peer primary-priority : 11
 Peer MAC address : 44:4c:a8:c6:42:dd
 Peer MAC routing supported : True
 Reload delay : 0 seconds
 Non-MLAG reload delay : 0 seconds
 Peer ports errdisabled : False
 Lacp standby : False
 Configured heartbeat interval : 4000 ms
 Effective heartbeat interval : 4000 ms
 Heartbeat timeout : 60000 ms
 Last heartbeat timeout : never
 Heartbeat timeouts since reboot : 0
 UDP heartbeat alive : True
 Heartbeats sent/received : 3189/3189
 Peer monotonic clock offset : unknown
 Agent should be running : True
 P2p mount state changes : 4
 Fast MAC redirection enabled : False
 Dual-primary detection delay : 5
 Dual-primary action : errdisable-all
```

When the dual-primary action is errdisable and detects the **dual-primary** condition, all the ethernet interfaces on the secondary MLAG peer are errdisabled with a **mlagdualprimary** reason.

Example

```
switch# show interfaces status errdisabled
 Port Name Status Reason
 ------------ ---------------- ----------------- --------------------
 Et1 errdisabled mlagdualprimary
 Et2 errdisabled mlagdualprimary
 Et14 errdisabled mlagdualprimary
...
```

11.3.5 Configuring MLAG

These sections describe the basic MLAG configuration steps:

- Configuring the MLAG Peers
- Configuring MLAG Services

11.3.5.1 Configuring the MLAG Peers

Connecting two switches as MLAG peers requires the establishment of the peer link and an SVI that defines local and peer IP addresses on each switch.

The peer link is composed of a LAG between the switches. When all devices that connect to the MLAG domain are dually connected to the switches through an MLAG, a peer link of two Ethernet interfaces is sufficient to handle MLAG control data and provide N+1 redundancy. When the domain connects to devices through only one MLAG peer, the peer link may require additional Ethernet interfaces to manage data traffic.

Disruptions to peer link connectivity due to forwarding agent restarts may cause an extended MLAG outage. Forwarding agent restart event include some configuration changes, such as port speed change or UFT mode change). The following precautions can reduce the risk of losing peer-link connectivity:

- all switches: constructing peer-links from port-channels in preference to a single Ethernet interface.
- modular systems: peer-link port-channel members should span multiple line cards.

- multi-chip systems: peer-link port-channel member should span multiple chips.

Managing Switch Configuration Settings describes modular systems.

The steps that configure two switches as MLAG peers include:

- Configuring the Port Channels, VLAN Interfaces, and IP addresses
- Configuring Peer Parameters
- Configuring MLAG Peer Gateway

11.3.5.1.1 Configuring the Port Channels, VLAN Interfaces, and IP addresses

The peer link is a normal port channel. The local address is the SVI that maps to the peer link port channel. The port channel and SVI must be configured on each peer switch. The port channel should be an active LACP port. The local and peer addresses must be located on the same IP address subnet. Autostate should be disabled on the SVI configured as the local interface.

Examples

- These commands create an active mode LACP port channel interface from two Ethernet interfaces and configure it as part of a trunk group on each switch.

  The **`switchport mode trunk`** command permits all VLANs on the interface by default, so all VLANs are permitted on port channel 10 in the following example. The configuration of a trunk group for a VLAN restricts only that specific VLAN to the associated ports: VLAN 4094 is only permitted on port channel 10, and not on any other ports on the switch. It is important to remember that all VLANs must be permitted between the peers on the peer link for correct operation.

  Switch 1

```
switch1# config
switch1(config)# vlan 4094
switch1(config-vlan-4094)# trunk group m1peer
switch1# config
switch1(config)# interface ethernet 1-2
switch1(config-if-et1-2)# channel-group 10 mode active
switch1(config-if-et1-2)# interface port-channel 10
switch1(config-if-po10)# switchport mode trunk
switch1(config-if-po10)# switchport trunk group m1peer
switch1(config-if-po10)# exit
switch1(config)#
```

  Switch 2

```
switch2# config
switch2(config)# vlan 4094
switch2(config-vlan-4094)# trunk group m1peer
switch2(config-vlan-4094)# exit
switch2(config)# interface ethernet 1-2
switch2(config-if-et1-2)# channel-group 10 mode active
switch2(config-if-et1-2)# interface port-channel 10
switch2(config-if-po10)# switchport mode trunk
switch2(config-if-po10)# switchport trunk group m1peer
switch2(config-if-po10)# exit
switch2(config)#
```

- These commands create an SVI for the local interface and associate it to the trunk group assigned to the peer link port channel.

  The SVI creates a Layer 3 endpoint in the switch and enables MLAG processes to communicate via TCP. The IP address can be any unicast address that does not conflict with other SVIs. STP is disabled for

the peer link ***vlan 4094*** to prevent any potential STP disruption of inter peer communications. Recall that the VLAN has been restricted to port-channel 10 by the earlier trunk group configuration thus preventing potential Layer 2 loop conditions within ***vlan 4094***.

Switch 1

```
switch1# config
switch1(config)# interface vlan 4094
switch1(config-if-vl4094)# ip address 10.0.0.1/30
switch1(config-if-vl4094)# no autostate
switch1(config-if-vl4094)# exit
switch1(config)# no spanning-tree vlan-id 4094
switch1(config)#
```

Switch 2

```
switch2# config
switch2(config)# interface vlan 4094
switch2(config-if-vl4094)# ip address 10.0.0.2/30
switch2(config-if-vl4094)# no autostate
switch2(config-if-vl4094)# exit
switch2(config)# no spanning-tree vlan-id 4094
switch2(config)#
```

11.3.5.1.2 Configuring Peer Parameters

Peer connection parameters configure the connection between the MLAG peer switches. This section describes the following peer configuration parameters.

- MLAG Configuration Mode
- Local VLAN Interface
- Peer Address
- Peer Link
- Domain ID
- Heartbeat Interval and Timeout
- Reload Delay Period
- Shutdown

11.3.5.1.2.1MLAG Configuration Mode

Peer connection parameters are configured in MLAG-configuration mode. The mlag configuration (global configuration) command places the switch in MLAG configuration mode.

Example

This command places the switch in MLAG configuration mode.

```
switch(config)# mlag configuration
switch(config-mlag)#
```

11.3.5.1.2.2Local VLAN Interface

The local interface specifies the SVI upon which the switch sends MLAG control traffic. The local IP address is specified within the definition of the VLAN associated with the local interface. The Peer Address configures the control traffic destination on the peer switch.

The **`local-interface`** command specifies a VLAN interface as the peer link SVI.

Example

This command configures ***vlan 4094*** as the local interface.

```
switch(config-mlag)#local-interface vlan 4094
switch(config-mlag)#
```

11.3.5.1.2.3Peer Address

The peer address is the destination address on the peer switch for MLAG control traffic. If the peer IP address is unreachable, MLAG peering fails and both peer switches revert to their independent state.

The **peer-address** command specifies the peer address.

Example

This command configures a peer address of ***10.0.0.2***.

```
switch(config-mlag)# peer-address 10.0.0.2
switch(config-mlag)#
```

11.3.5.1.2.4Peer Link

An MLAG is formed by connecting two switches through an interface called a peer link. The peer link carries MLAG advertisements, keepalive messages, and data traffic between the switches. This information keeps the two switches working together as one. While interfaces comprising the peer links on each switch must be compatible, they need not use the same interface number. Ethernet and port-channel interfaces can be configured as peer links.

The **peer-link** command specifies the interface the switch uses to communicates MLAG control traffic.

Example

This command configures ***port-channel 10*** as the peer link.

```
switch(config-mlag)# peer-link port-channel 10
switch(config-mlag)#
```

11.3.5.1.2.5Domain ID

The MLAG domain ID is a unique identifier for an MLAG domain. The MLAG domain ID must be the identical on each switch to facilitate MLAG communication.

The domain-id command configures the MLAG domain ID.

Example

This command configures ***mlagDomain*** as the domain ID:

```
switch(config-mlag)# domain-id mlagDomain
switch(config-mlag)#
```

11.3.5.1.2.6Heartbeat Interval and Timeout

The heartbeat interval specifies the period between the transmission of successive keepalive messages. Each MLAG switch transmits keepalive messages and monitors message reception from its peer. The heartbeat timeout is reset when the switch receives a keepalive message. If the heartbeat timeout expires, the switch disables MLAG under the premise that the peer switch is not functioning.

The heartbeat-interval (MLAG) command configures the heartbeat interval between ***1*** and ***30*** seconds, with a default value of ***4*** seconds. The heartbeat timeout expiry is 30 seconds.

Note: On 7500 and 7500E Series Switches, Arista recommends setting the heartbeat interval to ***10*** seconds.

Example

This command configures the heartbeat interval as ***2500*** milliseconds (2.5 seconds).

```
switch(config-mlag)# heartbeat-interval 2500
switch(config-mlag)#
```

11.3.5.1.2.7Reload Delay Period

The reload delay period specifies the interval that non-peer links are disabled after an MLAG peer reboots. This interval allows non-peer links to learn multicast and OSPF states and synchronize ARP caches before the ports start handling traffic. Each Arista switch defaults to the recommended reload-delay value, which varies by switch platform.

- Fixed configuration switches: ***300*** seconds (five minutes).
- Trident II platform modular switches: ***1200*** seconds (twenty minutes).
- Sand platform fixed configuration switches (7020 and 7280 series (except 7280CR2 and 7280SR2)): ***600*** seconds (ten minutes).
- Sand platform modular switches: ***1800*** seconds (thirty minutes).

In those cases where network topology requires additional time to stabilize or where a shorter delay can be tolerated, the reload-delay period can be configured using the reload-delay mlag command.

Example

This command configures the reload delay interval as ***2.5*** minutes (***150*** seconds).

```
switch(config-mlag)# reload-delay 150
switch(config-mlag)#
```

11.3.5.1.2.8Shutdown

The shutdown (MLAG) command disables MLAG operations without disrupting the MLAG configuration. The **`no mlag configuration`** command (global configuration mode) disables MLAG and removes the MLAG configuration. The **`no shutdown`** command resumes MLAG activity.

Examples

- This command disables MLAG activity on the switch.

```
switch(config-mlag)# shutdown
switch(config-mlag)#
```

- This command resumes MLAG activity on the switch.

```
switch(config-mlag)# no shutdown
switch(config-mlag)#
```

11.3.5.1.3 Configuring MLAG Peer Gateway

In an MLAG setup, routing on a MLAG peer switch is possible using its own bridge system MAC, VARP MAC, or VRRP MAC. On a peer receiving an IP packet with destination MAC set to one of these MACs, a packet gets routed if its hardware has enough information to route the packet. Configuring sending traffic to a cached MAC involves routing the session table and MLAG peer traffic if packets are received with the MAC peer.

Examples

- This command enables the MLAG peer gateway.

```
switch(config)# ip virtual-router mac-address mlag-peer
switch1(config)#
```

- This command disables the MLAG peer gateway.

```
switch(config)# no ip virtual-router mac-address mlag-peer
switch1(config)#
```

11.3.5.1.4 Configuring Ingress Replication to LAGs

Hardware support for ingress replication to LAGs is enabled by default when the user configures ingress replication. When multicast traffic is sent over the LAG, the hardware uses its built-in algorithm, based on the L2/L3/L4 headers, to load balance traffic over ports in the LAG. When a port goes down in a LAG, the hardware quickly hashes the multicast traffic over the remaining ports in the LAG, resulting in fewer drops than software based LAG support.

Examples

- This command enables ingress replication.

```
switch(config)# platform sand multicast replication default ingress
switch(config)#
```

- This command configures the maximum members (within a range of ***1*** through ***64***) for ingress only replication in a multicast group.

```
switch(config)# platform sand multicast replication ingress maximum 32
switch(config)#
```

11.3.5.2 Configuring MLAG Services

An MLAG is a pair of links that originate on a network attached device and terminate on the two MLAG peer switches. The MLAG switches coordinate traffic to the device through a common mlag (port-channel interface configuration) command on the interfaces that connect to the device.

The MLAG ID differs from the MLAG domain ID. The MLAG domain ID is assigned globally per switch in MLAG configuration mode, and the same MLAG domain ID must be on both switches.

Port channels configured as an MLAG must have identical port channel numbers. Although the MLAG ID is a distinct parameter from the port channel number, best practices recommend assigning the MLAG ID to match the port channel number.

Note: Arista recommends configuring the downstream switch or router connected to the MLAG peers to negotiate a LAG using LACP rather than configuring static LAGs. (On Arista switches, LACP is enabled using the command **`channel-group mode active`**. See Port Channels and LACP .) Using LAGs negotiated by LACP can help avoid problems caused by miscabling because LACP PDUs include the system ID and LACP key, which identify the switch and the LAG it belongs to. If MLAG miscabling happens using a LACP LAG, some ports will be inactive in the LAG which prevents it from bridging traffic erroneously. More importantly, when the MLAG state is primary on one switch and inactive on another, the peer switches are acting as two independent L2 switches rather than as one logical L2 switch. In this scenario, the downstream switch will have its LAG (logical L2 port) connected to two distinct L2 switches, which can result in spanning tree problems on the downstream switch because it will receive BPDUs with different system IDs on the same LAG. If spanning tree is not running, an L2 loop will be created between MLAG peer switches and the downstream device. Allowing LACP to negotiate the downstream LAG will avoid this problem because

only ports connected to a given peer will be active in the LAG., and the other set of ports will not be able to bridge traffic.

The example below does not follow this convention to emphasize the parameters that are distinct. The example in MLAG Implementation Example follows the best practices convention.

Examples

- These ***switch1*** commands bundle Ethernet interfaces ***3*** and ***4*** in ***port channel 20***, then associate that port channel with ***mlag 12***.

```
switch1(config)# interface ethernet 3-4
switch1(config-if-et3-4)# channel-group 20 mode active
switch1(config-if-et3-4)# interface port-channel 20
switch1(config-if-po20)# mlag 12
switch1(config-if-po20)# exit
switch1(config)#
```

- These ***switch2*** commands bundle Ethernet interfaces ***9*** and ***10*** in ***port channel 20***, then associate that port channel with ***mlag 12***.

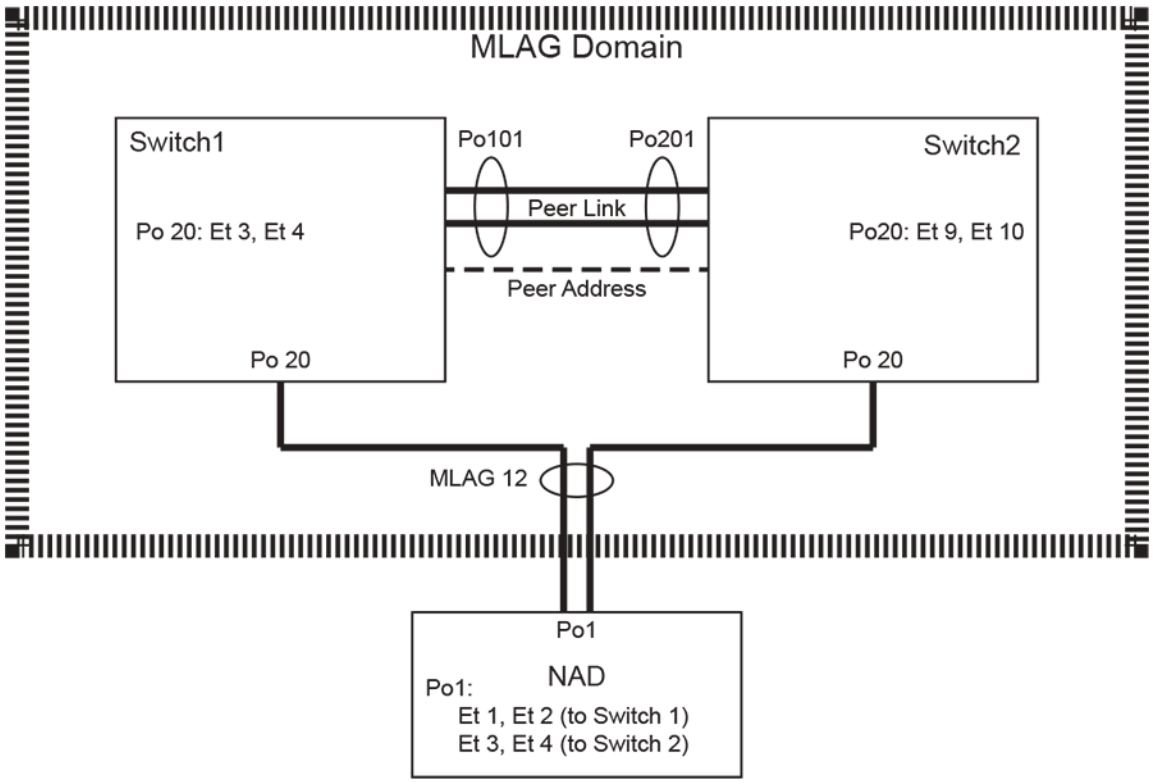
```
switch2(config)# interface ethernet 9-10
switch2(config-if-et9-10)# channel-group 20 mode active
switch2(config-if-et9-10)# interface port-channel 20
switch2(config-if-po20)# mlag 12
switch2(config-if-po20)# exit
switch2(config)#
```

- These commands configure the port channels that attach to the MLAG on network attached device:

```
NAD(config)# interface ethernet 1-4
NAD(config-if-Et1-4)# channel-group 1 mode active
NAD(config-if-Et1-4)# exit
NAD(config)#
```

The following figure displays the result of the interface MLAG configuration.

Figure 11-2: MLAG Interface Configuration



11.3.6 EVPN - MLAG Single-homed Hosts

Add a local VTEP to each MLAG peer for the control plane to advertise singly connected hosts as being directly behind a specific local VTEP-MLAG peer. Outgoing EVPN route advertisements contain nexthop and router MAC extended community when multi-VTEP MLAG mode is enabled. The following table summarizes the information. For symmetric IRB support, MLAG shared router MAC must be enabled.

Table 65: Route Advertisement

Routes	Single-homed Host	Multi-homed Host
MAC/IP routes (Type-2)	Local VTEP IP / Bridge MAC	MLAG VTEP IP / Shared MLAG Router MAC
IMET routes (Type-3)	MLAG VTEP IP	
IP Prefix routes (Type-5)	Local VTEP IP / Bridge MAC	

In multi-VTEP MLAG mode, IP prefix routes are advertised independently by each MLAG peer with its own local VTEP IP as nexthop, even when both peers are connected to that route. Additionally, egress VXLAN packets use the appropriate source IP to match what is advertised by the EVPN control plane.

Show Commands

The following displays the nexthop of locally generated EVPN Type-2 and Type-3 routes. The real IP address helps identify the VTEP being used. The nexthop of locally generated EVPN Type-5 routes are displayed as

"-". The shared MLAG VTEP IP is ***1.1.1.1*** and the local VTEP IPs of the two MLAG peers are ***1.0.1.1*** and ***1.0.2.2***.

```
switch# show bgp evpn
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

Network Next Hop  Metric LocPref Weight Path
* > RD: 1.0.1.1:200 mac-ip 52de.3c26.a0b0 10.2.0.2
                1.0.1.1  -  -     0  i
* > RD: 1.0.1.1:100 mac-ip 822c.0630.7ef4 10.1.0.2
                1.1.1.1   -  -    0   i
 RD: 1.0.2.2:100 mac-ip 822c.0630.7ef4 10.1.0.2
                1.1.1.1   -  100  0   i
* > RD: 1.0.2.2:200 mac-ip 8650.1ecc.3595 10.2.0.3
                1.0.2.2   -  100   0  i
* > RD: 1.0.1.1:100 imet 1.1.1.1
                1.1.1.1   -  -     0  i
* > RD: 1.0.1.1:200 imet 1.1.1.1
                1.1.1.1   -  -      0       i
* > RD: 31000:300 ip-prefix 10.1.0.0/16
                -    -  -     0  i
*  RD: 31000:300 ip-prefix 10.1.0.0/16
                1.0.2.2   -  100         0  i
```

Limitations

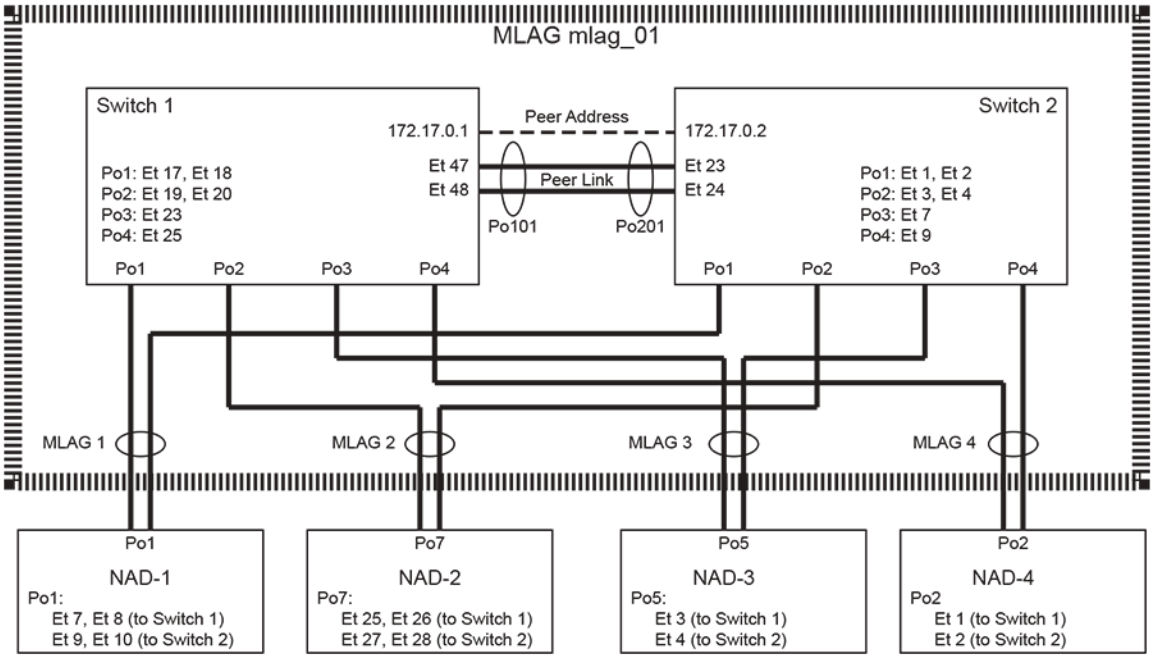
- BUM traffic is advertised on the MLAG VTEP IP. VXLAN flooded packets may be seen on the peer-link.
- The singly connected interfaces or non-MLAG interfaces have to be placed in an exclusive vlan that does not have any MLAG interface as a member.
- MLAG shared router MAC needs to be enabled to support symmetric IRB.
- For EVPN Type-5 setup, the VRF to VNI mapping must be configured on both VTEPs of the MLAG pair.
- For EVPN Type-5 setup as well as EVPN symmetric IRB setup, the VRF to VNI mapping must be configured to be the same across all the VTEPs in the network. That is, the same VRF must be mapped to the same value of VNI on all the VTEPs which participate in the EVPN network.

11.3.7 MLAG Implementation Example

This example creates an MLAG Domain, then configures MLAG connections between the peer switches and four Network Attached Devices (NADs). The MLAG switches connect through a LAG and communicate with

the NADs through MLAGs. Although the NADs can be any device that supports LACP LAGs, the devices in this example are Arista switches.

Figure 11-3: MLAG Implementation Example



11.3.7.1 Topology

Figure 25: MLAG Implementation Example displays the MLAG topology. Switch 1 and Switch 2 are MLAG peers that logically represent a single Layer 2 switch. The peer link between the switches contains the following interfaces:

- ***Switch 1: ethernet 47***, ***ethernet 48***
- ***Switch 2: ethernet 23***, ***ethernet 24***

The example configures MLAGs from the MLAG Domain to four network attached devices (NAD-1, NAD-2, NAD-3, NAD-4).

11.3.7.2 Configuring the Peer Switch Connections

To configure the switches in the described topology, perform the tasks in these sections:

- Configuring the Peer Switch Port Channels
- Configuring the Peer Switch SVIs
- Configuring the Peer Links

11.3.7.2.1 Configuring the Peer Switch Port Channels

These commands create the port channels the switches use to establish the peer link.

These commands create port channels on Switch1

```
switch1(config)# interface ethernet 47-48
switch1(config-if-et47-48)# channel-group 101 mode active
switch1(config-if-et47-48)# interface port-channel 101
switch1(config-if-po101)# switchport mode trunk
switch1(config-if-po101)# switchport trunk group peertrunk
```

```
switch1(config-if-po101)# exit
switch1(config)#
```

These commands create port channels on Switch2

```
switch2(config)# interface ethernet 23-24
switch2(config-if-et23-24)# channel-group 201 mode active
switch2(config-if-et23-24)# interface port-channel 201
switch2(config-if-po201)# switchport mode trunk
switch2(config-if-po201)# switchport trunk group trunkpeer
switch2(config-if-po201)# exit
switch2(config)#
```

11.3.7.2.2 Configuring the Peer Switch SVIs

For each peer switch, these commands create an SVI and associate it to the trunk group assigned to the peer link port channel. STP is disabled on the VLAN.

These commands configure the SVI on Switch1

```
switch1(config)# vlan 4094
switch1(config-vlan-4094)# trunk group peertrunk
switch1(config-vlan-4094)# interface vlan 4094
switch1(config-if-vl4094)# ip address 172.17.0.1/30
switch1(config-if-vl4094)# no autostate
switch1(config-if-vl4094)# exit
switch1(config)# no spanning-tree vlan-id 4094
switch1(config)#
```

These commands configure the SVI on Switch2

```
switch2(config)# vlan 4094
switch2(config-vlan-4094)# trunk group trunkpeer
switch2(config-vlan-4094)# interface vlan 4094
switch2(config-if-vl4094)# ip address 172.17.0.2/30
switch2(config-if-vl4094)# no autostate
switch2(config-if-vl4094)# exit
switch2(config)# no spanning-tree vlan-id 4094
switch2(config)#
```

11.3.7.2.3 Configuring the Peer Links

These commands create the peer links on each MLAG switch.

These commands create peer links on Switch1

```
switch1(config)# mlag configuration
switch1(config-mlag)# local-interface vlan 4094
switch1(config-mlag)# peer-address 172.17.0.2
switch1(config-mlag)# peer-link port-channel 101
switch1(config-mlag)# domain-id mlag_01
switch1(config-mlag)# heartbeat-interval 2500
switch1(config-mlag)# reload-delay 150
switch1(config-mlag)# exit
switch2(config)#
```

These commands create peer links on Switch2

```
switch2(config)# mlag configuration
switch2(config-mlag)# local-interface vlan 4094
switch2(config-mlag)# peer-address 172.17.0.1
switch2(config-mlag)# peer-link port-channel 201
switch2(config-mlag)# domain-id mlag_01
switch2(config-mlag)# heartbeat-interval 2500
switch2(config-mlag)# reload-delay 150
switch2(config-mlag)# exit
switch2(config)#
```

11.3.7.3 Configuring the Network Attached Devices

These commands create the LAGs on the Network Attached Devices that connect to the MLAG domain.

These commands configure the port channels on NAD-1

```
NAD-1(config)# interface ethernet 7-10
NAD-1(config-if-Et7-10)# channel-group 1 mode active
NAD-1(config-if-Et7-10)# exit
NAD-1(config)#
```

These commands configure the port channels on NAD-2

```
NAD-2(config)# interface ethernet 25-28
NAD-2(config-if-Et25-28)# channel-group 7 mode active
NAD-2(config-if-Et25-28)# exit
NAD-2(config)#
```

These commands configure the port channels on NAD-3

```
NAD-3(config)# interface ethernet 3-4
NAD-3(config-if-Et3-4)# channel-group 5 mode active
NAD-3(config-if-Et3-4)# exit
NAD-3(config)#
```

These commands configure the port channels on NAD-4

```
NAD-4(config)# interface ethernet 1-2
NAD-4(config-if-Et1-2)# channel-group 2 mode active
NAD-4(config-if-Et1-2)# exit
NAD-4(config)#
```

11.3.7.4 Verification

The following tasks verify the MLAG peer and connection configuration:

- Verify the Peer Switch Connection
- Verify the MLAGs
- Verify Spanning Tree Protocol (STP)
- Verify the MLAG Port Channel
- Verify the VLAN Membership

11.3.7.4.1 Verify the Peer Switch Connection

To display the MLAG configuration and the MLAG status on ***switch 1***, use the **show mlag** command:

```
switch1# show mlag
MLAG Configuration:
domain-id          :            mlag_01
local-interface    :          Vlan4094
peer-address       :       172.17.0.2
peer-link          :    Port-Channel101

MLAG Status:
state              :             Active
peer-link status   :                 Up
local-int status   :                 Up
system-id          :  02:1c:FF:00:15:38

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   4
```

To display the MLAG configuration and the MLAG status on ***switch 2***, use the **show mlag** command:

```
switch2# show mlag
MLAG Configuration:
domain-id          :            mlag_01
local-interface    :          Vlan4094
peer-address       :       172.17.0.1
peer-link          :    Port-Channel102

MLAG Status:
state              :             Active
peer-link status   :                 Up
local-int status   :                 Up
system-id          :  02:1c:FF:00:15:41

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   4
```

11.3.7.4.2 Verify the MLAGs

The show mlag interfaces command displays MLAG connections between the MLAG switches and the Network Attached Devices.

- This show mlag interfaces command displays MLAG connections between the MLAG peer ***switch 1*** and the network attached devices:

```
switch1# show mlag interfaces
                                                                  local/remote
  mlag     desc                    state     local     remote         status
-------------------------------------------------------------------------------
     1     sw1.po1          active-full       Po1       Po1          up/up
     2     sw1.po2          active-full       Po2       Po2          up/up
     3     sw1.po3          active-full       Po3       Po3          up/up
     4     sw1.po4          active-full       Po4       Po4          up/up
```

- The following show mlag interfaces command, with the **detail** option, displays MLAG connections between the MLAG peer ***switch 1*** and the network attached devices.

```
switch2#show mlag interfaces detail
                                local/remote
mlag       state local remote   oper   config          last change  changes
--------------------------------------------------------------------------------
   1 active-full   Po1    Po1  up/up  ena/ena  6 days, 2:08:28 ago        5
   2 active-full   Po2    Po2  up/up  ena/ena  6 days, 2:08:30 ago        5
   3 active-full   Po3    Po3  up/up  ena/ena  6 days, 2:08:33 ago        5
   4 active-full   Po4    Po4  up/up  ena/ena  6 days, 2:08:41 ago        5 switch2#
```

11.3.7.4.3 Verify Spanning Tree Protocol (STP)

STP functions can be displayed from each peer switch. MLAG interfaces are displayed as a single entry. Configured interfaces on each switch that are not included in an MLAG are displayed. Local interfaces have the normal notation; remote interfaces are preceded by ***P*** or ***Peer***.

VLAN Output 1: Assume VLAN 3903 includes MLAG 1

```
switch1# show spanning-tree vlan-id 3903
Spanning tree instance for vlan 3903
VL3903
  Spanning tree enabled protocol rapid-pvst
  Root ID    Priority    36671
             Address     001c.730c.3009
             Cost        1999 (Ext) 0 (Int)
             Port        105 (Port-Channel5)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    36671  (priority 32768 sys-id-ext 3903)
             Address     021c.7300.1319
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Po1              root       forwarding 1999      128.105  P2p switch1#
```

The output displays ***mlag 1*** under its local interface name (***Po1***). A peer interface is not displayed because spanning tree considers the local and remote Port Channels as a single MLAG interface.

VLAN Output 2: Assume VLAN 3908 does not include any MLAGs

```
switch1# show spanning-tree vlan-id 3908
Spanning tree instance for vlan 3908
VL3908
  Spanning tree enabled protocol rapid-pvst
  Root ID    Priority    36676
             Address     021c.7300.1319
             This bridge is the root

  Bridge ID  Priority    36676  (priority 32768 sys-id-ext 3908)
             Address     021c.7300.1319
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- ---------------
Et17             designated forwarding 2000      128.217  P2p
Et18             designated forwarding 2000      128.218  P2p
PEt17            designated forwarding 2000      128.17   P2p
PEt18            designated forwarding 2000      128.18   P2p
```

The output displays all interfaces from both switches. Each interface is explicitly displayed because they are individual units that STP must consider when selecting ports to block.

- ***Et17*** and ***Et18*** are located on the switch where the **`show spanning-tree`** command is issued.
- ***PEt17*** and ***PEt18*** are located on the remote switch from where the command was issued

An identical command issued on the peer switch displays similar information.

Verify the MLAG does not create topology loops (show spanning-tree blocked)

```
switch1# show spanning-tree blocked
Name       Blocked Interfaces List
-------------------------------------------------------

Number of blocked ports (segments) in the system : 0 switch1#
```

11.3.7.4.4 Verify the MLAG Port Channel

Issue the command **`show port-channel`** for channels ***1-4*** from ***switch 1***:

```
switch# show port-channel 1-4
Port Channel Port-Channel1:
  Active Ports: Ethernet17     Ethernet18     PeerEthernet1 PeerEthernet2
Port Channel Port-Channel2:
  Active Ports: Ethernet19     Ethernet20     Ethernet21     Ethernet22
                PeerEthernet3 PeerEthernet4 PeerEthernet5 PeerEthernet6
Port Channel Port-Channel3:
  Active Ports: Ethernet23     Ethernet24     PeerEthernet7 PeerEthernet8
Port Channel Port-Channel4:
  Active Ports: Ethernet25     Ethernet26     PeerEthernet9 PeerEthernet10
```

Issue the command **`show port-channel load-balance fields detailed`** command for ***channel 1*** from ***switch 2***:

```
switch2# show port-channel 1 detailed
Port Channel Port-Channel1:
  Active Ports:
       Port               Time became active       Protocol    Mode
     ------------------------------------------------------------------------
       Ethernet17         7/7/11 15:27:36          LACP        Active
       Ethernet18         7/7/11 15:27:36          LACP        Active
       PeerEthernet1      7/7/11 15:27:36          LACP        Active
       PeerEthernet2      7/7/11 15:27:36          LACP        Active
```

11.3.7.4.5 Verify the VLAN Membership

The **`show vlan`** command displays VLAN member ports, including MLAG ports and ports on each peer not bundled in an MLAG.

```
Switch1# show vlan 3903, 3908
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
3903  ar.mg.rn.172.17.254.16/29        active    Cpu, Po1
3908  po.ra.ar.mg.172.17.254.64/29     active    Cpu, Et17, Et18, PEt17, PEt18
```

11.3.8 MLAG Commands

Global MLAG Commands

- mlag configuration (global configuration)

MLAG Interface Configuration Commands

- mlag (port-channel interface configuration)

MLAG Configuration Commands

- domain-id

- dual-primary detection delay
- dual-primary recovery delay
- heartbeat-interval (MLAG)
- local-interface
- peer-address
- peer-address heartbeat
- peer-link
- reload-delay mlag
- reload-delay mode
- reload-delay non-mlag
- shutdown (MLAG)

Display Commands

- show mlag
- show mlag interfaces
- show mlag interfaces members
- show mlag interfaces states
- show mlag issu warnings

11.3.8.1 domain-id

The **domain-id** command specifies a name for the Multi-chassis Link AGgregation (MLAG) domain.

The **no domain-id** and **default domain-id** commands remove the MLAG domain name by deleting the **domain-id** statement from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

domain-id ***identifier***

no domain-id

default domain-id

Parameters

identifier alphanumeric string that names the MLAG domain.

Example

This command names the MLAG domain ***mlag1***.

```
switch(config)# mlag
switch(config-mlag)# domain-id mlag1
switch(config-mlag)#
```

11.3.8.2 dual-primary detection delay

Use the **dual-primary detection delay** command to configure a dual primary detection delay with an optional action to errdisable all interfaces on secondary MLAG peer after a dual primary condition is detected.

Command Mode

MLAG configuration mode

Command Syntax

dual-primary detection delay ***seconds*** [action erridisable all-interfaces]

no dual-primary detection delay ***seconds*** [action erridisable all-interfaces]

default dual-primary detection delay ***seconds*** [action erridisable all-interfaces]

Parameters

- ***seconds*** Dual primary detection delay in seconds.
- **action** Specifies the action when dual-primary is detected.
- **erridisable** Errdisable interfaces.
- **all-interfaces** Disables all Ethernet interfaces except the peer-link.

Examples

- In this example, the command errdisables all physical interfaces on secondary MLAG peer when dual primary condition is detected.

  ```
  switch(config-mlag)# dual-primary detection delay 5 action errdisable all-
  interfaces
  ```

  Both MLAG peers must have equivalent configurations.
- The following command removes the Dual Primary Detection feature. You must unconfigure both MLAG peers.

  ```
  switch(config-mlag)# no dual-primary detection
  ```

11.3.8.3 dual-primary recovery delay

Use the **dual-primary recovery delay** command to configure dual-primary detection recovery-delay for MLAG interfaces and non-MLAG interfaces. Negating the configurations or configuring default values makes both recovery delay values reset back to ***0***. The non-MLAG delay must always be less than the MLAG delay so you have more time for L3 convergence before enabling MLAG interfaces.

Command Mode

MLAG configuration mode

Command Syntax

dual-primary recovery delay mlag ***seconds*** non-mlag ***seconds***

no dual-primary recovery delay mlag ***seconds*** non-mlag ***seconds***

default dual-primary recovery delay mlag ***seconds*** non-mlag ***seconds***

Parameters

- **mlag** ***seconds*** Delay in seconds after dual-primary detection resolves until non peer-link ports that are part of an MLAG are enabled. Range ***0 - 1000*** seconds. A suggested value for MLAG is ***60*** seconds. These values can be adjusted depending on the network scale.
- **non-mlag** ***seconds*** Delay in seconds after dual-primary detection resolves until ports that are not part of an MLAG are enabled. Range ***0 - 1000*** seconds. A suggested value for non-MLAG is ***0*** seconds. These values can be adjusted depending on the network scale.

Example

```
switch(config)# mlag
switch(config-mlag)# dual-primary recovery delay mlag 60 non-mlag 0
```

11.3.8.4 heartbeat-interval (MLAG)

The **heartbeat-interval** command configures the interval at which heartbeat messages are issued in a Multi-chassis Link AGgregation (MLAG) configuration.

The **no heartbeat-interval** and **default heartbeat-interval** commands revert the heartbeat interval to the default setting by removing the **heartbeat-interval** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

heartbeat-interval ***period***

no heartbeat-interval

default heartbeat-interval

Parameters

period Interval duration in milliseconds. Value ranges from ***1000*** through ***30000*** milliseconds. Default interval is ***4000*** milliseconds.

Guidelines

Heartbeat messages flow independently in both directions between the MLAG peers. If a peer stops receiving heartbeat messages within the expected time frame (***30*** seconds), the other peer can assume it no longer functions and without intervention or repair, the MLAG becomes disabled. Both switches revert to their independent state.



Note: On 7500 and 7500E Series Switches, Arista recommends setting the heartbeat interval to ***10*** seconds.

Example

This command configures the heartbeat interval to ***15000*** milliseconds:

```
switch(config)# mlag
switch(config-mlag)# heartbeat-interval 15000
switch(config-mlag)#
```

11.3.8.5 local-interface

The **local-interface** command assigns a VLAN interface for use in Multi-chassis Link AGgregation (MLAG) configurations. The VLAN interface is used for both directions of communication between the MLAG peers.

The **no local-interface** and **default local-interface** commands delete the VLAN interface assignment by removing the **local-interface** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

local-interface vlan ***vlan_number***

no local-interface

default local-interface

Parameters

vlan_number VLAN number, in the range from ***1*** through ***4094***.

Guidelines

When configuring the local interface, the VLAN interface must exist already. To configure a VLAN interface, issue the command **`interface vlan`**.

Example

This command assigns ***VLAN 4094*** as the local interface.

```
switch(config)# mlag
switch(config-mlag)# local-interface vlan 4094
switch(config-mlag)#
```

11.3.8.6 mlag (port-channel interface configuration)

The **`mlag`** command assigns an MLAG ID to a port-channel. MLAG peer switches form an MLAG when each switch configures the same MLAG ID to a port-channel interface. Only one MLAG ID can be assigned to an interface. An individual MLAG number cannot be assigned to more than one interface.

The **`no mlag`** and **`default mlag`** commands remove the MLAG ID assignment from the configuration mode interface by deleting the corresponding **`mlag`** command from ***running-config***.

Command Mode

Interface-Port Channel Configuration

Command Syntax

`mlag` ***number***

`no mlag`

`default mlag`

Parameters

number Number used as MLAG ID. Value ranges from ***1*** to ***2000***.

Example

These commands configures a port channel and assigns it ***mlag 4***.

```
switch(config)# interface ethernet 5-10
switch(config-if-Et5-10)# channel-group 1 mode active
switch(config-if-Et5-10)# interface port-channel 4
switch(config-if-Po4)# switchport trunk group group4
switch(config-if-Po4)# mlag 4
switch(config-if-Po4)# exit
switch(config)#
```

11.3.8.7 peer-address

The **`peer-address`** command specifies the peer IPv4 address for a Multi-chassis Link AGgregation (MLAG) domain. MLAG control traffic, including keepalive messages, is sent to the peer IPv4 address. If the peer IPv4 address is unreachable, then MLAG peering fails and both peer switches revert to their independent state.

The **`no peer-address`** and **`default peer-address`** commands remove the MLAG peer's IPv4 address assignment by deleting the peer-address command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

peer-address ***ipv4_addr***

no peer-address

default peer-address

Parameters

ipv4_addr MLAG peer IPv4 address.

Example

These commands configure the MLAG peer address.

```
switch(config)# mlag
switch(config-mlag)# peer-address 10.0.0.2
switch(config-mlag)#
```

11.3.8.8 peer-address heartbeat

The peer-address heartbeat command causes the MLAG agent to start using ***Peer-IP*** address in the given VRF for UDP-based heartbeat control messages.

To enable MLAG dual primary detection feature, the command must be configured on both MLAG peers in the MLAG config mode.

Command Mode

MLAG configuration mode

Command Syntex

peer-address heartbeat ***Peer-IP*** [vrf ***vrf_name***]

no peer-address heartbeat ***Peer-IP*** [vrf ***vrf_name***]

default peer-address heartbeat ***Peer-IP*** [vrf ***vrf_name***]

Parameters

- ***Peer-IP*** The Management IP address of the MLAG peer reachable in the VRF **VRF-NAME** (or default VRF if there is no VRF configured).
- **vrf** ***vrf_name*** Named VRF.

Examples

- ```
  switch(config)# mlag
  switch(config-mlag)# peer-address heartbeat 172.30.118.190
  ```
- This example removes the feature.

  ```
  switch(config-mlag)# no peer-address heartbeat
  ```

11.3.8.9 peer-link

The **peer-link** command specifies the interface that connects Multi-chassis Link AGgregation (MLAG) peers. To form an MLAG, two switches are connected through an interface called a peer link. The peer link carries control and data traffic between the two switches. Control traffic includes MLAG-related advertisements and keepalive messages. This information keeps the two switches working as one.

The **no peer-link** and **default peer-link** command remove the peer link by deleting the **peer-link** command from ***running-config***.

Command Mode
  ```

MLAG Configuration

Command Syntax

`peer-link` INT_NAME

`no peer-link`

`default peer-link`

Parameters

INT_NAME denotes the interface type and number of the interface. Values include:

- **ethernet *e_num*** Ethernet interface range specified by ***e_num***.
- **port-channel *p_num*** Channel group interface range specified by ***p_num***.

Example

These commands creates a peer link.

```
switch(config)# mlag configuration
switch(config-mlag)# peer-link port-channel 10
switch(config-mlag)
```

11.3.8.10 mlag configuration (global configuration)

The **`mlag configuration`** command enters MLAG configuration mode to configure Multi-chassis Link AGgregation (MLAG) features. MLAG configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **`exit`** command does not affect the configuration.

The **`no mlag configuration`** and **`default mlag configuration`** commands remove all MLAG configuration commands from ***running-config***.

The **`exit`** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

`mlag configuration`

`no mlag configuration`

`default mlag configuration`

`mlag` and **`mlag configuration`** are identical commands.

Guidelines

An MLAG is formed by connecting two switches through an interface called a peer link. The peer link carries control and data traffic between the switches, including advertisements and keepalive messages. This information coordinates the switches. Functioning peers are in the **active** state.

Each peer switch uses IP-level connectivity between their local addresses and the MLAG peer IP address to form and maintain the peer link.

Commands Available in MLAG Configuration Mode

- domain-id
- heartbeat-interval (MLAG)
- local-interface
- peer-address
- peer-link
- reload-delay mlag

- shutdown (MLAG)

Example

These commands enter MLAG configuration mode and configure MLAG parameters:

```
switch(config)# mlag
switch(config-mlag)# local-interface vlan 4094
switch(config-mlag)# peer-address 10.0.0.2
switch(config-mlag)# peer-link port-channel 10
switch(config-mlag)# domain-id mlagDomain
switch(config-mlag)# heartbeat-interval 2500
switch(config-mlag)# reload-delay 2000
switch(config-mlag)# exit
switch(config)#
```

11.3.8.11 reload-delay mlag

The **`reload-delay mlag`** command configures the reload delay period for MLAG links. The command also specifies the reload delay period for non-MLAG links when the reload-delay non-mlag command is not configured.

Each Arista switch defaults to the recommended reload-delay value, which varies by switch platform:

- **Fixed configuration switches**: ***300*** seconds
- **Trident II modular switches**: ***1200*** seconds
  - 7304
  - 7308
  - 7316
  - 7300X series
- **Sand platform fixed configuration switches**: ***600*** seconds
  - 7280 series (except 7280CR2 and 7280SR2)
  - 7020 series
- **Sand platform modular switches**: ***1800*** seconds
  - 7504
  - 7508
  - 7500E series
  - 7548S
- **Sand Jericho+ fixed configuration switches**: *900* seconds
  - 7280CR2 series
  - 7280SR2 series

The **`no reload-delay mlag`** and **`default reload-delay mlag`** commands restore the default value by deleting the **`reload-delay mlag`** statement from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

`reload-delay` [**`mlag`**] **PERIOD**

`no reload-delay` [**`mlag`**]

`default reload-delay` [**`mlag`**]

Parameters

- **PERIOD** Period that non-peer links are disabled after an MLAG peer reboots. Options include:
  - **infinity** link is not enabled after reboot.
  - ***0*** to ***86400*** disabled link interval (seconds). Default varies by switch platform as described above.

Guidelines

The **reload-delay** and **reload-delay mlag** commands are equivalent.

Example

These commands configure the reload-delay interval to ***15*** minutes.

```
switch(config)# mlag configuration
switch(config-mlag)# reload-delay mlag 900
switch(config-mlag)#
```

11.3.8.12 reload-delay mode

The **reload-delay mode** command specifies the state of LACP LAG ports during the MLAG reload delay period. By default, MLAG ports remain in the errdisabled state during reload delay. This command configures MLAG ports to come up to standby mode before the expiration of the reload delay period.

The **no reload-delay mode** and **default reload-delay mode** commands restore the default behavior of MLAG ports by deleting the **reload-delay mode** statement from ***running-config***. The default behavior is for the MLAG ports to remain in the errdisabled state until the expiration of the reload delay period

Command Mode

MLAG Configuration

Command Syntax

reload-delay mode lacp standby

no reload-delay mode

default reload-delay mode

Related Commands

reload-delay mlag configures the MLAG reload delay period.

Example

These commands configure the MLAG port to come up to standby state before the end of the reload delay period.

```
switch(config)# mlag configuration
switch(config-mlag)# reload-delay mode lacp standby
switch(config-mlag)#
```

11.3.8.13 reload-delay non-mlag

The **reload-delay non-mlag** command specifies the period that non-MLAG links are disabled after an MLAG peer reboots. This interval allows non peer links to learn multicast and OSPF states before the ports start handling traffic. The recommended minimum value required to ensure the forwarding hardware is initialized with the topology state depends on the switch platform:

- Fixed configuration switches: ***300*** seconds (five minutes)
- Sand platform fixed configuration switches (7020 and 7280 series (except 7280CR2 and 7280SR2)): ***600*** seconds (ten minutes)
- Modular switches: ***1200*** seconds (twenty minutes)

When the **reload-delay non-mlag** command is not configured, the reload-delay mlag command specifies the reload delay time for non-MLAG and MLAG links.

The **no reload-delay non-mlag** and **default reload-delay non-mlag** command restores the default behavior by deleting the **reload-delay non-mlag** statement from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

reload-delay non-mlag PERIOD

no reload-delay non-mlag

default reload-delay non-mlag

Parameters

PERIOD Period that non-MLAG links are disabled after an MLAG peer reboots. Options include:

- **infinity** links are not enabled after reboot.
- ***0*** to ***86400*** disabled link interval (seconds). Values range from ***0*** to ***86400*** (24 hours).

Example

These commands configure the reload-delay interval of non-MLAG links to ***20*** minutes.

```
switch(config)# mlag configuration
switch(config-mlag)# reload-delay non-mlag 1200
switch(config-mlag)#
```

11.3.8.14 show mlag interfaces members

The **show mlag interfaces members** command displays information about the Multi-chassis Link AGgregation (MLAG) members on bridged Ethernet interfaces.

Command Mode

EXEC

Command Syntax

show mlag interfaces members

Example

This command displays the MLAG interface members.

```
switch# show mlag interface members
Mlag4 is Port-Channel4
  Active Ports: Ethernet3 PeerEthernet3
Mlag5 is Port-Channel5
  Active Ports: Ethernet14
Mlag7 is Port-Channel7
  Active Ports: Ethernet5 PeerEthernet5
Mlag8 is Port-Channel8
  Active Ports: Ethernet10 PeerEthernet10
Mlag9 is Port-Channel9
  Active Ports: Ethernet15 Ethernet21 PeerEthernet19 PeerEthernet20
Mlag10 is Port-Channel10
  Active Ports: Ethernet19 Ethernet20 PeerEthernet21 PeerEthernet22 switch#
```

11.3.8.15 show mlag interfaces states

The **show mlag interfaces states** command displays information about the Multi-chassis Link AGgregation (MLAG) states on bridged Ethernet interfaces.

Command Mode

EXEC

Command Syntax

show mlag interfaces [**MLAGS**] **states** [**STATE_NAMES**][**INFO_LEVEL**]

Parameters

- **MLAGS** MLAG channels for which command displays data. Options include:
  - ***no parameter*** command displays data for all MLAGs.
  - ***mlag_id*** specifies MLAG for which command displays data. Value ranges from ***1*** to ***2000***.
- **STATE_NAMES** MLAG channels for which command displays data. Parameter may specify more than one name, which can be listed in any order. Valid state names include:
  - **active-full** includes active-full interfaces.
  - **active-partial** includes active-partial interfaces.
  - **configured** includes configured interfaces.
  - **disabled** includes disabled interfaces.
  - **inactive** includes inactive interfaces.
- **INFO_LEVEL** specifies information displayed by command. Options include:
  - ***no parameter*** command displays basic MLAG interface parameters.
  - **detail** command displays detailed MLAG interface state parameters.

Example

This command displays the MLAG interface states that are active-full.

```
switch# show mlag interfaces states active-full
                                                                  local/remote
  mlag     desc                       state     local     remote         status
-------- -------------------- --------------- --------- ---------- ------------
     4     b.po1                active-full       Po4        Po4          up/up
     7     ar.mg.au.po1         active-full       Po7        Po7          up/up
     8     co.po1               active-full       Po8        Po8          up/up
     9     k.po5                active-full       Po9        Po9          up/up
    10     ar.mg.pt.ir.po10     active-full      Po10       Po10          up/up switch#
```

11.3.8.16 show mlag interfaces

The **show mlag interfaces** command displays information about the Multi-chassis Link AGgregation (MLAG) configuration on bridged Ethernet interfaces.

Command Mode

EXEC

Command Syntax

show mlag interfaces [**MLAGS**][**INFO_LEVEL**]

Parameters

- **MLAGS** MLAG channels for which command displays data. Options include:
  - ***no parameter*** command displays data for all MLAGs.
  - ***mlag_id*** specifies MLAG for which command displays data. Value ranges from ***1*** to ***2000***.
- **INFO_LEVEL** specifies information displayed by command. Options include:

- ***no parameter*** command displays basic MLAG interface parameters.
- **detail** command displays detailed MLAG interface parameters.

Example

This command displays output from the **show mlag interfaces detail** command:

```
switch> show mlag interfaces detail
                                local/remote
mlag       state local remote   oper   config          last change  changes
--------------------------------------------------------------------------------
   4 active-full   Po4    Po4  up/up  ena/ena  6 days, 1:19:26 ago        5
   5 active-full   Po5    Po5  up/up  ena/ena  6 days, 1:19:24 ago        5
   6 active-full   Po6    Po6  up/up  ena/ena  6 days, 1:19:23 ago        5
   7 active-full   Po7    Po7  up/up  ena/ena  6 days, 1:19:23 ago        5
```

11.3.8.17 show mlag issu warnings

The **show mlag issu warnings** command displays a warning message regarding the backward-compatibility of this feature before you upgrade.

Command Mode

EXEC

Command Syntax

show mlag issu warnings

Example

This command displays the MLAG backward-compatibility warning message. Refer to the latest version of the release notes for additional information before you upgrade.

```
switch# show mlag issu warnings
If you are performing an upgrade, and the Release Notes for the new
version of EOS indicate that MLAG is not backwards-compatible with the
currently installed version, the upgrade will result in packet loss.

Stp is not restartable. Topology changes will occur during the upgrade
process.

switch#
```

11.3.8.18 show mlag

The **show mlag** command displays information about the Multi-chassis Link AGgregation (MLAG) configuration on bridged Ethernet interfaces.

Command Mode

EXEC

Command Syntax

show mlag [**INFO_LEVEL**]

Parameters

INFO_LEVEL specifies information displayed by command. Options include:

- ***no parameter*** command displays MLAG configuration, status, and ports.
- **detail** command displays MLAG configuration, status, ports, and detailed status.

Example

This command displays output from the **show mlag** command:

```
switch> show mlag
MLAG Configuration:
domain-id          :          ar.mg.mlag
local-interface    :           Vlan3901
peer-address       :       172.17.254.2
peer-link          :      Port-Channel1

MLAG Status:
state              :              Active
peer-link status   :                  Up
local-int status   :                  Up
system-id          :  02:1c:73:00:13:19

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   5 switch>
```

11.3.8.19 shutdown (MLAG)

The **shutdown** command disables MLAG on the switch without modifying the MLAG configuration.

The **no shutdown** and **default shutdown** commands re-enable MLAG by removing the **shutdown** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Example

These commands disable MLAG on the switch.

```
switch(config)# mlag configuration
switch(config-mlag)# shutdown
switch(config-mlag)#
```

11.4 Data Transfer

Arista switches support the transfer of packets (network layer) and frames (data link layer). This chapter describes concepts and processes that are referenced by routing and switching protocols that Arista switches support.

Sections in this chapter include:

- Data Transfer Introduction
- Data Transfer Methods
- MAC Address Table
- Configuring Ports

- Monitoring Links
- PHY test pattern CLI
- Data Transfer Commands

11.4.1 Data Transfer Introduction

Arista switches transfer data through switching, routing, and Layer 3 switching. This chapter provides an introduction to these transfer methods.

Data structures and processes that support data transfer methods and referenced in specific protocol chapters are also described, including:

- routed ports
- switched ports
- MAC address table
- port mirroring
- storm control
- loopback interfaces
- route redistribution
- null0 interfaces
- MTUs

11.4.2 Data Transfer Methods

This section describes these data transfer methods:

- Switching and Bridging
- Routing
- Layer 3 Switching

11.4.2.1 Switching and Bridging

Switching and bridging operations transmit data link layer frames between devices within a single subnet. Each port is assigned a 48 bit Media Access Control (MAC) address. Frames arriving at a hub are bridged, or sent to all other ports on the subnet. Switches can associate ports with their MAC addresses, obviating the need to flood the subnet when sending a frame.

Subnets in the switch are defined by VLANs. A Virtual Local Area Network (VLAN) is a group of devices that are configured to communicate as if they are attached to the same network regardless of their physical location. **`VLANs`** describes VLANS.

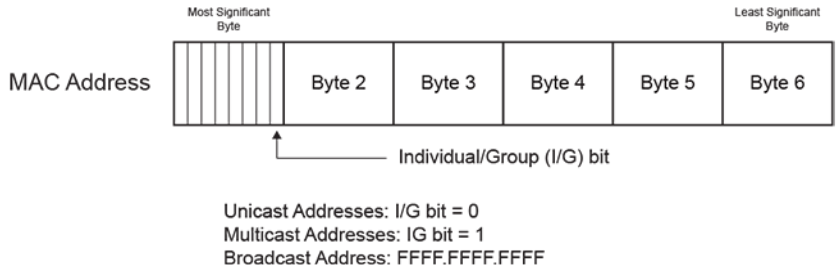
Four MAC address types identify the scope of LAN interfaces that an address represents:

- **unicast:** represents a single interface.
- **broadcast:** represents all interfaces.
- **multicast:** represents a subset of all interfaces.
- **reserved:** assigned to nodes that have no configured MAC address.

The Individual/Group (I/G) bit distinguishes unicast MAC addresses from multicast addresses. As shown in Figure 26: MAC Address Format , the I/G bit is the least significant bit of the most significant byte in a MAC address.

11.4.2.1.1 MAC Address Format

Figure 11-4: MAC Address Format



- Unicast address: the I/G bit is ***0***: ***1234.1111.1111*** is a unicast MAC address (the most significant byte is an even number).
  - Reserved address: all bits set to ***0 (0000.0000.0000)***.
- Multicast address: the I/G bit is ***1***: ***1134.1111.1111*** is a multicast MAC address (the most significant byte is an odd number).
- Broadcast address: all bits set to ***1 (FFFF.FFFF.FFFF)***.

Examples

- The following are unicast MAC addresses:

```
0200.0000.0000
1400.0000.0000
```

- The following are multicast MAC addresses:

```
0300.0000.0000
2500.0000.0000
```

The following sections describe MAC address functions and data structures:

- Assigning a MAC Address to an Interface
- MAC Address Table

11.4.2.2 Routing

Routing transmits network layer packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range. Connected subnets have IP address ranges that do not overlap. A router connects multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address.

IPv4 and IPv6 are internet layer protocols that facilitate packet-switched networking, including transmissions across multiple networks.

These chapters describe available IP features:

- IPv4
- IPv6

11.4.2.2.1 Static Routing

Static routes are entered through the CLI and are typically used when dynamic protocols are unable to establish routes to a specified destination prefix. Static routes are also useful when dynamic routing protocols are not available or appropriate.

Creating a static route associates a destination IP address with a local interface. The routing table refers to these routes as ***connected*** routes that are available for redistribution into routing domains defined by dynamic routing protocols.

These sections describe static route configuration commands:

- IPv4 Address Configuration
- Configuring Default and Static IPv6 Routes

11.4.2.2.2 Dynamic Routing

Dynamic routes are established by dynamic routing protocols. These protocols also maintain the routing table and modify routes to adjust for topology or traffic changes. Routing protocols assist the switch in communicating with other devices to exchange network information, maintaining routing tables, and establishing data paths.

The switch supports these dynamic routing protocols:

- Open Shortest Path First – Version 2
- Open Shortest Path First – Version 3
- Border Gateway Protocol (BGP)
- Routing Information Protocol
- IS-IS

11.4.2.3 Layer 3 Switching

Layer 3 switches establish data paths through routing processes (Layer 3) and transfer data as a switch (Layer 2) through speed-optimized hardware. Layer 3 switches use a control plane (routing) and data plane (switching) to manage these processes.

11.4.2.3.1 Control plane

The control plane builds and maintains the IP routing table, which identifies IP packet routes in terms of destination addresses. The routing table defines a route by its next hop address and the egress interface that accesses the next hop.

The control plane derives routing information from three sources:

- Status of physical and virtual interfaces on the switch.
- Static routes entered through the CLI.
- Routes established through dynamic routing protocols.

Applying an ACL to the Control Plane

The control plane supports routing and management functions, handling packets that are addressed to the switch without regard to any switch interface.

To apply an IP ACL to the control plane, enter ip access-group (Control Plane mode) in control-plane mode. The system control-plane command places the switch in control-plane mode.

ACLs and Route Maps describes access control lists.

Example

These commands place the switch in control-plane mode and assigns ***CP-Test1*** to the control plane.

```
switch(config)# system control-plane
switch(config-system-cp)# ip access-group CP-Test1 in
switch(config-system-cp)#
```

11.4.2.3.2 Data plane

The data plane routes IP packets based on information derived by the control plane. Each packet's path includes Layer 2 addresses that reach its next hop destination. The data plane also performs other operations required by IP routing, such as recalculating IP header checksums and decrementing the Time-To-Live (TTL) field.

Arista data planes support these packet forwarding modes:

- **Store and forward:** the switch accumulates entire packets before forwarding them.
- **Cut through:** the switch begins forwarding frames before their reception is complete.

Cut through mode reduces switch latency at the risk of decreased reliability. Packet transmissions can begin immediately after the destination address is processed. Corrupted frames may be forwarded because packet transmissions begin before CRC bytes are received.

Packet forwarding mode availability varies by switch platform:

- **Arad**: store and forward mode only.
- **FM6000**: both modes are available.
- **Petra**: store and forward mode only.
- **Trident**: both modes are available.
- **Trident II**: both modes are available.

The data plane is also referred to as the forwarding plane.

Data Plane Forwarding Mode Configuration

The switch forwarding-mode command specifies the forwarding mode of the switch's data plane. This command is available on Trident, Trident II, and FM6000 platform switches. The forwarding mode is **store-and-forward** on Arad and Petra platform switches.

Examples

- This command changes the forwarding mode to **store-and-forward**.

```
switch(config)# switch forwarding-mode store-and-forward
switch(config)#
```

- The show switch forwarding-mode command displays the switch's forwarding mode.

```
switch(config)# show switch forwarding-mode
Current switching mode:    store and forward
Available switching modes: cut through, store and forward
```

11.4.3 MAC Address Table

The switch maintains a MAC address table for switching frames efficiently between ports. The MAC address table contains static and dynamic MAC addresses.

- Static MAC addresses are entered into the table through a CLI command.
- Dynamic MAC addresses are entered into the table when the switch receives a frame whose source address is not listed in the MAC address table. The switch builds the table dynamically by referencing the source address of frames it receives.

11.4.3.1 MAC Address Table Configuration

These sections describe MAC address table configuration tasks.

- Static MAC Address Table Entries

- Dynamic MAC Address Table Entries

11.4.3.1.1 Static MAC Address Table Entries

The MAC address table accepts static MAC addresses, including multicast entries. Each table entry references a MAC address, a VLAN, and a list of Layer 2 (Ethernet or port channel) ports. The table supports three entry types: unicast drop, unicast, and multicast.

- A drop entry does not include a port.
- A unicast entry includes one port.
- A multicast entry includes at least one port.

Packets with a MAC address (source or destination) and VLAN specified by a drop entry are dropped. Drop entries are valid for only unicast MAC addresses.

The mac address-table static command adds a static entry to the MAC address table.

Examples

- This command adds a static entry for unicast MAC address ***0012.3694.03ec*** to the MAC address table.

```
switch(config)# mac address-table static 0012.3694.03ec vlan 3 interface
 Ethernet
7
switch(config)# show mac address-table static
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   3    0012.3694.03ec    STATIC      Et7
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- This command adds the static entry for the multicast MAC address ***0112.3057.8423*** to the MAC address table.

```
switch(config)# mac address-table static 0112.3057.8423 vlan 4 interface
port-channel 10 port-channel 12
switch(config)# show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
Total Mac Addresses for this criterion: 0

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   4    0112.3057.8423    STATIC      Po10 Po12
Total Mac Addresses for this criterion: 1
```

```
switch(config)#
```

11.4.3.1.2 Dynamic MAC Address Table Entries

Learning Mode

The switch maintains a MAC address table for switching frames efficiently between VLAN ports. When the switch receives a frame, it associates the MAC address of the transmitting interface with the recipient VLAN and port. When MAC address learning is enabled for the recipient port, the entry is added to the MAC address table. When MAC address learning is not enabled, the entry is not added to the table.

The switchport mac address learning command enables MAC address learning for the configuration mode interface. MAC address learning is enabled by default on all Ethernet and port channel interfaces.

Example

These commands disables MAC address learning for ***interface ethernet 8***, then displays the active configuration for the interface.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# no switchport mac address learning
switch(config-if-Et8)# show active
interface Ethernet8
no switchport mac address learning
switch(config-if-Et8)#
```

Aging Time

Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

Aging time ranges from ***10*** to ***1000000*** seconds with a default of ***300*** seconds (five minutes).

Example

This command sets the MAC address table aging time to two minutes (***120*** seconds).

```
switch(config)# mac address-table aging-time 120
switch(config)#
```

The mac address-table aging-time command configures the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

Mac Moves

Secure MAC addresses is allowed to move when they appear on another interface, when configured. By default, secure MAC addresses does not move.

```
switch(config)# default switchport port-security mac address moveable
switch(config)#
```

Persistent Port Security

When the persistent PortSec-Protect is enabled, secure MAC addresses persist across device reboots and interface flaps. These MAC addresses can still be aged or moved when configured using the commands

mac address-table aging-time and **default switchport port-security mac address moveable**. Persistent port security is enabled by default, and can be disabled.

```
switch(config)# default switchport port-security persistence disabled
```

Example

show port-security command displays the settings for the new global port security configurations, including MAC aging, MAC moves, and persistent port security.

```
switch(config)# show port-security
Secure address moves: disabled
Secure address aging: disabled
Secure address reboot persistence: enabled
Secure address link down persistence: enabled
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
               (Count)       (Count)          (Count)
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Total Addresses in System: 0
```

Clearing Dynamic Addresses

The clear mac address-table dynamic command removes specified dynamic entries from the MAC address table. Entries are identified by their VLAN and Layer 2 (Ethernet or port channel) interface.

Example

This command clears all dynamic mac address table entries for ***port channel 5*** on ***VLAN 34***.

```
switch(config)# clear mac address-table dynamic vlan 34 interface port-channel
 5
switch(config)
```

11.4.3.2 Displaying the MAC Address Table

The show mac address-table command displays the specified MAC address table entries.

Example

This command displays the MAC address table.

```
switch# show mac address-table
          Mac Address Table
------------------------------------------------------------------
Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
 101    001c.8224.36d7    DYNAMIC     Po2        1       9 days, 15:57:28 ago
 102    001c.8220.1319    STATIC      Po1
 102    001c.8229.a0f3    DYNAMIC     Po1        1       0:05:05 ago
 661    001c.8220.1319    STATIC      Po1
 661    001c.822f.6b22    DYNAMIC     Po7        1       0:20:10 ago
3000    001c.8220.1319    STATIC      Po1
3000    0050.56a8.0016    DYNAMIC     Po1        1       0:07:38 ago
3909    001c.8220.1319    STATIC      Po1
3909    001c.822f.6a80    DYNAMIC     Po1        1       0:07:08 ago
3911    001c.8220.1319    STATIC      Po1
3911    001c.8220.40fa    DYNAMIC     Po8        1       1:19:58 ago
3912    001c.822b.033e    DYNAMIC     Et11       1       9 days, 15:57:23 ago
3913    001c.8220.1319    STATIC      Po1
3913    001c.822b.033e    DYNAMIC     Po1        1       0:04:35 ago
3984    001c.8220.178f    DYNAMIC     Et8        1       4 days, 15:07:29 ago
3992    001c.8220.1319    STATIC      Po1
3992    001c.8221.07b9    DYNAMIC     Po6        1       4 days, 15:13:15 ago
Total Mac Addresses for this criterion: 24
```

```
          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0
```

Beginning with ***EOS Release 4.26.0F***, **PortSec-Protect** enforces a limit on the number of MAC addresses, that can be learn. For example, **PortSec-Protect** is configured with a maximum of 1, **`show mac address-table`** shows a single address installed.

```
switch# show mac address-table
          Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
 101    001c.8224.36d7    DYNAMIC     Po2        1       9 days, 15:57:28 ago
Total Mac Addresses for this criterion: 1
```

11.4.3.3 MAC Address Learning Per-VLAN

MAC address learning per-VLAN enables or disables MAC address learning per-VLAN instead of per-port. When MAC address learning is enabled for the recipient port, the entry is added to the MAC address table. When MAC address learning is disabled, the entry is not added to the table.

11.4.3.3.1 MAC Address Learning Configuration

The mac address learning command enables MAC address learning on a VLAN interface. By default, MAC address learning on a VLAN is enabled.

The switch maintains a MAC address table for switching frames between VLAN ports. When the switch receives a frame, it associates the MAC address of the transmitting interface with the recipient VLAN and port. When MAC address learning is enabled for the recipient port, the entry is added to the MAC address table. When MAC address learning is not enabled, the entry is not added to the table.

To disable MAC learning on a particular VLAN, use **`no mac address learning`** command on a VLAN configuration.

Examples

- These commands enable MAC address learning on ***vlan 10*** configuration.

```
switch(config)# vlan 10
switch(config-vlan-10)# mac address learning
```

- These commands disable MAC address learning on ***vlan 10*** configuration.

```
switch(config)# vlan 10
switch(config-vlan-10)# no mac address learning
```

11.4.4 Configuring Ports

This section describes these port properties:

- Port Mirroring
- Storm Control
- Switched and Routed Ports
- Loopback Ports
- MAC Security
- Null0 Interface

- Maximum Transmission Units (MTU)

11.4.4.1 Port Mirroring

Port mirroring, also known as port monitoring, is the duplication of traffic from a collection of source ports to a destination port. A mirror session correlates a set of source ports to a destination port.

Valid mirror sources are Ethernet or port channel interfaces, including port channels which are part of an MLAG. Mirror destination ports are usually Ethernet interfaces; port channel destination ports are also supported on some platforms.

Note: On platforms which support the use of port channels as mirror destinations, a port channel *must not* be used as a mirror destination if it is a member of an MLAG.

Layer 2 control protocols do not run on destination ports. An interface cannot be in more than one mirror session and cannot simultaneously be a source and destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from only one direction.

- **Ingress Mirroring**: Packets received by a source port are duplicated, including all valid data frames and L2 control PDUs. Ports mirror data before forwarding logic is applied. Packets subsequently dropped because of forwarding decisions are mirrored.
- **Egress Mirroring**: Packets transmitted by a source port are duplicated, with these exceptions:
  - **Flooded/Multicast Packets**: Packets sent to multiple mirror ports generate one copy, except in multi-chip devices when the mirror source and destination ports are on different chips; in this case, an extra copy is generated.
  - **Dropped Packets**: Packets dropped by forwarding decisions (such as output STP state checks) on egress sources are not duplicated. Packets dropped because of congestion may be duplicated.
- **Filtered Mirroring**: Specific packets are selected for mirroring based on PERMIT and DENY configurations.
- **Mirroring to GRE Tunnel**: Mirrored packets are encapsulated with GRE protocols for transiting Layer 3 network.

VLAN tags on duplicate packets from an egress source are identical to tags on inbound source packets.

When a packet's path through the switch includes multiple mirror source ports in different mirror sessions, the traffic is duplicated once and sent to the destination of the highest numbered session.

11.4.4.1.1 Port Mirroring Capacity

Port mirroring capacity varies by platform. This section describes session limits for each platform.

FM6000 Platform Switches

- **Maximum Number of Sessions:** 4.
- **Session Sources:** Ethernet interfaces (any number), Port channel interfaces (any number).
- **Session Destinations:** Ethernet interfaces (any number), Port channel interfaces (any number), CPU.
- Egress IP ACL on destination port is not supported.

Sessions can mirror Rx, Tx, or both ways without impacting the number of available sessions.

Enabling each of the following features reduces the number of available sessions by one: ACL Logging, MLAG Peer Link, sFlow, VTEP Learning (VXLAN), LANZ Sampling

Arad Platform Switches

- **Maximum Number of Sessions:** 14.
- **Session Sources:** Ethernet interfaces (any number), Port channel interfaces (any number).
- **Session Destinations:** Ethernet interfaces (one).
- Egress IP ACL on destination port is not supported.

Sessions can mirror Rx, Tx, or both ways without impacting number of available sessions.

Although the number of configured source interfaces is unlimited, the number of interfaces that can be effectively mirrored is restricted by the destination port speed.

Petra Platform Switches

- **Maximum Number of Sessions:** 16.
- **Session Sources:** Ethernet interfaces (eight for Rx or Tx sessions; four for both ways).
- **Session Destinations:** Ethernet interfaces (eight for Rx or Tx sessions; four for both ways).
- Egress IP ACL on destination port is not supported.

  Sessions can mirror Rx, Tx, or both ways without impacting number of available sessions.

Trident Platform Switches

- **Maximum Number of Sessions:** 4.
- **Session Sources:** Ethernet interfaces (any number), Port channel interfaces (any number).
- **Session Destinations:** Ethernet interfaces (one).
- Egress IP ACL on destination port is supported.

  Mirroring Rx or Tx requires one session. Mirroring both ways requires two sessions.

Trident II Platform Switches

- **Maximum Number of Sessions:** 4.
- **Session Sources:** Ethernet interfaces (any number), Port channel interfaces (any number).
- **Session Destinations:** Ethernet interfaces (one).
- Egress IP ACL on Destination Port is supported.

  Mirroring Rx or Tx requires one session. Mirroring both ways requires two sessions.

11.4.4.1.2 Configuring Mirror Ports

Mirror sessions associate a set of source ports to a destination port using the monitor session source and monitor session destination commands. An interface cannot be used in more than one mirror session and cannot be simultaneously a source and a destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from one direction. On Trident and Trident II platform switches (DCS-7050, DCS-7050X, DCS-7250X, and DCS-7300X series), all frames mirrored on egress are prefixed with an 802.1Q VLAN tag, even when the egress port is configured as an access port. If the capture device cannot process VLAN tags properly, mirroring should be configured exclusively for ingress traffic by specifying **rx** in the monitor session source command.

Filtering on TX traffic in a mirror session is not supported.

Example

These commands configure ***interface ethernet 7*** as the source port and ***Ethernet interface 8*** as the destination port for the ***redirect_1*** mirroring session. The session mirrors ingress and egress traffic.

```
switch(config)# monitor session redirect_1 source ethernet 7
switch(config)# monitor session redirect_1 destination ethernet 8
```

The show monitor session command displays the configuration of the specified port mirroring session.

Example

This command shows the configuration of the ***redirect_1*** mirroring session.

```
switch(config)# show monitor session

Session redirect_1
------------------------

Source Ports

  Both:        Et7

Destination Port: Et8

switch(config)#
```

The monitor session ip access-group command configures an ACL to filter the traffic being mirrored to the destination port.

Example

These commands create an ACL and apply it to filter the traffic mirrored to the destination port by session ***redirect_1***.

```
switch(config)# ip access-list allow-host
switch(config-acl-allow-host)# 10 permit ip host 192.168.11.24 host 10.0.215.23
switch(config-acl-allow-host)# 20 deny ip any any
switch(config-acl-allow-host)# exit
switch(config)# monitor session redirect_1 ip access-group allow-host
switch(config)#
```

11.4.4.1.3 Configuring Filtered Mirroring

Filtered mirroring allows for configuring IPv4, IPv6, and MAC access lists and then updating a monitor session with corresponding configuration changes. EOS mirrors the packets that match permit statements. EOS does not select those packets for mirroring that match deny statements.

Note: EOS supports all standard IPv4, IPv6, and MAC qualifiers.

On Strata series platforms, packets from a single monitor source can be mirrored in multiple sessions that use the same access-list. You can attach multiple monitor sources with various access-lists to a monitor session. Each monitor session should contain one access-list type only. Hence, IPv4, IPv6, and MAC access-lists from the same monitor source must appear in different monitor sessions.

When multiple IPv6 monitor sessions share the same monitor source, only one of the monitor sessions remains active and others are automatically inactivated. When the active monitor session is removed from the monitor source, the system automatically activates the inactive monitor sessions.

Packets matching both IP and MAC access lists behave differently on various platforms.

Platform Series	Behavior of Filtered Mirroring
DCS-7050/7050X, DCS-7250X, and DCS-7300X	When entry packets match both IPv4 and MAC access-lists, mirrored copies are created for both IPv4 and MAC access-lists; and forwarded to configured destinations.
DCS-7280SE and DCS-7500E	When entry packets match both IPv4 and MAC access-lists, a mirrored copy is created only for IPv4 access-list. The behavior of filtered mirroring varies in the following ways when a packet matches an entry in both access-list types: • Mirroring is permitted when a packet contradicts with permit and deny configurations. • Mirroring is denied when an entry packet matches deny configurations in both. • IP access-list is prioritized over MAC access-list when an entry packet matches permit configurations in both.

Note: User-Defined Field (UDF) qualifiers in filtered mirroring access-lists allow matching packets using arbitrary user-defined patterns.

Use the **system profile** command to enable the Mirroring ACL profile that supports matching on IPv6, MAC and UDFs.

The following table provides the matching types supported in default and Mirroring ACL profiles.

Profiles	IPv4	IPv6	MAC	UDF
Default	Yes	No	No	No
Mirroring ACL	Yes	Yes	Yes	Yes

Note: MAC mirroring-ACLs do not accept routed IPv4/IPv6 packets and bridged IPv6 packets.

Examples

- These commands create an IPv4 access-list and then attach the access-list to monitor sessions.

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# 10 permit tcp any any rst
switch(config-acl-acl1)# 20 permit tcp any any syn
switch(config-acl-acl1)# 30 permit tcp any any ack

switch(config)# monitor session 1 source Ethernet1 rx ip access-group acl1
switch(config)# monitor session 1 source Ethernet2 rx ip access-group acl1
switch(config)# monitor session 1 destination <destination>
```

- These commands create an IPv6 access-list and then attach the access-list to monitor sessions.

```
Arista(config)# ipv6 access-list acl2
Arista(config-ipv6-acl-acl2)# 10 permit ipv6 any any

Arista(config)#monitor session 2 source Ethernet4 rx ipv6 access-group acl2
Arista(config)#monitor session 2 destination Ethernet5
```

- These commands configure the same monitor source in multiple monitor sessions.

```
switch(config)# monitor session 1 source Ethernet1 rx ip access-group acl1
```

```
switch(config)# monitor session 1 destination <destination 1>

switch(config)# monitor session 2 source Ethernet1 rx ip access-group acl2
switch(config)# monitor session 2 destination <destination 2>
```

- This command configures access-list priorities for dictating the matching order across multiple access-lists that are attached to the same monitor source.

```
switch(config)# monitor session 1 source Ethernet1 rx ip access-group acl1
priority 1
switch(config)# monitor session 1 destination <destination 1>

switch(config)# monitor session 2 source Ethernet1 rx ip access-group acl2
priority 2
switch(config)# monitor session 2 destination <destination 2>
```

- This command enables the Mirroring ACL profile.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# system profile mirroring-acl
switch(config-hw-tcam)# show hardware tcam profile
                    Configuration        Status
FixedSystem         mirroring-acl        mirroring-acl
switch(config-hw-tcam)#
```

11.4.4.1.4 Filtered Mirroring to CPU

Filtered mirroring to CPU adds a special destination to port mirroring that allows mirrored traffic to be sent to the switch supervisor. The traffic can then be monitored and analyzed locally without the need of a remote port analyzer. Filtered mirroring to CPU can also be used for debugging and troubleshooting configured to mirror RX traffic, TX traffic or both, with up to 14 mirroring profiles used simultaneously. In addition, mirroring to CPU uses control plane protection to limit the rate of the traffic sent to the CPU.

Examples

- These commands configure the source for normal mirroring and the destination to CPU.

```
switch(config)# monitor session mySession source ethernet 3/1 both
switch(config)# monitor session mySession destination cpu
switch(config)#
```

- These commands configure reserved bandwidth and shape rate of mirrored traffic.

```
switch(config)# policy-map type copp copp-system-policy
switch(config-pmap-control-plane-copp-system-policy)# class copp-system-m
irroring
switch(config-pmap-c-copp-system-policy-copp-system-mirroring)# bandwidth
 kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-mirroring)# shape kbps
 4000
switch(config-pmap-c-copp-system-policy-copp-system-mirroring)#
```

- These commands show the current status of mirroring to CPU from the CLI, and display the control plane protection configuration for mirroring to CPU.

```
switch(config)# show monitor session

                      Session mySession

                      ------------------------
```

```
                          Source Ports:

                            Both : Et3/1

                          Destination Ports:

                            Cpu : active (mirror0)
switch(config)#
```

- These commands show the current status of mirroring to CPU from the CLI, and display the control plane protection configuration for mirroring to CPU.

```
switch(config)# show policy-map type copp copp-system-policy class cop-
system-mirroring

                            Class-map: copp-system-mirroring (match-any)

                               shape : None

                               bandwidth : None
switch(config)#
```

11.4.4.1.5 Configuring Filtered Mirroring to GRE Tunnel

The monitor session source and monitor session destination commands configure source and destination ports to the specified port mirroring session in a GRE tunnel.

On DCS-7010T, DCS-7050/7050X, DCS-7060X, DCS-7250X, DCS-7260X, DCS-7300X, a special GRE tunnel destination is supported to mirror ingress packets that are dropped during ASIC forwarding. This GRE destination is referred as the “forwarding-drop” destination, and the corresponding session is called as the “forwarding-drop” session.

Note: Forwarding-drop sessions are the sessions corresponding to forwarding-drop destinations.

Note: From Release ***EOS 4.25.2F*** onwards platforms DCS-7050X, DCS-7060X, DCS-7250X, DCS-7260X, CCS-720X started supporting the **tx** keyword, which specifies that outgoing packets should be mirrored.

Examples

- These commands configure ingress filtered mirroring to a GRE tunnel.

```
switch(config)# monitor session abc source Ethernet1 rx ip access-group acl1
switch(config)# monitor session abc destination tunnel mode gre source
 1.1.1.1
destination 2.2.2.2 ttl 128 dscp 0 protocol 0x88be
```

- These commands configure egress filtered mirroring to a GRE tunnel.

```
switch(config)# monitor session abc source Ethernet1 tx ip access-group acl1
switch(config)# monitor session abc destination tunnel mode gre source
 2.2.2.2
destination 2.2.2.2 ttl 128 dscp 0 protocol 0x88be
```

- This command configures forwarding-drop sessions.

```
switch(config)# monitor session 1 forwarding-drop destination tunnel mode
 gre source 1.1.1.1 destination 2.2.2.2
```

- A forwarding-drop session is configured by using the **forwarding-drop** keyword when configuring the GRE destination:

```
switch(config)# monitor session 1 source <source>
switch(config)# monitor session 1 forwarding-drop destination tunnel mode
 gre
                                  source <sourceIp>
                                  destination <destIp>
                                  [ ttl <value> ]
                                  [ dscp <value> ]
                                  [ protocol <value> ]
                                  [ vrf <value> ]
```

- A mirroring to GRE destination can be configured as follows:

```
switch(config)# monitor session 1 source <source> rx | tx
switch(config)# monitor session 1 destination tunnel mode gre
                                  source <sourceIp>
                                  destination <destIp>
                                  [ ttl <value> ]
                                  [ dscp <value> ]
                                  [ protocol <value> ]
                                  [ vrf <value> ]
```

  The **rx** keyword specifies that incoming packets should be mirrored.

11.4.4.1.6 Security ACL Filtered Mirroring

Security ACL Filtered Mirroring is configured using port security ACLs.

Configuring Security ACL Filtered Mirroring

The following configures ***interface ethernet 8*** as the destination port for the ***redirect_1*** mirroring session, and ***interface ethernet 9*** as the destination port for the ***redirect_2*** mirroring session. A source port is not needed to create a mirror session. Other destination options for monitor sessions such as GRE or CPU are also configurable.

```
switch (config)# monitor session redirect_1 destination ethernet 8
switch (config)# monitor session redirect_2 destination ethernet 9
```

Examples

Egress IPv4 ACL

The following commands create an IPv4 access-list, and then attach the access-list to ***interface ethernet 7*** in the out direction with the following rules.

- matching Rule 10 will be mirrored to ***interface ethernet 8***.
- matching Rule 20 will not be mirrored.
- matching Rule 30 will be mirrored to ***interface ethernet 9***.
- matching Rule 40 will be dropped and not mirrored.

Specifying a mirror session in a deny rule for egress ACL has no effect.

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# 10 permit ip host 10.0.0.4 any mirror session
 redirect_1
switch(config-acl-acl1)# 20 permit ip host 10.0.0.5 any
switch(config-acl-acl1)# 30 permit ip host 10.0.0.6 any mirror session
 redirect_2
switch(config-acl-acl1)# 40 deny ip any any
```

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# ip access-group acl1 out
```

Note: Security ACL Filtered Mirroring has higher priority over standard Port Mirroring.

Using the same configuration as above with ***interface ethernet 7*** as the source port of ***redirect_1***, the following configuration displays the impact on packets egressing from ***interface ethernet 7***.

```
switch(config)# monitor session redirect_1 source ethernet 7
```

- matching Rule 10 and Rule 20 will be mirrored to ***interface ethernet 8***.
- matching Rule 30 will be mirrored to ***interface ethernet 9***.
- matching Rule 40 will be dropped and not mirrored.

Egress IPv6 ACL

The following commands create an IPv6 access-list, and then attach the access-list to ***interface ethernet 7*** in the egress direction.

```
switch(config)# ipv6 access-list acl1
switch(config-ipv6-acl-acl1)# 10 permit ipv6 host 10:10:10:10:10:10:10:1 any
 mirror session redirect1
switch(config-ipv6-acl-acl1)# 20 permit ipv6 host 10:10:10:10:10:10:10:5 any
switch(config-ipv6-acl-acl1)# 30 permit ipv6 host 10:10:10:10:10:10:10:6 any
 mirror session redirect2
switch(config-ipv6-acl-acl1)# 40 deny ipv6 any any

switch(config)# interface ethernet 7
switch(config-if-Et7)# ipv6 access-group acl1 out
```

Note: The mirroring behavior of egress IPv6 ACL is identical to egress IPv4 ACL. The egress IPv6 ACL is supported only on R3 Series and forward.

Egress MAC ACL

The following commands create a MAC access-list, and then attach the access-list to ***interface ethernet 7*** in the out direction. The mirroring behavior of egress MAC ACL is identical to egress IPv4 ACL.

```
switch(config)# mac access-list acl1
switch(config-mac-acl-acl1)# 10 permit 0000.1111.4444 0000.0000.0000 any mirror
 session redirect_1
switch(config-mac-acl-acl1)# 20 permit 0000.1111.5555 0000.0000.0000 any
switch(config-mac-acl-acl1)# 30 permit 0000.1111.6666 0000.0000.0000 any mirror
 session redirect_2
switch(config-mac-acl-acl1)# 40 deny any any

switch(config)# interface ethernet 7
switch(config-if-Et7)# mac access-group acl1 out
```

Ingress IPv4 ACL

The following commands create an IPv4 access-list, and then attach the access-list to ***interface ethernet 7*** in the in direction with the following rules.

- matching Rule 10 and Rule 20 will be mirrored to ***interface ethernet 8***.
- matching Rule 30 will be mirrored to ***interface ethernet 9*** since Security ACL Filtered Mirroring has higher priority.

- matching Rule 40 will be dropped and mirrored to ***interface ethernet 8***.

```
switch(config)# ip access-list acl2
switch(config-acl-acl2)# 10 permit ip host 10.0.0.4 any mirror session
 redirect_1
switch(config-acl-acl2)# 20 permit ip host 10.0.0.5 any
switch(config-acl-acl2)# 30 permit ip host 10.0.0.6 any mirror session
 redirect_2
switch(config-acl-acl2)# 40 deny ip host 10.0.0.7 any mirror session redirect_1

switch(config)# interface ethernet 7
switch(config-if-Et7)# ip access-group acl2 in

switch(config)# monitor session redirect_1 source ethernet 7
```



Note: Unlike egress ACL, mirror session specified in a deny rule for ingress ACL will take effect.

The mirroring behavior of ingress IPv6 and MAC ACLs are identical to ingress IPv4 ACL.

Limitations

- The feature is not supported in AlgoMatch mode.
- Egress Security ACL Filtered Mirroring works on IPv4 - permit rules, and MAC - permit rules.
- By default, egress MAC ACL is disabled. Egress MAC ACL is required to be enabled.
- By default, bridged traffic is not subject to Egress IP ACLs, therefore, the bridged packets will not be mirrored.
- RACL and subinterface ACL are not supported for filtering mirroring.
- If a packet is dropped by an ingress ACL and the destination is GRE, the metadata of the GRE packet cannot be computed as expected.

11.4.4.2 Storm Control

A traffic storm is a flood of packets entering a network, resulting in excessive traffic and degraded performance. Storm control prevents network disruptions by limiting traffic beyond specified thresholds on individual physical LAN interfaces.

Storm control monitors inbound traffic levels over one-second intervals and compares the traffic level with a specified benchmark.

Storm control has three modes:

- **Storm control all**: When inbound traffic exceeds the specified threshold within a one-second control interval, all traffic is dropped until the end of the interval.
- **Storm control broadcast**: When inbound broadcast traffic exceeds the specified threshold within a one-second control interval, broadcast traffic is dropped until the end of the interval.
- **Storm control multicast**: When inbound multicast traffic exceeds the specified threshold within a one-second control interval, multicast traffic is dropped until the end of the interval.

Broadcast and multicast storm control are independent features and can be enabled simultaneously. The **`storm control all`** threshold overrides broadcast and multicast thresholds.

Storm Control Configuration

The storm-control command configures and enables broadcast or multicast storm control on the configuration mode interface. The command provides three mode options:

- **storm-control all** unicast, multicast, and broadcast inbound packet control.
- **storm-control broadcast** broadcast inbound packet control.
- **storm-control multicast** multicast inbound packet control.

An interface configuration can contain three storm-control statements, one with each mode setting. The **storm-control all** threshold overrides broadcast and multicast thresholds.

When storm control is enabled, the switch monitors inbound traffic levels over one second intervals and compares the traffic level with a specified threshold. The threshold is a percentage of the total available port bandwidth and is configurable on each interface for each transmission mode.

Examples

- These commands enable multicast storm control on Ethernet interfaces ***2*** through ***4*** and set a threshold of ***65%***. During each one second interval, the interface drops inbound multicast traffic in excess of ***65%*** of capacity.

```
switch(config)# interface ethernet 2/3/4
switch(config-if-Et4/4/4)# storm-control multicast level 65
switch(config-if-Et4/4/4)#
```

- These commands clear multicast storm control on Ethernet interfaces ***2*** through ***4***.

```
switch(config)# interface ethernet 2/3/4
switch(config-if-Et2/3/4)# no storm-control multicast
switch(config-if-Et2/3/4)#
```

- These commands enable broadcast storm control on Ethernet interfaces ***2*** through ***4*** and set broadcast traffic to ***50%***. During each one second interval, the interface drops inbound multicast traffic in excess of ***50%*** of capacity.

```
switch(config)# interface ethernet 2/3/4
switch(config-if-Et2/3/4)# storm-control broadcast level 50
switch(config-if-Et2/3/4)#
```

- These commands enable broadcast storm control on Ethernet interfaces ***2*** through ***4*** and set a threshold of ***5000*** packets per second (pps).

```
switch(config)# interface ethernet 2/3/4
switch(config-if-Et2/3/4)# storm-control broadcast level pps 5000
switch(config-if-Et2/3/4)#
```

Note: User cannot configure a PPS setting and a percentage setting on the same interface at the same time, they are mutually exclusive.

- These commands clear broadcast storm control on Ethernet interfaces ***2*** through ***4***.

```
switch(config)# interface ethernet 2/3/4
switch(config-if-Et2/3/4)# no storm-control broadcast
switch(config-if-Et2/3/4)#
```

The show storm-control command displays the storm-control level and interface inbound packet capacity for the specified interface.

Examples

- This command displays the storm control configuration for Ethernet ports ***2*** through ***4***.

```
switch(config-if-Et2/3/4)# show storm-control
Port            Type     Level   Units Rate(Mbps)   Status   Drops Reason
Et2/3/4          all     75.00       %       7500   active       0
           multicast     55.00       %       5500   active       0
           broadcast     50.00       %       5000   active       0
switch(config-if-Et2/3/4)#
```

- The output of **show storm-control** command displaying the PPS settings:

```
switch(config-if-Et5)# show storm-control
show storm-control
Port Type      Level Units Rate(Mbps) Status Drops Reason
Et5  broadcast 5000  pps   -          active 101
```

11.4.4.3 Switched and Routed Ports

A switched port is an Ethernet or port channel interface that is configured as a Layer 2 interface. Switched ports bridge frames and are assigned to at least one VLAN. Switched ports are not associated with any IP addresses. By default, Ethernet and port channel interfaces are in switched port mode.

A routed port is an Ethernet or port channel interface that is configured as a Layer 3 interface. Routed ports do not bridge frames and are not members of any VLANs. Routed ports can have IP addresses assigned to them and packets are routed directly to and from the port.

Configuring an interface as a routed port is similar to creating a VLAN with spanning-tree disabled, making the port the only member of that VLAN and configuring the IP address on the switch virtual interface (SVI) associated with the VLAN.

All IP-level interface configuration commands, except **autostate** and **ip virtual-router**, can be used to configure a routed interface. If the interface is reverted to switched port mode, ***running-config*** maintains IP level interface configuration statements. These changes become active again if the interface is configured back to routed port mode.

A LAG that is created with the **channel-group** command inherits the mode of the member port. A LAG created from a routed port becomes a routed LAG. IP-level configuration is not propagated to the LAG from its component members.

The broadcast queue towards the CPU is shared among all interfaces of the forwarding chip. Broadcast storm on a single port adversely impacts other interfaces of the same chip by potentially dropping even low rate broadcast frames. Routed port storm control attempts to mitigate this effect by performing storm control on the broadcast frames for routed ports.

Routed Port Configuration

The switching-routing configuration of Ethernet and port channel interfaces is specified by the switchport and **no switchport** commands. These commands only toggle the interface between switched and routed modes. They have no effect on other configuration states.

The no switchport command places the configuration mode interface in **routed port** mode. Routed ports behave as Layer 3 interfaces. They do not bridge packets and are not VLAN members. An IP address can be assigned to a routed port for the direct routing of packets to and from the interface.

When an interface is configured as a routed port, the switch transparently allocates an internal VLAN whose only member is the routed interface. Internal VLANs are created in the range from ***1006*** to ***4094***. VLANs that are allocated internally for a routed interface cannot be directly created or configured. Allocating Internal VLANs describes VLAN allocation configuration procedures.

Example

This command places ***interface ethernet 5*** in routed port mode.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no switchport
```

Switched Port Configuration

The switchport command places the configuration mode interface in **switched port (Layer 2)** mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments. By default, Ethernet and port channel interfaces are switched ports.

Example

This command places ***interface ethernet 5*** in switched port mode.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport
```

The switchport default mode routed command places the configuration mode interface for a switch with all ports in **switched port (Layer 3) routed** mode, changing the switch with all ports from **switchport default mode access**.

Examples

- This command places a switch with all ports in routed mode.

```
switch(config)# switchport default mode routed
```

- This command places a switch with all ports in access mode.

```
switch(config)# switchport default mode access
```

11.4.4.4 Loopback Interfaces

A loopback interface is a virtual network interface implemented in software that is not tied to a specific hardware interface. Loopback interface configuration mode is used for creating loopback interfaces and modifying their operating parameters.

Internet protocols reserve specific addresses for loopback network segments:

- IPv4 designates ***127/8*** as loopback subnet, which includes ***127.0.0.0*** through ***127.255.255.255***.
- IPv6 designates ::1/128 as the loopback address, which includes ***0:0:0:0:0:0:0:1*** (also written as ::1).

Arista switches support the configuration of 1001 loopback interfaces, numbered from 0 to 1000.

Loopback Interface Configuration

Loopback ports are instantiated by entering loopback interface configuration mode for the desired loopback interface number. Loopback interface configuration mode also provides access to loopback configuration commands. Previously instantiated ports are edited by entering loopback interface configuration mode for the specified interface.

The interface loopback command places the switch in loopback interface configuration mode for the specified interface, creating the specified loopback interface if it does not exist. Configuration mode can also be entered for a range of loopback interfaces, but they must all have been previously created

Example

These commands instantiate ***interface loopback 2*** and assign it IP address ***10.1.1.42/24***.

```
switch(config)# interface loopback 2
switch(config-if-Lo2)# ip address 10.1.1.42
switch(config-if-Lo2)# show active
interface Loopback2
```

```
   ip address 10.1.1.42/24
switch(config-if-Lo2)#
```

11.4.4.5 MAC Security

MAC security restricts input to a switched port by limiting the number of MAC addresses that can access the port. Ports with MAC security enabled restrict traffic to a limited number of hosts, as determined by their MAC addresses. When the limit is exceeded, the port becomes errdisabled.

Port Security Configuration

MAC address security is enabled by switchport port-security . The default MAC address limit on an interface where port security is enabled is one; to change that default limit, use the switchport port-security mac-address maximum command.

Example

These commands enable MAC security on ***interface ethernet 7***, set the maximum number of assigned MAC addresses to 2, assign two static MAC addresses to the interface, and clear the dynamic MAC addresses for the interface.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# switchport port-security
switch(config-if-Et7)# switchport port-security mac-address maximum 2
switch(config-if-Et7)# exit
switch(config)# mac address-table static 0034.24c2.8f11 vlan 10 interface ethernet 7
switch(config)# mac address-table static 4464.842d.17ce vlan 10 interface ethernet 7
switch(config)# clear mac address-table dynamic interface ethernet 7
switch(config)# show port-security
Secure Port      MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                    (Count)        (Count)      (Count)
--------------------------------------------------------------------------
     Et7             2             2            0            Shutdown
--------------------------------------------------------------------------
Total Addresses in System: 1
switch(config)# show port-security mac-address
          Secure Mac Address Table
------------------------------------------------------------
Vlan    Mac Address       Type                      Ports   Remaining Age
                                                              (mins)
----    -----------       ----                      -----   -------------
  10    0034.24c2.8f11    SecureConfigured          Et7     N/A
  10    4464.842d.17ce    SecureConfigured          Et7     N/A
------------------------------------------------------------------------
Total Mac Addresses for this criterion: 2
switch(config)#
```

MAC Security LLDP Bypass

When MAC address security configuration is applied on the interface, it encrypts and decrypts all the other protocols PDU and other data packets. **LLDP bypass** allows LLDP packets to be sent or received from the port even when the port is not authorized.

The following configuration allows LLDP packets to be received or sent from an interface where the MAC security profile is applied.

```
switch(config)# mac security
switch(config-mac-security)# profile test
switch(config-mac-security-profile-test)# l2-protocol lldp bypass unauthorized
```

unauthorized allows the LLDP packet to be received and sent out when MKA session between the MACsec peers is yet to come up.

Show Command

The following command shows LLDP packets is bypassed for encryption or decryption.

```
switch(config)# show mac security interface ethernet 4/4/1 detail
Interface: Ethernet4/4/1
Profile: profile1
SCI: d4:af:f7:2e:67:b0::786
SSCI: 00000002
Controlled port: True
Key server priority: 1
Session rekey period: 30
Traffic: Protected
Bypassed protocols: LLDP
Key in use: c0645d4332ba2e1d4d5fb17f:129
Latest key: None
Old key: c0645d4332ba2e1d4d5fb17f:129(RT)
```

11.4.4.6 Null0 Interface

The **null0 interface** is a virtual interface that drops all inbound packets. A null0 route is a network route whose destination is **null0 interface**. Inbound packets to a null0 interface are not forwarded to any valid address. Many interface configuration commands provide **null0** as an interface option.

11.4.4.7 Maximum Transmission Units (MTU)

The MTU of a communications protocol refers to the size in bytes of the largest frame (Ethernet) or packet (IP) that can be sent on the network.

Different protocols support a variety of MTU sizes. Most IP over Ethernet implementations use the Ethernet V2 frame format, which specifies an MTU of ***1500*** bytes. Jumbo frames are Ethernet frames containing more than ***1500*** bytes.

11.4.4.7.1 Switching interface MTU size

On Arista devices, layer two interfaces (either trunk or access ports) are set with a default ethernet MTU of 9236 bytes. This value cannot be changed and is derived as follows: 9214 + 6 (source MAC) + 6 (dst MAC) + 4 (VLAN tag) + 2 (ether type) + 4 (crc) totals 9236 bytes.

The output of show interfaces command for a layer two interface displays the following:

Trunk

```
Ethernet1 is up, line protocol is up (connected)
 Hardware is Ethernet, address is 001c.731c.5073 (bia 001c.731c.5073)
Ethernet MTU 9214 bytes , BW 1000000 kbit
```

Access

```
Ethernet3 is up, line protocol is up (connected)
 Hardware is Ethernet, address is 001c.731c.5075 (bia 001c.731c.5075)
 Ethernet MTU 9214 bytes , BW 1000000 kbit
```

11.4.4.7.2 Routing Interface MTU Size

The MTU size on Layer 3 interfaces varies between a minimum of ***68*** to the maximum ***9214*** bytes. The default size is ***1500*** bytes. The **`show interface`** output for a Layer 3 interface displays the following:

VLAN Routed Interface

```
Vlan100 is up, line protocol is up (connected)
 Hardware is Vlan, address is 001c.731c.5072 (bia 001c.731c.5072)
 Internet address is 10.1.1.2/24
 Broadcast address is 255.255.255.255
 Address determined by manual configuration
 IP MTU 9214 bytes
```

Physical Routed Interface

```
Ethernet4 is down, line protocol is down (connect)
 Hardware is Ethernet, address is 001c.731c.5072
 Internet address is 10.10.10.10/24
 Broadcast address is 255.255.255.255
 Address determined by manual configuration
 IP MTU 9214 bytes
```

A routed interface fragments packets that exceed the configured IP MTU on the interface. For example, if a ***2000*** byte packet is received on routed interface 1 and is forwarded from routed interface 2 then routed interface 2 fragments the packet into a ***1500*** byte packet plus an additional packet containing the remaining data. This fragmentation should be avoided by configuring a consistent IP MTU across all systems within the operational domain.

The IP MTU set on a routed interface is valid for both IPv4 and IPv6 packets.

MTU Configuration

The mtu command configures the IPv4 and IPv6 Maximum Transmission Unit (MTU) size for the configuration mode interface. An interface's MTU value is displayed with the **`show interface`** command. The command is valid for all routable interfaces.

Examples

- This command sets the MTU size of ***1492*** bytes on ***VLAN interface 20***.

```
switch(config-if-Vl20)# mtu 1492
switch(config-if-Vl20)#
```

- This command displays status for a routed interface.

```
switch(config-if-Et3)# show interface e3
Ethernet3 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.731c.5072
  Internet address is 10.1.1.2/24
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  IP MTU 1500 bytes , BW 1000000 kbit
  Full-duplex, 1Gb/s, auto negotiation: on, uni-link: unknown
  Up 22 days, 7 hours, 47 minutes, 58 seconds
switch(config)#
```

- Using ping on a Linux host, you can test the maximum transmission through the interface.
-

```
[user@linux ~]$ ping -M do -s 1472 10.1.1.2
PING 10.1.1.2 (10.1.1.2) 1472(1500) bytes of data.
1480 bytes from 10.1.1.2: icmp_seq=1 ttl=64 time=0.206 ms
1480 bytes from 10.1.1.2: icmp_seq=2 ttl=64 time=0.191 ms
--- 10.1.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
```

```
rtt min/avg/max/mdev = 0.191/0.198/0.206/0.015 ms
```

The size ***1472*** has ***8*** bytes of ICMP information added and ***20*** bytes of IP headers added, generating a total packet size of ***1500*** bytes.

- The option **-M do** specifies that fragmentation is prohibited for this test.
- The option **-s** specifies the size of the packet being generated.
- A capture of the frame displays total length of ***1514*** bytes on the wire which includes the Ethernet headers and type field.

11.4.5 Monitoring Links

These section describes link monitoring and object tracking processes:

- Object Tracking
- Errdisabled Ports
- Error Disable Detect Cause for ACL
- Configuring Error Disable Recovery Interval for each Cause
- Link Flap Monitoring
- Fabric Link Monitoring
- Rapid Automated Indication of Link-Loss

11.4.5.1 Object Tracking

Object tracking makes it possible for the switch to take action in response to changes in specific switch properties by creating an object to track those properties. When the tracked property changes, the object then changes state, allowing configured agents to react accordingly.

Object Tracking Configuration

The track command creates an object that changes state to reflect changes in a specific switch property. Agents configured to track that object are then able to react to the change.

Example

These commands create an object that tracks the line protocol state on ***interface ethernet 8***, then configures ***interface ethernet 5*** to disable VRRP when that tracked object changes state to **down**.

```
switch(config)# track ETH8 interface ethernet 8 line-protocol
switch(config)# interface ethernet 5
switch(config-if-Et5)# vrrp 1 tracked-object ETH8 shutdown
switch(config-if-Et5)#
```

These commands use object tracking:

- link tracking group
- vrrp tracked-object

11.4.5.2 Errdisabled Ports

The switch places an Ethernet or management interface in **error-disabled** state when it detects an error on the interface. **Error-disabled** is an operational state that is similar to link-down state. Conditions that error-disable an interface include:

- **bpduguard**
- **link-flap**
- **no-internal-vlan**
- **portchannelguard**

- **portsec**
- **tapagg**
- **uplink-failure-detection**
- **xcvr_unsupported**

Most conditions are programmed by the configuration of other features, such as Spanning Tree protocol (bpduguard). Link flap error-disabling is configured through errdisable commands or link flap monitor commands (Link Flap Monitoring).

Error-disabled interfaces are recovered either through manual or automated methods.

To manually recover an interface, enter its configuration mode and execute **shutdown** and **no shutdown** commands.

Example

These commands manually recover ***interface ethernet 30*** from the errdisable state.

```
switch(config)# interface ethernet 30
switch(config-if-Et30)# shutdown
switch(config-if-Et30)# no shutdown
switch(config-if-Et30)#
```

Automated recovery of Ethernet interfaces that are error-disabled by a specified condition is enabled by errdisable recovery cause . The errdisable recovery interval specifies the period that an interface remains disabled until it is enabled and begins operating normally. When the disabling condition persists, recovered interfaces eventually return to the error-disabled state.

Example

These commands configure automated recovery for all interfaces that are error-disabled from link flap and **bpduguard** conditions. Automated recovery begins five minutes after the port is disabled.

```
switch(config)# errdisable recovery cause link-flap
switch(config)# errdisable recovery cause bpduguard
switch(config)# errdisable recovery interval 300
switch(config)#
```

11.4.5.3 Error Disable Detect Cause for ACL

The **no errdisable detect cause acl** command configures routed ports, subinterfaces, and physical ports to not get into the **errdisabled** state on ACL failure, the default behavior. To reestablish the default behavior, use the **errdisable detect cause acl** command.

The following displays the output when ***errdisabling*** is enabled for ACLs.

```
switch(config) #show errdisable detect
   Errdisable Reason           Detection Status
------------------------------ ----------------
   acl                         Enabled
```

The following displays the output when ***errdisabling*** is disabled for ACLs.

```
switch(config)# show errdisable detect
   Errdisable Reason           Detection Status
------------------------------ ----------------
   acl                         Disabled
```

11.4.5.4 Configuring Error Disable Recovery Interval for each Cause

The duration after which an interface tries to recover from being error disabled is programmable for each trigger which causes the interface to be error disabled using the **`errdisable recovery cause NAME_OF_CAUSE interval DURATION`** command. The command applies only to interfaces that are enabled for error recovery after being error disabled.

Example

This command configures interfaces to recover in ***30*** seconds when the cause is **bpduguard**.

```
switch(config)# errdisable recovery cause bpduguard interval 30
```

Example

Either of these commands revert the interval to the global value when the cause is **bpduguard**.

```
switch(config)# no errdisable recovery cause bpduguard interval
```

```
switch(config)# default errdisable recovery cause bpduguard interval
```

This command displays the status of the interfaces.

```
switch# show errdisable recovery
Errdisable Reason              Timer Status   Timer Interval
------------------------------ ---------------- --------------
   bpduguard                      Disabled                 30
   hitless-reload-down            Disabled                300
   lacp-no-portid                 Disabled                N/A
   lacp-rate-limit                Disabled                300
   license-enforce                Disabled                N/A
   link-flap                      Disabled                300
   no-internal-vlan               Disabled                300
   uplink-failure-detection       Disabled                300
```

11.4.5.5 Link Flap Monitoring

Link flap frequency is the quantity of link flaps (connection state changes) over a specified period. Excessive link flaps result in network stability issues, including spanning tree and routing recalculations. Link flaps are often caused by Layer 1 issues, such as a bad cable or duplex mismatch. Link flap monitoring specifies link flap thresholds and disables a port when a threshold is exceeded.

Link flap monitoring can be enabled on all interfaces through errdisable link flap commands or on individual interfaces with the link flap monitor.

11.4.5.5.1 Global Link Flap Monitor

Global link flap detection is configured through two global configuration mode commands:

- errdisable flap-setting cause link-flap configures the link-flap frequency that defines link-flap errors on an Ethernet interface.

- errdisable detect cause link-change enables the error-disabling of Ethernet interfaces that exceed the threshold link flap frequency.

Link-flap detection is enabled by default.

Example

These commands sets the link flap error criteria of ***15*** connection state changes over a ***30*** second period, then enables error detection on all interfaces.

```
switch(config)# errdisable flap-setting cause link-flap max-flaps 15 time 30
switch(config)# errdisable detect cause link-change
switch(config)#
```

11.4.5.5.2 Interface Link Flap Monitor

An interface is monitored for link flap errors with link flap profiles. A link flap profile specifies conditions that define a link-flap error. Profiles are assigned to Ethernet interfaces. Multiple profiles can be assigned to an interface to monitor a set of error conditions.

The global link flap monitor is used by interfaces that are not individually monitored for link flap errors.

Configuring Link Flap Profiles

Link flap profiles are configuration statements that define a link flap error in terms of these criteria:

- **flaps** Threshold number of interface state changes.
- **period** Interval when link flaps accumulate to trigger an error condition.
- **violations** Number of link flap errors (threshold exceeded over specified period).
- **intervals** Quantity of periods.

The monitor link-flap policy command places the switch in link-flap configuration mode for configuring link flap profiles and compiling a default-profile set. The profile max-flaps (Link Flap Configuration) command configures link flap profiles.

The default-profile set is a list of link-flap profiles that define error-disable criteria for interfaces where link flap monitoring is enabled but link-flap profiles are not assigned. The default-profile set may contain zero, one, or multiple profiles. When the default-profile set is empty, errdisable flap-setting cause link-flap specifies default error-disable criteria. When the default-profile set contains multiple profiles, the criteria is satisfied when conditions match any profile.

Example

These commands enter link flap configuration mode and create four link flap profiles.

```
switch(config)# monitor link-flap policy
switch(config-link-flap)# profile LF01 max-flaps 15 time 60
switch(config-link-flap)# profile LF02 max-flaps 10 time 30 violations 5
 intervals 10
switch(config-link-flap)# profile LF03 max-flaps 20 time 75 violations 2
 intervals 6
switch(config-link-flap)# profile LF04 max-flaps 30 time 100 violations 4
 intervals 7
switch(config-link-flap)# show active
monitor link-flap policy
   profile LF01 max-flaps 15 time 60 violations 1 intervals 1
   profile LF02 max-flaps 10 time 30 violations 5 intervals 10
   profile LF02 max-flaps 20 time 75 violations 2 intervals 6
   profile LF02 max-flaps 30 time 100 violations 4 intervals 7
switch(config-link-flap)#
```

The default-profiles command specifies the set of link-flap profiles that define error-disable criteria for interfaces where link flap monitoring is enabled without a link flap profile assignment. Entering a default-profile command replaces the current default-profile statement in ***running-config***.

The default-profile set may contain zero, one, or multiple profiles. When the default-profile set is empty, errdisable flap-setting cause link-flap specifies default error-disable criteria. When the default-profile set contains multiple profiles, error-disable criteria is satisfied when conditions match any profile. Multiple profiles are assigned to the default-profile set through a single default-profiles command.

Example

This command assigns configures ***LF01*** and ***LF02*** as the default-profile set.

```
switch(config)# monitor link-flap policy
switch(config-link-flap)# default-profiles LF01 LF02
switch(config-link-flap)# show active
monitor link-flap policy
   profile LF01 max-flaps 15 time 60 violations 1 intervals 1
   profile LF02 max-flaps 10 time 30 violations 5 intervals 10
   profile LF02 max-flaps 20 time 75 violations 2 intervals 6
   profile LF02 max-flaps 30 time 100 violations 4 intervals 7
   default-profiles LF01 LF02
switch(config-link-flap)#
```

Interface Link Flap Profile Assignments

Link flap monitoring is enabled on individual Ethernet interfaces and can optionally specify one or more profiles to define link-flap error-disabling criteria. When link flap monitoring is enabled on an interface, the link-flap conditions determine when the interface is error-disabled. Multiple profiles can be assigned to an interface to monitor a set of error conditions; a port is disabled when conditions match any of the profiles assigned to an interface.

The monitor link-flap profiles command controls link-flap monitoring on a configuration mode interface. The command provides these link flap detection options:

- **`monitor link-flap`** ***(no profiles listed)***: Interface detects link flaps using default-profile set criteria.
- **`monitor link-flap`** ***(at least one profile listed)***: Interface detects link flaps using listed profile criteria.
- **`default monitor link-flap`**: The interface uses global link flap monitor commands (Global Link Flap Monitor).
- **`no monitor link-flap`**: The interface does not detect link flaps.

Examples

- This command assigns ***LF03*** and ***LF04*** link flap profiles to ***interface ethernet 33***.

```
switch(config)# interface ethernet 33
switch(config-if-Et33)# monitor link-flap profiles LF03 LF04
switch(config-if-Et33)# show active
interface Ethernet33
   monitor link-flap profiles LF04 LF03
switch(config-if-Et33)#
```

- This command disables link-flap monitoring on ***interface ethernet 34***.

```
switch(config)# interface ethernet 34
switch(config-if-Et34)# no monitor link-flap
switch(config-if-Et34)# show active
interface Ethernet34
   no monitor link-flap
switch(config-if-Et34)#
```

- This command assigns the default-profile set to ***interface ethernet 35***.

```
switch(config)# interface ethernet 35
switch(config-if-Et35)# monitor link-flap
switch(config-if-Et35)# show active
interface Ethernet35
   monitor link-flap
switch(config-if-Et35)#
```

- This command configures ***interface ethernet 36*** to use the global link flap monitoring commands.

```
switch(config)# interface ethernet 36
switch(config-if-Et36)# default monitor link-flap
switch(config-if-Et36)# show active
interface Ethernet36
switch(config-if-Et36)#
```

11.4.5.6 Fabric Link Monitoring

Fabric link monitoring enables EOS to monitor low error rate errors on all fabric links for long durations, and automatically isolates fabric links on consistent error detection over an extended time interval. Isolated fabric links are restored when the error rate drops below a configured threshold.

The error rate over each configurable polling interval is derived by comparing the number of cells with CRC errors against the total number of received cells. Links are automatically isolated when the error rate is above the configured threshold for the configured consecutive number of polling intervals.

On an isolated fabric link, control cells (but not data cells) are sent. Once the error rate drops below a set threshold for the configured consecutive number of polling intervals, EOS revives the fabric link to continue sending data traffic.

11.4.5.6.1 Configuring Fabric Link Monitoring

Configuration mode commands globally enable and disable fabric link monitoring and syslog messages for the settings described below.

The **no platform sand monitor** command disables fabric link monitoring.

Generate Serdes Error Syslog

The platform sand monitor serdes error log command generates syslog fabric link monitoring for serdes error logging.

Example

This command enables the serdes error log for fabric link monitoring.

```
switch(config)# platform sand monitor serdes error log
switch(config)#
```

The following syslog messages are not enabled by default. Fabric link monitoring syslog is enabled by configuring the platform sand monitor serdes error log command.

Examples

- The following Syslog message is generated when a fabric link for serdes is automatically withdrawn:

```
%SAND-4-SERDES_WITHDRAWN_FROM_FABRIC: Serdes withdrawn from the switch
 fabric.
```

- Here is another instance where a Syslog message is generated when a fabric link is automatically withdrawn:

```
%SAND-4-SERDES_WITHDRAWN_FROM_FABRIC: Serdes Arad10/5-FabricSerdes-11
 withdrawn from the switch fabric.
```

- The following Syslog message is generated when a fabric link is restored:

```
%SAND-4-SERDES_RESTORED_TO_FABRIC: Serdes restored to the switch fabric.
```

- Here is another instance where a Syslog message is generated when a fabric link is restored:

```
%SAND-4-SERDES_RESTORED_TO_FABRIC: Serdes Arad10/5-FabricSerdes-11 restored
 to the switch fabric.
```

Generate Serdes Error Threshold

The platform sand monitor serdes error threshold command generates a fabric link monitoring serdes error threshold.

Example

This command monitors serdes error thresholds over the specified number of received cells, resulting in the isolation of a fabric link between ***200*** and ***30000*** received cells.

```
switch(config)# platform sand monitor serdes error threshold 200 30000
switch(config)#
```

Enable Serdes Poll Period

The platform sand monitor serdes poll period command sets the serdes poll period.

Example

This command changes the serdes polling period for fabric link monitoring to ***6*** seconds.

```
switch(config)# platform sand monitor serdes poll period 6
switch(config)#
```

Monitor Serdes Poll Threshold Isolation

The platform sand monitor serdes poll threshold isolation command sets and enables fabric link monitoring for serdes poll threshold isolation.

Example

This command changes the number of consecutive polls in which the threshold needs to be detected to isolate a link. In this case the number is ***5*** consecutive polls.

```
switch(config)# platform sand monitor serdes poll threshold isolation 5
switch(config)#
```

Monitor Serdes Poll Threshold Recovery

The platform sand monitor serdes poll threshold recovery command sets and enables fabric link monitoring for serdes poll threshold recovery.

Example

This command changes the number of consecutive serdes polls used for threshold recovery to ***6*** seconds.

```
switch(config)# platform sand monitor serdes poll threshold recovery 6
switch(config)#
```

Show Fabric Monitoring Health

The show fabric monitoring health command displays the fabric monitoring connected state status with isolated links.

Example

When fabric links are isolated, their connected state status is shown with isolated links.

```
switch(config)# show platform sand health
Fabric serdes isolated by fabric monitoring: (36 total)

Arad5/0 serdes [0-1, 10-19, 2, 20-29, 3, 30-35, 4-9]

Top fabric serdes list by number of times isolated by monitoring:
Arad5/0 serdes 0: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 1: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 10: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 11: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 12: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 13: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 14: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 15: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 16: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 17: 1 (last occurred: 0:01:04 ago)

switch(config)#
```

11.4.5.7 Rapid Automated Indication of Link-Loss

Rapid Automated Indication of Link-Loss (RAIL) is a software feature that reduces the wait time of applications on hosts that are blocked due to a failed link. When a link goes down because of link-flapping or the unavailability of a directly connected server, the switch drops all traffic to servers whose next-hop destination was learned on the port connected to the link. Applications that drive the traffic (clients on source hosts) are blocked because of the dropped edge-switch traffic. Connection timeout varies by application and is usually measured in seconds or minutes.

RAIL is functional on a switch if it is routing-enabled and available for servers that set the switch as the default router.

11.4.5.7.1 RAIL Method

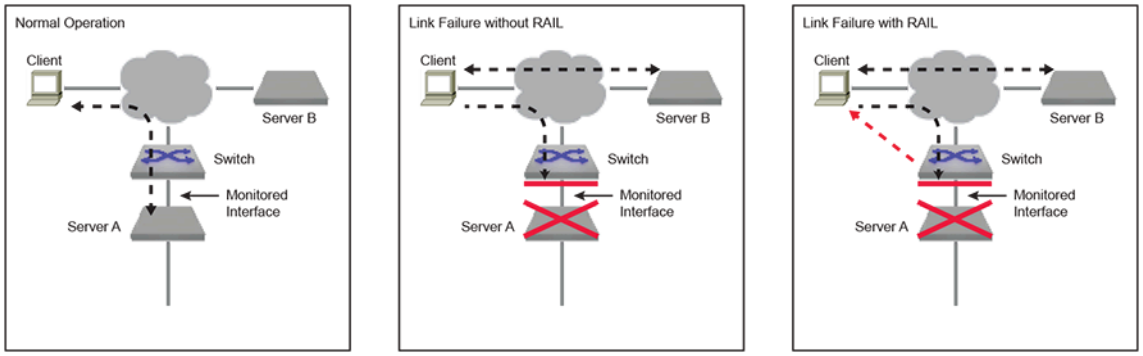
When a link monitored by RAIL goes down, the switch performs these steps for servers that the switch proxies:

1. IP addresses of servers on the failed link are extracted from ARP cache. The interface that accesses the server is determined by searching for the MAC address in the hardware MAC address tables.
2. Upon link shutdown, a dynamic MAC entry is added in the MAC address table for each server that was learned on the failed interface. Each new entry lists its interface as **CPU**.

The figure below titled **RAIL Scenarios** depicts three switch-server scenarios: link is up, link is down with RAIL disabled, and link is down with RAIL enabled. A failed link with RAIL enabled results in these behaviors:

1. All ingress packets whose destination MAC address matches an address added to the MAC address table are sent to the CPU.
2. For packets scheduled to be forwarded to the source address, the switch sends one of the following, based on the type of received segment:
   - **TCP**: TCP RST segment to the source IP address and port.
   - **UDP**: ICMP unreachable segment to the source IP address and port.
3. The client closes the socket associated with the transmitted segment and notifies the application. The application reacts immediately instead of maintaining the block until connection timeout expiry.

Figure 11-5: RAIL Scenarios



11.4.5.7.2 RAIL Implementation

RAIL defines a state machine that manages the RAIL activity level relative to a specified server. The state machine consists of four states:

- **Up**: Transitions to this state from ***Inactive*** when ARP and MAC entries are added for the server.
- **Proxying**: Transitions to this state from ***Up*** when Link Down is detected and RAIL proxying is enabled. The switch is a proxy for messages to the server.
- **Down**: Transitions to this state from ***Up*** when Link Down is detected and RAIL proxying is not enabled. Messages from the client remain unanswered and the application recovers only after timeout expiry.
- **Inactive**: Transitions to this state upon any of the following conditions:
  - Server's MAC address or ARP entry is deleted (from any state).
  - Proxy timeout expiry (from ***Proxying*** state).
  - Link down timeout expiry (from ***Down*** state).

11.4.5.7.3 RAIL Configuration

Server-failure configuration mode commands globally enable RAIL and configure RAIL parameters. RAIL is functional on individual interfaces only when it is globally enabled and enabled on the interface. RAIL monitors an interface for link errors when RAIL is globally enabled and enabled on the interface.

Entering Server-failure Configuration Mode

The monitor server-failure command places the switch in server-failure configuration mode. The **`exit`** command returns the switch to global configuration mode. Server-failure mode is not a group change mode; ***running-config*** is changed when commands are entered and not affected by exiting the mode.

The **`no monitor server-failure`** deletes all server-failure mode commands from ***running-config***.

Examples

- These commands place the switch in the ***server-failure*** configuration mode.

```
switch(config)# monitor server-failure
switch(config-server-failure)#
```

- This command deletes all server-failure configuration mode commands from ***running-config***.

```
switch(config)# no monitor server-failure
switch(config)#
```

Enabling RAIL on the Switch

RAIL is disabled by default and is enabled by no shutdown (server-failure configuration mode). The **`shutdown`** command disables RAIL without removing RAIL commands from ***running-config***.

Examples

- These commands enable RAIL globally.

```
switch(config)# monitor server
switch(config-server-failure)# no shutdown
switch(config-server-failure)# show active
monitor server-failure
   no shutdown
switch(config-server-failure)#
```

- This command disables RAIL globally.

```
switch(config-server-failure)# shutdown
switch(config-server-failure)#
```

Enabling Proxy Mode

The proxy (server-failure configuration mode) command sets the RAIL proxy setting to ***enabled*** and specifies the interval that RAIL responds to messages sent to servers on failed links. The proxy timeout is measured individually for each server whose link has failed. The switch enters RAIL proxy state only when the proxy setting is enabled.

When RAIL is enabled but the proxy setting is disabled, the switch maintains a list of unavailable servers without responding to messages sent to the servers. The RAIL proxy setting is ***disabled*** by default. When RAIL proxy is enabled, the default period is three minutes.

The **`no proxy`** and **`default proxy`** commands return the RAIL proxy setting to ***disabled***. The **`no proxy lifetime`** and **`default proxy lifetime`** commands set the proxy timeout to its default of three minutes if the RAIL proxy setting is **enabled**. The lifetime commands have no effect if RAIL proxy is **disabled**.

Examples

- These commands enable the RAIL proxy and sets the proxy timeout period of ***10*** minutes.

```
switch(config)# monitor server
switch(config-server-failure)# proxy lifetime 10
switch(config-server-failure)# show active
monitor server-failure
   proxy lifetime 10
switch(config-server-failure)#
```

- This command sets the proxy timeout period to its default value of ***3*** minutes.

```
switch(config-server-failure)# no proxy lifetime
switch(config-server-failure)# show active
monitor server-failure
   proxy
switch(config-server-failure)#
```

- This command disables the RAIL proxy.

```
switch(config-server-failure)# no proxy
switch(config-server-failure)# show active
switch(config-server-failure)#
```

Selecting Networks to Monitor

The network (server-failure configuration mode) command specifies the IPv4 network space that Rapid Automated Indication of Link-Loss (RAIL) monitors for failed links to connected servers. ***Running-config*** can contain multiple **network** statements, allowing RAIL to monitor multiple disjoint network spaces.

When a server on the specified network is blocked because of a failed Ethernet or port channel link, the switch becomes a proxy for the unavailable server and responds with **TCP RST** or ***ICMP Unreachable*** segments to devices sending packets to the unavailable server.

Example

These commands specify two IPv4 network spaces that RAIL monitors for server failures.

```
switch(config)# monitor server
switch(config-server-failure)# network 10.1.1.0/24
switch(config-server-failure)# network 10.2.1.96/28
switch(config-server-failure)# show active
monitor server-failure
   network 10.2.1.96/28
   network 10.1.1.0/24
switch(config-server-failure)#
```

Enabling RAIL on an Interface

RAIL monitors an interface for link errors only when RAIL is globally enabled and enabled for the interface. The monitor server-failure link command enables RAIL on the configuration mode interface. Configuration settings are effective for all Ethernet and port channel interfaces that enable RAIL.

Example

These commands enable RAIL on ***port channel interface 100***.

```
switch(config)# interface port-channel 100
switch(config-if-Po100)# monitor server-failure link
switch(config-if-Po100)# show active
interface Port-Channel100
   monitor server-failure link
switch(config-if-Po100)#
```

11.4.5.7.4 Displaying RAIL Status

The switch provides commands to display RAIL configuration and status information:

Displaying RAIL Configuration settings

The show monitor server-failure command displays Rapid Automated Indication of Link-Loss (RAIL) configuration settings and the number of servers on each monitored network.

Example

This command displays RAIL configuration status and lists the number of servers that are on each monitored network.

```
switch> show monitor server-failure
Server-failure monitor is enabled
Proxy service: disabled
Networks being monitored: 3
   10.2.1.96/28      : 0 servers
   10.1.1.0/24       : 0 servers
   10.3.0.0/16       : 3 servers switch>
```

Displaying RAIL History for All Connected Servers

The show monitor server-failure history command displays the time of all link failures detected by Rapid Automated Indication of Link-Loss (RAIL) and includes the interface name for each failure.

Example

This command displays the link failure history from the time RAIL is instantiated on the switch.

```
switch> show monitor server-failure history
Total server failures: 4

Server IP   Server MAC             Interface           Last Failed
----------- -----------------      -----------         -------------------
10.1.67.92  01:22:ab:cd:ee:ff      Ethernet17          2013-02-02 11:26:22
44.11.11.7  ad:3e:5f:dd:64:cf      Ethernet23          2013-02-10 00:07:56
10.1.1.1    01:22:df:42:78:cd      Port-Channel6       2013-02-09 19:36:09
10.1.8.13   01:33:df:ee:39:91      Port-Channel5       2013-02-10 00:03:39
 switch>
```

Displaying Server Configuration and Status

The show monitor server-failure servers command displays status and configuration data about each server that RAIL monitors. The display format depends on the parameter specified by the command:

Examples

- This command displays RAIL information for the server at IP address ***10.11.11.7***.

```
switch> show monitor server-failure servers 10.11.11.7
Server information:
Server Ip Address        : 10.11.11.7
MAC Address              : ad:3e:5f:dd:64:cf
Current state            : down
Interface                : Ethernet23
Last Discovered          : 2013-01-06 06:47:39
Last Failed              : 2013-02-10 00:07:56
Last Proxied             : 2013-02-10 00:08:33
Last Inactive            : 2013-02-09 23:52:21
Number of times failed   : 3
Number of times proxied  : 1
Number of times inactive : 18 switch>
```

- This command displays RAIL information for the all servers on configured interfaces.
-

```
switch> show monitor server-failure servers all
```

```
Total servers monitored: 5

Server IP   Server MAC         Interface       State Last Failed
----------  -----------------  --------------  ---------  -----------
10.1.67.92  01:22:ab:cd:ee:ff  Ethernet17      inactive   7 days, 12:47:48 ago
44.11.11.7  ad:3e:5f:dd:64:cf  Ethernet23      down       0:06:14 ago
10.1.1.1    01:22:df:42:78:cd  Port-Channel6   up         4:38:01 ago
10.1.8.13   01:33:df:ee:39:91  Port-Channel5   proxying   0:10:31 ago
132.23.23.1 00:11:aa:bb:32:ad  Ethernet1       up         never switch>
```

11.4.6 PHY test pattern CLI

Use the Ethernet Physical Layer (PHY) test pattern CLI to check the quality of the physical layer for an Ethernet interface. You can do this by generating a specific test pattern to a peer, and having the peer check the test pattern that is received, and vice versa. Because the test pattern is a well-known sequence of bits, the peer can check that the pattern received matches this well-known sequence; any difference is a bit error introduced by the peculiarities of the physical layer. The quality of the link is determined based on the acceptable bit errors, as published by the hardware vendors.

To enable the test pattern generator, configure a specific test pattern on the transmitter side of an interface. The test pattern checker is enabled by configuring the test pattern to be checked on the receiver side of the interface. PRBS is the test pattern supported by EOS.

Note: Physical links are bidirectional; to test both directions, the generator and checker both need to be enabled on both sides of the link. Both directions can be tested simultaneously or separately. The order of testing does not matter.

11.4.6.1 Configuration

You can configure a test pattern is configured using the **`phy diag`** interface configuration mode command.

1. Enter interface configuration mode, entering the targeted interface name.

```
switch(config)# interface <interfaceName>
```

2. Enable a test pattern on an interface using the **`phy diag`** command. You can select the transmitter or the receiver. To display the available interfaces, select **test pattern ?**.

```
switch(config-if)# phy diag [ transmitter | receiver ] test pattern ?
  PRBS11  Configure the PRBS11 test pattern
  PRBS15  Configure the PRBS15 test pattern
  PRBS23  Configure the PRBS23 test pattern
  PRBS31  Configure the PRBS31 test pattern
  PRBS49  Configure the PRBS49 test pattern
  PRBS58  Configure the PRBS58 test pattern
  PRBS7   Configure the PRBS7 test pattern
  PRBS9   Configure the PRBS9 test pattern
```

3. To disable a test pattern on an interface, enter the following command. You can select the tranmitter or the receiver, as well as the selected named test pattern.

```
switch(config-if)# no phy diag [transmitter|receiver] test
 pattern TestPattern
```

4. By default, a test pattern is disabled.

```
switch(config-if)# default phy diag [transmitter|receiver] test pattern
```

5. The following command clears the recorded test pattern status data for all the interfaces. Upon running the command, all the counter values are set to ***0*** and link states are marked as **`not locked`**.

```
switch# clear phy diag test pattern
```

11.4.6.2 Show Commands

To display the configured and operational test pattern, as well as the test patterns available for an interface, use the **show interfaces** command.

In the following example, interfaces ***ethernet 36/1*** and ***ethernet 31/1*** are selected for display. The user-configured test pattern is displayed under the **Configured** column, which is divided based on transmitter and receiver configuration. The currently operational test pattern is displayed under the **Operational** column. The **Available** column lists the test patterns available for the interface.

```
switch# show interfaces ethernet 26/1,31/1 phy diag test pattern
                    Configured       Operational
Interface        Transmit Receive Transmit Receive Available

---------------- -------- ------- -------- ------- -------------
-----------
Ethernet26/1     PRBS15   PRBS15  PRBS15   PRBS15  PRBS 7,9,11,15,23,
31,58
Ethernet31/1     PRBS7    PRBS31  PRBS7    PRBS31  PRBS 7,9,11,15,23,
31,58
```

Use the **show interfaces [<interface range>] phy detail** command to display the operational test pattern for an interface. In the example below, the **Test pattern** field will not be available, on disabling the test pattern.

Note: This command is not available on DCS-7060PX4 and DCS-7060DX4.

```
switch# show interfaces ethernet 26/1 phy detail | i Test pattern
  Test pattern                enabled
switch# show interfaces ethernet 31/1 phy detail | i Test pattern
  Test pattern                enabled
```

Use the **show interfaces [<interface range>] phy diag test pattern counters** to display test pattern link state and error information.

Available error information:

- **Link state:** whether or not the checker locked on to the configured test pattern.
- **Bit Errors:** the accumulated number of bit errors.
- **Largest Burst:** the largest burst of errors that occurred.
- **Burst Count:** the number of occurrences of errors.
- **Last Error Time:** the last time an error has occurred, ‘never’ if no errors have occurred.

```
switch# show interfaces ethernet 26/1,31/1 phy diag test pattern
 counters
Current System Time: Wed May 30 22:24:32 2018
                                                  Largest    Burst
Interface        Lane  Link State  Bit Errors   Burst      Count
 Last Error Time
---------------- ----- ----------- ------------ ---------- --------
 -----------------
```

```
Ethernet26/1     0     locked      409266       409266     1
 0:21:27 ago
Ethernet26/1     1     locked      347084       347084     1
 0:21:27 ago
Ethernet26/1     2     locked      420681       420681     1
 0:21:27 ago
Ethernet26/1     3     locked      392969       392969     1
 0:21:27 ago
Ethernet31/1     0     not locked  1417655      651822     3
 0:03:20 ago
Ethernet31/1     1     not locked  1782238      736819     3
 0:03:20 ago
Ethernet31/1     2     not locked  1760538      866185     3
 0:03:20 ago
Ethernet31/1     3     not locked  1817413      923941     3
 0:03:20 ago
```

Use the **`show interfaces [<interface range>] phy diag test pattern counters`** to display the lock state of an interface along with a detailed information on the recorded bit errors.

Available detailed information:

- **Last clear:** the time when the test pattern results were last cleared.
- **Operational test pattern:** the test pattern operational at the receiver side.
- **Bit rate:** the transmission bit rate.
- **Lock state:** the current lock status, number of times it changed and the last time the lock status got changed.
  - **locked:** receiver is able to lock on to the incoming test pattern.
  - **not locked:** receiver is not able to lock on to the incoming test pattern.
- **Largest burst:** the largest burst of errors that occurred.
- **Bit errors*:** the accumulated number of errors, number of occurrences of errors, and last time errors were captured. The ***** suffix, indicating that data may not be accurate due to loss of lock, is applied if the current lock status is not locked or if the lock status has changed more than once. This suffix is cleared when the test pattern status data is cleared via the CLI listed above.
- **Total Bits:** the total bits received.
- **Bit error rate (BER)*:**the ratio of captured bit errors to the total bit received. The ***** suffix, indicating that data may not be accurate due to loss of lock, is applied if the current lock status is not locked or if the lock status has changed more than once. This suffix is cleared when the test pattern status data is cleared via the CLI listed above.
- **Bit errors since last lock:** the accumulated number of errors since last time lock was gained.
- **Total bits since last lock:** the total bits received since last lock.
- **BER since last lock:** the ratio of captured bit errors to the total bit received since last lock.

```
switch# show interfaces ethernet 26/1,31/1 phy diag test pattern
 counters detail
*: Data may not be accurate due to loss of lock.

Current System Time:  Wed May 30 23:36:34 2018
Ethernet26/1
  Last clear                    1:33:29 ago
  Operational test pattern      PRBS15
                                Current State     Changes      Last
 Change
                                -------------     -------
 -----------
```

```
  Lane 0
     Bit rate                    25.781 Gbps
     Lock state                  locked                 1
 1:33:28 ago
     Largest burst               409266
     Bit errors                  409266                 1
 1:33:28 ago
     Total bits                  144,607.648 Gb
     Bit error rate              2.83E-09
     Bit errors since last lock  409266
     Total bits since last lock  161,542.986 Gb
     BER since last lock         2.53E-09
  Lane 1
     Bit rate                    25.781 Gbps
     Lock state                  locked                 1
 1:33:28 ago
     Largest burst               347084
     Bit errors                  347084                 1
 1:33:28 ago
     Total bits                  144,607.668 Gb
     Bit error rate              2.40E-09
     Bit errors since last lock  347084
     Total bits since last lock  161,543.006 Gb
     BER since last lock         2.15E-09
  Lane 2
     Bit rate                    25.781 Gbps
     Lock state                  locked                 1
 1:33:28 ago
     Largest burst               420681
     Bit errors                  420681                 1
 1:33:28 ago
     Total bits                  144,607.658 Gb
     Bit error rate              2.91E-09
     Bit errors since last lock  420681
     Total bits since last lock  161,542.996 Gb
     BER since last lock         2.60E-09
  Lane 3
     Bit rate                    25.781 Gbps
     Lock state                  locked                 1
 1:33:28 ago
     Largest burst               392969
     Bit errors                  392969                 1
 1:33:28 ago
     Total bits                  144,607.678 Gb
     Bit error rate              2.72E-09
     Bit errors since last lock  392969
     Total bits since last lock  161,543.016 Gb
     BER since last lock         2.43E-09

Ethernet31/1
  Last clear                     1:33:29 ago
  Operational test pattern       PRBS31
                                 Current State     Changes       Last
 Change
                                 -------------     -------
 -----------
  Lane 0
     Bit rate                    25.781 Gbps
     Lock state                  not locked             3
 1:15:22 ago
     Largest burst               651822
     Bit errors                  1417655*               3
 1:15:22 ago
     Total bits                  144,626.220 Gb
```

```
 Bit error rate              > 9.80E-09*
    Bit errors since last lock  765833*
    Total bits since last lock  144,471.763 Gb
    BER since last lock         > 5.30E-09*
 Lane 1
    Bit rate                    25.781 Gbps
    Lock state                  not locked              3
1:15:22 ago
    Largest burst               736819
    Bit errors                  1782238*                3
1:15:22 ago
    Total bits                  144,626.240 Gb
    Bit error rate              > 1.23E-08*
    Bit errors since last lock  1147126*
    Total bits since last lock  144,471.783 Gb
    BER since last lock         > 7.94E-09*
 Lane 2
    Bit rate                    25.781 Gbps
    Lock state                  not locked              3
1:15:22 ago
    Largest burst               866185
    Bit errors                  1760538*                3
1:15:22 ago
    Total bits                  144,626.230 Gb
    Bit error rate              > 1.22E-08*
    Bit errors since last lock  894353*
    Total bits since last lock  144,471.773 Gb
    BER since last lock         > 6.19E-09*
 Lane 3
    Bit rate                    25.781 Gbps
    Lock state                  not locked              3
1:15:22 ago
    Largest burst               923941
    Bit errors                  1817413*                3
1:15:22 ago
    Total bits                  144,626.250 Gb
    Bit error rate              > 1.26E-08*
    Bit errors since last lock  893472*
    Total bits since last lock  144,471.793 Gb
    BER since last lock         > 6.18E-09*
```

11.4.6.3 Bit Error Rate (BER)

Bit error rate is the ratio of the recorded bit errors to the total bits received for the duration of the test run. To achieve a reliable transmission, BER should be relatively small. As per IEEE 802.3 standard, the minimum BER requirement for Ethernet links is 1E-12. Therefore, links with BER lower than 1E-12 are to be considered reliable.

The BER reported by the **test pattern** CLI is the pre-FEC (Forward Error Correction) BER. For links that have FEC enabled, it is expected to see a higher BER, in the range of ***1E-4*** to ***1E-8***, because they are calculated before FEC is applied on the link. Based on the type of FEC applied on the link, these errors could get corrected to achieve the minimum BER requirement of 1E-12 or less.

11.4.6.4 Limitations

The configuration of test patterns is supported only on a few types of ports. The available test patterns that may be configured on an interface are found in the **`Available`** field of the **`show interfaces phy diag test pattern`** CLI command.

The test pattern CLI calculates only pre-FEC BER.

If one end of the system is from another vendor, consult the vendor's documentation for the equivalent command(s) to achieve the appropriate behavior.

11.4.7 Data Transfer Commands

Control Plane and Data Plane Commands

- ip access-group (Control Plane mode)
- show switch forwarding-mode
- switch forwarding-mode
- system control-plane

Errdisable Commands

- errdisable detect cause link-change
- errdisable flap-setting cause link-flap
- errdisable recovery cause
- errdisable recovery interval
- show errdisable recovery

Fabric Link Monitoring Commands

- platform sand monitor serdes error log
- platform sand monitor serdes error threshold
- platform sand monitor serdes poll period
- platform sand monitor serdes poll threshold isolation
- platform sand monitor serdes poll threshold recovery
- show fabric monitoring health
- show platform trident mirroring

RAIL Commands

- clear server-failure servers inactive
- monitor server-failure
- monitor server-failure link
- network (server-failure configuration mode)
- proxy (server-failure configuration mode)
- show monitor server-failure
- show monitor server-failure history
- show monitor server-failure servers
- shutdown (server-failure configuration mode)

Link Flap Monitor Commands

- default-profiles
- monitor link-flap policy
- monitor link-flap profiles
- profile max-flaps (Link Flap Configuration)

MAC Address Table Commands

- clear mac address-table dynamic
- mac address-table aging-time
- mac address-table static

- show bridge mac-address-table aging timeout
- show mac address-table
- show mac address-table count
- show mac address-table mlag-peer
- show mac address-table multicast
- show mac address-table multicast brief
- switchport mac address learning

Port Configuration Commands

- clear counters
- description
- interface loopback
- load interval
- mac address learning
- mtu
- phy diag
- show interfaces
- show interfaces description
- show interfaces phy diag
- show port-channel load-balance
- switchport
- switchport default mode access
- switchport default mode routed

Port Mirroring Commands

- monitor session destination
- monitor session destination cpu
- monitor session forwarding-drop
- monitor session ip access-group
- monitor session source
- monitor session source ip access-group
- monitor session truncate
- no monitor session
- show monitor session

Port Security Commands

- show port-security
- show port-security interface
- show port-security mac-address
- switchport port-security
- switchport port-security mac-address maximum
- switchport port-security violation

Storm Control Commands

- show storm-control
- storm-control

Tracking Commands

- links minimum
- link tracking group
- link tracking group (interface)
- show link tracking group
- show track
- track
- traffic-loopback

11.4.7.1 clear counters

The **`clear counters`** command resets the counters to zero for the specified interfaces. The command provides the following options:

- ***No parameter*:** When no option is selected, the counters are reset on the switch.
- ***Session parameter*:** The command resets the counters in software for the current CLI session, establishing a baseline upon which subsequent **`show interfaces`** or **`show interfaces counters`** commands are relative. Counters are not affected for other CLI sessions.

Note: The **`clear counters`** command (and other commands that reset counters to zero) do not reset SNMP counters (such as IF-MIB::ifInOctets). As specified in ***RFC 2578***, ***sections 7.1.6*** and ***7.1.10***, a single value of a counter in SNMP has no information content. Instead, meaningful information is given by the difference between two separate fetches of a particular counter. SNMP counters automatically reset to ***0*** when they reach their maximum values.

Command Mode

Privileged EXEC

Command Syntax

`clear counters` [**INTERFACE**][**SCOPE**]

Parameters

- **INTERFACE** Interface type and number. Options include:
  - ***no parameter*** Display information for all interfaces.
  - **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
  - **loopback *l_range*** Loopback interface specified by ***l_range***.
  - **management *m_range*** Management interface range specified by ***m_range***.
  - **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
  - **vlan *v_range*** VLAN interface range specified by ***v_range***.
  - **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.
- Valid ***e_range***, ***l_range***, ***m_range***, ***p_range***, ***v_range***, and ***vx_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **SCOPE** Duration of the reset results. Options include:
  - ***no parameter*** counters are cleared on the switch.
  - **session** counters are reset only for the current session.

Example

These commands display interface counters, clear the counters, then display the counters again.

```
switch# show interfaces ethernet 1
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.7302.2fff (bia 001c.7302.2fff)
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 minutes input rate 301 bps (0.0% with framing), 0 packets/sec
```

```
  5 minutes output rate 0 bps (0.0% with framing), 0 packets/sec
     2285370854005 packets input, 225028582832583 bytes
     Received 29769609741 broadcasts, 3073437605 multicast
     113 runts, 1 giants
     118 input errors, 117 CRC, 0 alignment, 18 symbol
     27511409 PAUSE input
     335031607678 packets output, 27845413138330 bytes
     Sent 14282316688 broadcasts, 54045824072 multicast
     108 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output

switch# show interfaces ethernet 1-5 counters
Port                InOctets     InUcastPkts     InMcastPkts     InBcastPkts
Et1           225028582833321   2252527806659      3073437611     29769609741
Et2            20706544058626    121703943738       7619026884     43349412335
Et3            17473231954010     84335312119      18987530444     25136247381
Et4            21909861242537    119410161405       3792251718     48470646199
Et5                         0               0                0               0

Port               OutOctets    OutUcastPkts    OutMcastPkts    OutBcastPkts
Et1            27845413138330    266703466918     54045824072     14282316688
Et2            39581155181762    384838173282     34879250675     15500233246
Et3            25684397682539    256695349801     25193361878     16244203611
Et4           428040746505736   2285287022532     44408620604     19503612572
Et5                         0               0                0               0

switch# clear counters session

switch# show interfaces ethernet 1
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.7302.2fff (bia 001c.7302.2fff)
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters 0:00:10 ago
  5 minutes input rate 322 bps (0.0% with framing), 0 packets/sec
  5 minutes output rate 0 bps (0.0% with framing), 0 packets/sec
     6 packets input, 835 bytes
     Received 0 broadcasts, 6 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     0 packets output, 0 bytes
     Sent 0 broadcasts, 0 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output

switch# show interfaces ethernet 1-5 counters
Port                 InOctets     InUcastPkts     InMcastPkts     InBcastPkts
Et1                      1204               0               9               0
Et2                      1204               0               9               0
Et3                      1204               0               9               0
Et4                      1204               0               9               0
Et5                         0               0               0               0

Port                OutOctets    OutUcastPkts    OutMcastPkts    OutBcastPkts
Et1                         0               0               0               0
Et2                         0               0               0               0
Et3                         0               0               0               0
Et4                         0               0               0               0
Et5                         0               0               0               0 switch#
```

11.4.7.2 clear mac address-table dynamic

The **clear mac address-table dynamic** command removes specified dynamic entries from the MAC address table. Entries are identified by their VLAN and Layer 2 (Ethernet or port channel) interface.

- To remove a specific entry, include its VLAN and interface in the command.
- To remove all dynamic entries for a VLAN, do not specify an interface.
- To remove all dynamic entries for an interface, do not specify a VLAN.
- To remove all dynamic entries, do not specify a VLAN or an interface.

Command Mode

Privileged EXEC

Command Syntax

`clear mac address-table dynamic` [**VLANS**][**INTERFACE**]

Parameters

- **VLANS** Table entries are cleared for specified VLANs. Options include:
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** VLAN specified by ***v_num***.
- **INTERFACE** Table entries are cleared for specified interfaces. Options include:
  - ***no parameter*** all Ethernet and port channel interfaces.
  - **interface ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
  - **interface port-channel *p_range*** port channel interfaces specified by ***p_range***.
  - **vxlan *vx_range*** VXLAN interfaces specified by ***vx_range***.

  Valid ***range*** formats include number, range, or comma-delimited list of numbers and ranges.

Example

This command clears all dynamic mac address table entries for ***port channel 5*** on ***vlan 34***.

```
switch# clear mac address-table dynamic vlan 34 interface port-channel 5
switch#
```

11.4.7.3 clear server-failure servers inactive

The **`clear server-failure servers inactive`** command removes all inactive server entries from the server failed history list. The switch maintains this list, even after a server's ARP entry is removed, to maintain a list of servers that are connected to the switch and log the most recent time of the failure of the link that connects the switch to the server.

Command Mode

Privileged EXEC

Command Syntax

`clear server-failure servers inactive`

Related Command

show monitor server-failure history

Example

This command clears the inactive servers from the server failed history list.

```
switch# clear server-failure servers inactive
switch#
```

11.4.7.4 default-profiles

The **`default-profiles`** command specifies the set of link-flap profiles that define error-disable criteria for interfaces where link flap monitoring is enabled without a link flap profile assignment. Entering a **`default-profile`** command replaces the current default-profile statement in ***running-config***.

The default-profile set may contain zero, one, or multiple profiles. When the default-profile set is empty, errdisable flap-setting cause link-flap specifies default error-disable criteria. When the default-profile set contains multiple profiles, error-disable criteria is satisfied when conditions match any profile. Multiple profiles are assigned to the default-profile set through a single **`default-profiles`** command.

The **no default-profiles** and **default default-profiles** commands restore the empty default-profile set by deleting the **default-profiles** command from ***running-config***.

Command Mode

Link-flap Configuration

Command Syntax

default-profiles [**LF_PROFILES**]

no default-profiles

default default-profiles

Parameters

LF_PROFILES Name of link-flap profiles assigned to default profile set. Parameter may contain zero, one, or multiple link-flap profile names:

- ***no parameter*** default-profile set is empty.
- ***profile*** name of single link-flap profile.
- ***profile_1 profile_2 ... profile_N*** list of link-flap profile names.

Related Commands

- monitor link-flap policy places the switch in ***link-flap-profiles*** configuration mode.
- profile max-flaps (Link Flap Configuration) configures link flap profiles.

Guidelines

The errdisable flap-setting cause link-flap statement is also configurable through the profile max-flaps (Link Flap Configuration) command.

Example

This command assigns configures ***LF01*** and ***LF02*** as the default-profile set.

```
switch(config)# monitor link-flap policy
switch(config-link-flap)# default-profiles LF01 LF02
switch(config-link-flap)# show active
monitor link-flap policy
   profile LF01 max-flaps 15 time 60 violations 1 intervals 1
   profile LF02 max-flaps 10 time 30 violations 5 intervals 10
   profile LF03 max-flaps 25 time 100 violations 2 intervals 12
   profile LF04 max-flaps 5 time 15 violations 1 intervals 3
   default-profiles LF01 LF02
switch(config-link-flap)#
```

11.4.7.5 description

The **description** command adds comment text for the configuration mode interface. The text provides information about the interface and has no effect on interface functions. The show interfaces description command displays interface description text.

The **no description** command removes the description text for the configuration mode interface from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Interface-VXLAN Configuration

Command Syntax

`description` ***label_text***

`no description`

`default description`

Parameters

label_text character string assigned to description attribute.

Example

These commands add description text to ***interface ethernet 23***, then displays the text through the show interfaces description command.

```
switch(config)# interface ethernet 23
switch(config-if-Et23)# description external line
switch(config-if-Et23)# show interfaces ethernet 23 description
Interface              Status     Protocol    Description
Et23                   up         up          external line
```

11.4.7.6 errdisable detect cause link-change

The **`errdisable detect cause link-change`** command enables the error-disabling of Ethernet interfaces when the switch detects a link flap error on the interface. The errdisable flap-setting cause link-flap command defines a link flap error in terms of the frequency of connection state changes.

The switch places an interface in **error-disabled** state when it detects an error on the interface. **Error-disabled** is an operational state that is similar to link-down state. To re-enable an error-disabled interface, enter **`shutdown`** and **`no shutdown`** command in the configuration mode for the interface.

By default, link flap detection is enabled. The **`no errdisable detect cause link-change`**command disables the triggering of error-disable actions. The **`errdisable detect cause link-change`** and **`default errdisable detect cause link-change`** commands enable the triggering of error-disable actions by removing the **`no errdisable detect cause link-change`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`errdisable detect cause link-change`

`no errdisable detect cause link-change`

`default errdisable detect cause link-change`

Examples

- This command disables error detection on the switch.

```
switch(config)# no errdisable detect cause link-change
switch(config)#
```

- These commands sets the link flap error criteria of ***15*** connection state changes over a ***30*** second period, then enables error detection on the switch.

```
switch(config)# errdisable flap-setting cause link-flap max-flaps 15 time 30
switch(config)# errdisable detect cause link-change
```

```
switch(config)#
```

11.4.7.7 errdisable flap-setting cause link-flap

The **errdisable flap-setting cause link-flap** command configures the link-flap frequency that defines an link-flap error on an Ethernet interface. The errdisable detect cause link-change command uses this criteria to trigger an error-disable action.

The link-flap frequency is defined by the quantity of link flaps (connection state changes) over a specified period. The default settings are five link flaps and ten seconds.

The **no errdisable flap-setting cause link-flap** and **default errdisable flap-setting cause link-flap** commands restore the default link flap cause settings by removing the **errdisable flap-setting cause link-flap** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

errdisable flap-setting cause link-flap max-flaps *quantity* time ***period***

no errdisable flap-setting cause link-flap

default errdisable flap-setting cause link-flap

Parameters

- ***quantity*** Number of link flaps. Value ranges from ***1*** to ***100***. Default value is ***5***.
- ***period*** Interval over which link flaps accumulate to trigger an error condition (seconds). Value ranges from ***1*** to ***1800***. Default value is ***10***.

Example

This command sets the link flap error criteria of ***15*** connection state changes over ***30*** second periods.

```
switch(config)# errdisable flap-setting cause link-flap max-flaps 15 time 30
switch(config)#
```

11.4.7.8 errdisable recovery cause

The **errdisable recovery cause** command enables the automated recovery of error-disabled Ethernet interfaces. An interface that is disabled as a result of a specified condition attempts normal operation after a specified interval. When the disabling condition persists, recovered interfaces eventually return to the error-disabled state.

When automated recovery is not enabled, interfaces are recovered manually by entering **shutdown** and **no shutdown** from the interface's configuration mode.

Running-config can simultaneously store **errdisable recovery cause** statements for each error-disable condition. By default, error-disable recovery is disabled for all conditions.

The **no errdisable recovery cause** and **default errdisable recovery cause** commands disable automated recovery for interfaces disabled by the specified condition by removing the corresponding **errdisable recovery cause** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

errdisable recovery cause CONDITION

`no errdisable recovery cause` CONDITION

`default errdisable recovery cause` CONDITION

Parameters

CONDITION Disabling condition for which command automates recovery. Options include:

- **arp-inspection**
- **bpduguard**
- **link-flap**
- **no-internal-vlan**
- **portchannelguard**
- **portsec**
- **tapagg**
- **uplink-failure-detection**
- **xcvr_unsupported**

Related Command

errdisable recovery interval configures the period that an ethernet interface remains disabled before automated recovery begins.

Example

This command enables error-disable recovery for interfaces that are disabled by link-flap and bpduguard conditions and sets the errdisable recovery period at ***10*** minutes.

```
switch(config)# errdisable recovery cause bpduguard
switch(config)# errdisable recovery cause link-flap
switch(config)# errdisable recovery interval 600
switch(config)# show running-config
! Command: show running-config

errdisable recovery cause bpduguard
errdisable recovery cause link-flap
errdisable recovery interval 600
!

switch(config)#
```

11.4.7.9 errdisable recovery interval

The **`errdisable recovery interval`** command specifies the period that an error-disabled Ethernet interface remains disabled before automated errdisable recovery begins. This command affects only interfaces whose automated recovery is enabled for the disabling condition (errdisable recovery cause). When automated recovery is not enabled, interfaces are recovered manually by entering **`shutdown`** and **`no shutdown`** from the interface's configuration mode.

The **`no errdisable recovery interval`** and **`default errdisable recovery interval`** commands restore the default error recovery period of ***300*** seconds by removing the **`errdisable recovery interval`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`errdisable recovery interval` ***period***

`no errdisable recovery interval`

default errdisable recovery interval

Parameters

period Error disable recovery period (seconds). Value ranges from ***30*** to ***86400***. Default value is ***300***.

Related Command

errdisable recovery cause enables the automated recovery of error-disabled Ethernet interfaces.

Example

This command enables error-disable recovery for interfaces that are disabled by link-flap conditions and sets the errdisable recovery period at ***10*** minutes.

```
switch(config)# errdisable recovery cause link-flap
switch(config)# errdisable recovery interval 600
switch(config)# show running-config
! Command: show running-config

!
errdisable recovery cause link-flap
errdisable recovery interval 600
!

!
i
switch(config)#
```

11.4.7.10 interface loopback

The **interface loopback** command places the switch in loopback interface configuration mode for the specified interface and creates a loopback interface if one does not exist. It can also be used to configure multiple loopback interfaces if they have all been previously created.

The command can specify a single interface or multiple interfaces:

- **Single interface:** Command creates an interface if it specifies one that was not previously created.
- **Multiple interfaces:** Command is valid only if all specified interfaces were previously created.

The **no interface loopback** command removes the specified interfaces from ***running-config***, including all interface configuration statements. The **default interface loopback** command removes all configuration statements for the specified loopback interface without deleting the loopback interface from ***running-config***.

The following commands are available in loopback interface configuration mode:

- **description**
- **exit**
- **ip address**
- **ip proxy-arp**
- **ipv6 address**
- **ipv6 enable**
- **load interval**
- **logging event**
- **mtu**
- **shutdown (Interfaces)**
- **snmp trap**

Command Mode

Global Configuration

Command Syntax

interface loopback ***l_range***

no interface loopback ***l_range***

default interface loopback ***l_range***

Parameters

l_range Loopback interfaces (number, range, or comma-delimited list of numbers and ranges). Loopback number ranges from ***0*** to ***1000***.

Examples

- This command enters loopback interface configuration mode for loopback interfaces ***1*** through ***5***.

```
switch(config)# interface loopback 1-5
switch(config-if-Lo1-5)#
```

- This command creates interface ***23*** and enters loopback interface configuration mode.

```
switch(config)# interface loopback 23
switch(config-if-Lo23)#
```

- This command removes loopback interfaces ***5*** through ***7*** from ***running-config***.

```
switch(config)# no interface loopback 5-7
switch(config)#
```

11.4.7.11 ip access-group (Control Plane mode)

The **ip access-group** command applies an IPv4 or standard IPv4 Access Control List (ACL) to the control plane.

The **no ip access-group** and **default ip access-group** commands remove the corresponding **ip access-group** command from ***running-config***.

Command Mode

Control-plane Configuration

Command Syntax

ip access-group ***list_name*** [**VRF_INSTANCE**] **DIRECTION**

no ip access-group [***list_name***][**VRF_INSTANCE**] **DIRECTION**

default ip access-group [***list_name***][**VRF_INSTANCE**] **DIRECTION**

Parameters

- ***list_name*** name of ACL assigned to interface.
- **VRF_INSTANCE** specifies the VRF instance being modified.
  - ***no parameter*** changes are made to the default VRF.
  - **vrf** ***vrf_name*** changes are made to the specified user-defined VRF.
- **DIRECTION** transmission direction of packets, relative to interface. Valid options include:
  - **in** inbound packets.

Example

These commands apply the IPv4 ACL named ***test2*** to the control plane.

```
switch(config)# system control-plane
switch(config-system-cp)# ip access-group test2 in
```

```
switch(config-system-cp)#
```

11.4.7.12 link tracking group (interface)

The **`link tracking group`** command adds the configuration mode interface to a link-state group and specifies whether it is upstream or downstream.

The **`no link tracking group`** and **`default link tracking group`** commands remove the specified link-state group assignment for the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Interface-VXLAN Configuration

Command Syntax

`link tracking group` *group_name* **DIRECTION**

`no link tracking group` [*group_name*]

`default link tracking group` [*group_name*]

Parameters

- ***group_name*** link tracking group name.
- **DIRECTION** position of the interface in the link-state group. Valid options include:
  - **upstream**
  - **downstream**

Example

These commands create link-state group "***xyz***" and add ***VLAN interface 100*** to the group as an upstream interface.

```
switch(config)#link tracking group xyz
switch(config-link-state-xyz)#show active
link tracking group xyz
switch(config-link-state-xyz)#exit
switch(config)#interface vlan 100
switch(config-if-Vl100)#link tracking group xyz upstream
switch(config-if-Vl100)#show active
 interface Vlan100
   link state group xyz upstream
switch(config-if-Vl100)#
```

11.4.7.13 link tracking group

The **`link tracking group`** command creates and enables a link-state group and places the switch in link-state-group configuration mode. A link-state group consists of "upstream" interfaces (connections to servers) and "downstream" interfaces (connections to switches and clients). In the event of a failure of all upstream interfaces in the link-state group, the downstream interfaces are shut down.

The **`no link tracking group`** and **`default link tracking group`** commands delete the link tracking group from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`link tracking group` ***group_name***

`no link tracking group` ***group_name***

`default link tracking group` ***group_name***

Parameters

group_name link-state group name.

Commands available in link-state Configuration Mode

links minimum configures the minimum number of links that the link-state group requires.

Example

This command creates and enables ***link-state group 1***.

```
switch(config)# link tracking group 1
switch(config-link-state-1)#
```

11.4.7.14 links minimum

The **`links minimum`** command specifies the minimum number of links the configuration mode link-state group requires.

The **`no links minimum`** and **`default links minimum`** commands restore the default minimum value of 1 by deleting the corresponding **`links minimum`** statement from ***running-config***.

Command Mode

Link-State Configuration

Command Syntax

`links minimum` ***quantity***

`no links minimum`

`default links minimum`

Parameters

quantity Minimum number of links. Value ranges from ***1*** to ***100000***. Default value is ***1***.

Related Commands

- link tracking group creates and enables a link-state group and places the switch in ***link-state*** configuration mode.
- link tracking group (interface) adds the configuration mode interface to the specified link-state group.

Example

These commands configure link-state tracking group ***link-a*** to have at least ***60*** links.

```
switch(config)# link tracking group link-a
switch(config-link-state-1ink-a)# links minimum 60
switch(config-link-state-link-a)#
```

11.4.7.15 load interval

The **load-interval** command changes the load interval for the configuration mode interface. Load interval is the time period over which data is used to compute interface rate counters. Interface rates are exponentially weighted moving averages; recent data samples have greater influence than older samples. Statistics calculated with shorter load intervals are usually more sensitive to short traffic bursts.

The **no load-interval** and **default load-interval** commands restore the default value of ***300*** seconds by removing the corresponding **load-interval** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Interface-VXLAN Configuration

Command Syntax

load-interval ***delay***

no load-interval

default load-interval

Parameters

delay Load interval delay. Values range from ***5*** to ***600*** (seconds). Default value is ***300*** (five minutes).

Example

These commands set the load interval for ***interface ethernet 7*** at ***60*** seconds.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# load-interval 60
switch(config-if-Et7)#
```

11.4.7.16 mac address learning

The **mac address learning** command enables MAC address learning on a VLAN configuration mode. By default, MAC address learning is enabled by on a VLAN.

The **no mac address learning** command disables MAC address learning for the VLAN configuration mode. The **mac address learning** and **default mac address learning** commands enable MAC address learning for the VLAN configuration mode by deleting the corresponding **no mac address learning** command from the ***running-config***.

Command Mode

Interface-VLAN Configuration

Command Syntax

mac address learning local limit

no mac address learning local limit

default mac address learning local limit

Parameter

local limit Maximum number of locally learned dynamic hosts. Range ***0-10000***. To reset the learning limit threshold to have no limit, use the **`mac address learning`** command.

Examples

- These commands enable MAC address learning on ***vlan 10*** configuration.

```
switch(config)# vlan 10
switch(config-vlan-10)# mac address learning
```

- These commands disable MAC address learning on ***vlan 10*** configuration.

```
switch(config)# vlan 10
switch(config-vlan-10)# no mac address learning
```

- An example for ***5,000*** MACs:

```
switch(config-vla-10)# mac address learning local limit 5000 hosts
```

Mac address learning local limit ***5000*** host.

No mac address learning local limit ***5000*** host.

Default mac address learning local limit ***5000*** host.

11.4.7.17 mac address-table aging-time

The **`mac address-table aging-time`** command configures the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

The **`no mac address-table aging-time`** and **`default mac address-table aging-time`** commands reset the aging time to its default by removing the **`mac address-table aging-time`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`mac-address-table aging-time` ***period***

`no mac-address-table aging-time`

`default mac-address-table aging-time`

Parameters

- ***period*** MAC address table aging time. Default is ***300*** seconds. Options include:
  - ***0*** disables deletion of table entries on the basis of aging time.
  - ***10*** through ***1000000*** (one million) aging period (seconds).

Example

This command sets the MAC address table aging time to two minutes (***120*** seconds).

```
switch(config)# mac address-table aging-time 120
switch(config)#
```

11.4.7.18 mac address-table static

The **mac address-table static** command adds a static entry to the MAC address table. Each table entry references a MAC address, a VLAN, and a list of Layer 2 (Ethernet or port channel) ports. The table supports three entry types: unicast drop, unicast, and multicast.

- A drop entry does not include a port.
- A unicast entry includes one port.
- A multicast entry includes at least one port.

Packets with a MAC address (source or destination) and VLAN specified by a drop entry are dropped. Drop entries are valid for only unicast MAC addresses.

The command replaces existing dynamic or static table entries with the same VLAN-MAC address. Static entries are not removed by aging (mac address-table aging-time). Static MAC entries for mirror destinations or LAG members are typically avoided.

The most important byte of a MAC address distinguishes it as a unicast or multicast address:

- **Unicast:** most significant byte is an even number. Examples: ***0200.0000.0000*** ***1400.0000.0000***.
- **Multicast:** most significant byte is an odd number. Examples: ***0300.0000.0000*** ***2500.0000.0000***.

The **no mac address-table static** and **default mac address-table static** commands remove corresponding **mac address-table static** commands from ***running-config*** and MAC address table entries.

Command Mode

Global Configuration

Command Syntax

mac address-table static ***mac_address*** vlan ***v_num*** [**DESTINATION**]

no mac address-table static ***mac_address*** vlan ***v_num*** [**DESTINATION**]

default mac address-table static ***mac_address*** vlan ***v_num*** [**DESTINATION**]

Parameters

- ***mac_address*** Table entry's MAC address (dotted hex notation – H.H.H).
- ***v_num*** Table entry's VLAN.
- **DESTINATION** Table entry's port list.

  For multicast MAC address entries, the command may contain multiple ports, listed in any order. The CLI accepts only one interface for unicast entries.
- **drop** creates drop entry in table. Valid only for unicast addresses.
  - **interface ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
  - **interface port-channel *p_range*** Port channel interfaces specified by ***p_range***.
  - ***no parameter*** Valid for **no** and **default** commands that remove multiple table entries.

    e_range and ***p_range*** formats include number, range, comma-delimited list of numbers and ranges.

Examples

- This command adds a static entry for unicast MAC address ***0012.3694.03ec*** to the MAC address table.

```
switch(config)# mac address-table static 0012.3694.03ec vlan 3 interface
 ethernet 7
switch(config)# show mac address-table static
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports    Moves   Last Move
----    -----------       ----        -----    -----   ---------
```

```
   3    0012.3694.03ec    STATIC      Et7
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- These commands adds a static drop entry for MAC address ***0012.3694.03ec*** to the MAC address table, then displays the entry in the MAC address table.

```
switch(config)# mac address-table static 0012.3694.03ec vlan 3 drop
switch(config)# show mac address-table static
          Mac Address Table
------------------------------------------------------------

Vlan    Mac Address       Type        Ports    Moves   Last Move
----    -----------       ----        -----    -----   ---------
   1    0012.3694.03ec    STATIC
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- This command adds a static entry for the multicast MAC address ***0112.3057.8423*** to the MAC address table.

```
switch(config)# mac address-table static 0112.3057.8423 vlan 4 interface
port-channel 10 port-channel 12
switch(config)# show mac address-table
          Mac Address Table
------------------------------------------------------------

Vlan    Mac Address       Type        Ports    Moves   Last Move
----    -----------       ----        -----    -----   ---------
Total Mac Addresses for this criterion: 0

          Multicast Mac Address Table
------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   4    0112.3057.8423    STATIC      Po10 Po12
Total Mac Addresses for this criterion: 1
switch(config)#
```

11.4.7.19 monitor link-flap policy

The **`monitor link-flap policy`** command places the switch in link-flap configuration mode for configuring link flap profiles and compiling a default-profile set. Link-flap configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **`exit`** command does not affect the configuration.

Link flap profiles are assigned to Ethernet interfaces and specify conditions that define a link-flap error. When link flap monitoring is enabled on an interface, the link-flap conditions determine when the interface is error-disabled. Multiple profiles can be assigned to an interface to monitor a set of error conditions.

Command Mode

Global Configuration

Command Syntax

`monitor link-flap policy`

Commands Available in Link-flap Configuration Mode

- default-profiles configures the set of profiles that define the default-profile set.
- profile max-flaps (Link Flap Configuration) configures a link-flap profile.

Examples

- These commands place the switch in ***link-flap*** configuration mode.

```
switch(config)# monitor link-flap policy
switch(config-link-flap)#
```

- This command returns the switch to ***global*** configuration mode.

```
switch(config-link-flap)# exit
switch(config)#
```

11.4.7.20 monitor link-flap profiles

The **`monitor link-flap profiles`** command enables link-flap monitoring on the configuration mode interface and specifies the error-disable criteria for the interface. Entering a **`monitor link-flap profiles`** command replaces the corresponding statement in ***running-config***.

The command enables the following link flap detection options:

- **`monitor link-flap`** (***no profiles listed***): The interface detects link flaps using the criteria defined by the default-profile set (default-profiles).
- **`monitor link-flap profiles`** (***at least one profile listed***): The interface detects link flaps using the criteria of the listed profiles. Error-disable criteria require conditions that match at least one profile.
- **`default monitor link-flap`**: The interface detects link flaps using the **`errdisable flap-setting cause link-flap`** and **`errdisable recovery cause`** commands.
- **`no monitor link-flap`**: The interface does not detect link flaps.
- **Default monitor link flap** is the default setting.

Command Mode

Interface-Ethernet Configuration

Interface-Management Configuration

Command Syntax

`monitor link-flap` [**LF_PROFILES**]

`no monitor link-flap`

`default monitor link-flap`

Parameters

LF_PROFILES Name of link-flap profiles assigned to interface. Parameter may contain zero, one, or multiple link-flap profile names:

- ***no parameter*** Link flap criteria determined by default-profile set.

- **profiles *profile_name*** Name of single link-flap profile.
- **profiles *profile_name_1 profile_name_2 ... profile_name_N*** List of link-flap profile names.

Examples

- This command applies the ***LF03*** and ***LF04*** link flap profiles to ***interface ethernet 33***.

```
switch(config)# interface ethernet 33
switch(config-if-Et33)# monitor link-flap profiles LF03 LF04
switch(config-if-Et33)# show active
interface Ethernet33
   monitor link-flap profiles LF04 LF03
switch(config-if-Et33)#
```

- This command disables link-flap monitoring on ***interface ethernet 34***.

```
switch(config)# interface ethernet 34
switch(config-if-Et34)# no monitor link-flap
switch(config-if-Et34)# show active
interface Ethernet34
   no monitor link-flap
switch(config-if-Et34)#
```

11.4.7.21 monitor server-failure link

The **`monitor server-failure link`** command enables Rapid Automated Indication of Link-loss (RAIL) on the configuration mode interface. RAIL must be properly configured globally or this command has no effect on switch operation.

When an interface monitored by RAIL goes down, the switch performs these steps for servers that the switch accesses from the interface:

1. IP addresses of the servers are removed from ARP cache.
2. A dynamic MAC entry is added to the MAC address table for each server. The port for each entry is listed as CPU.

The **`no monitor server-failure link`** and **`default monitor server-failure link`** commands disable RAIL on the configuration mode interface by deleting the corresponding **`monitor server-failure link`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`monitor server-failure link`

`no monitor server-failure link`

`default monitor server-failure link`

Related Commands

monitor server-failure places the switch in server-failure configuration mode for configuring RAIL.

Example

These commands enable RAIL on ***interface port-channel 100***.

```
switch(config)# interface port-channel 100
switch(config-if-Po100)# monitor server-failure link
switch(config-if-Po100)# show active
```

```
interface Port-Channel100
   monitor server-failure link
switch(config-if-Po100)#
```

11.4.7.22 monitor server-failure

The **monitor server-failure** command places the switch in server-failure configuration mode. Rapid Automated Indication of Link-loss (RAIL) settings are configured in server-failure configuration mode. RAIL is disabled by default and is enabled by the **no shutdown** command in server-failure configuration mode.

The **no monitor server-failure** and **default monitor server-failure** commands disable RAIL and restore all settings to their default state by removing all server-failure configuration mode statements from ***running-config***.

Server-failure configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting server-failure configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

monitor server-failure

no monitor server-failure

default monitor server-failure

Commands Available in server-failure Configuration Mode

- network (server-failure configuration mode)
- proxy (server-failure configuration mode)
- shutdown (server-failure configuration mode)

Examples

- These commands place the switch in server-failure configuration mode and enables RAIL.

```
switch(config)# monitor server-failure
switch(config-server-failure)# show active
switch(config-server-failure)# no shutdown
switch(config-server-failure)# show active
monitor server-failure
   no shutdown
switch(config-server-failure)#
```

- This command deletes all server-failure configuration mode commands from ***running-config***.

```
switch(config)# no monitor server-failure
switch(config)#
```

11.4.7.23 monitor session destination cpu

The **monitor session destination cpu** command configures the CPU as the destination port of a specified port mirroring session. The **monitor session source** command configures the source port of the mirroring session. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from one direction.

The CPU can only be configured as a destination for a mirroring session, not as a source. However, the CPU can serve as the destination for multiple mirroring sessions. Traffic mirrored to the CPU can be viewed using tcpdump.

The **no monitor session destination cpu** and **default monitor session destination cpu** commands remove the mirror session destination assignment by deleting the corresponding **monitor session destination cpu** command from ***running-config***. The **no monitor session** command removes the entire mirror session.

Command Mode

Global Configuration

Command Syntax

monitor session ***session_name*** **destination cpu**

no monitor session ***session_name*** **destination cpu**

default monitor session ***session_name*** **destination cpu**

Parameters

session_name Label assigned to port mirroring session.

Guidelines

To view the traffic mirrored to the CPU from a source port, use tcpdump from the Bash shell, with the source interface as an argument. This causes tcpdump to capture packets from the kernel interface of the source port.

Examples

- These commands configure ***interface ethernet 35*** as the source and the CPU as the destination port for the ***redirect_1*** mirroring session, then display the mirror interface.

```
switch(config)# monitor session redirect_1 destination cpu
switch(config)# monitor session redirect_1 source ethernet 35
switch(config)# show monitor session

Session redirect_1
------------------------
Source Ports:

  Both:        Et35

Destination Ports:

    Cpu :  active (mirror0)

switch(config)#
```

- This command uses tcpdump to view the traffic mirrored by the ***redirect_1*** mirroring session. The CPU mirror interface specified in the previous output must be used in the tcpdump expression (in this case, ***mirror0***).

```
switch# bash tcpdump -i mirror0
tcpdump: WARNING: mirror0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on mirror0, link-type EN10MB (Ethernet), capture size 65535 bytes
09:51:12.478363 00:1c:73:27:a6:d3 (oui Arista Networks) > 01:80:c2:00:00:00
 (oui
Unknown), 802.3, length 119: LLC, dsap STP (0x42) Individual, ssap STP
 (0x42)
Command, ctrl 0x03: STP 802.1s, Rapid STP, CIST Flags [Proposal, Learn,
 Forward,
Agreement], length 102
09:51:14.478235 00:1c:73:27:a6:d3 (oui Arista Networks) > 01:80:c2:00:00:00
 (oui
```

```
Unknown), 802.3, length 119: LLC, dsap STP (0x42) Individual, ssap STP
 (0x42)
Command, ctrl 0x03: STP 802.1s, Rapid STP, CIST Flags [Proposal, Learn,
 Forward,
Agreement], length 102 switch#
```

11.4.7.24 monitor session destination

The **monitor session destination** command configures an interface as the destination port of a specified port mirroring session. The destination is usually an Ethernet interface, but other options are available on certain platforms (see **Guidelines**). The monitor session source command configures the source port of the mirroring session.

An interface cannot be used in more than one mirror session and cannot be simultaneously used as both source and destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic only from one direction.

Note: On platforms which support the use of port channels as mirror destinations, a port channel *must not* be used as a mirror destination if it is a member of an MLAG.

The **no monitor session destination** and **default monitor session destination** commands remove the mirroring session destination assignment by deleting the corresponding **monitor session destination** command from ***running-config***. The **no monitor session** removes the entire mirroring session.

Command Mode

Global Configuration

Command Syntax

monitor session ***session_name*** **destination**{cpu | ethernet ***e_range*** | port-channel ***p_range*** | tunnel mode}

no monitor session ***session_name*** **destination**

default monitor session ***session_name*** **destination**

Parameters

- ***session_name*** label assigned to the port mirroring session.
- **cpu** configures a CPU as the destination interface.
- **ethernet *e_range*** configures Ethernet interfaces specified by ***e_range*** as the destination interface. The ethernet interface value ranges from ***1*** to ***50***.
- **port-channel *p_range*** configures port channel interfaces specified by ***p_range*** as the destination interface. The port-channel value ranges from ***1*** to ***2000***.
- **tunnel mode** configures a tunnel as the destination interface. Option includes:
  - **gre** configures GRE-tunnel as the destination interface.

Guidelines

Tunnel mode is supported on select platforms only.

Port mirroring capacity varies by platforms. The session destination capacity of switches on each platform is listed below:

- **Arad Platform:** Ethernet interfaces (one).
- **FM6000 Platform:** Ethernet interfaces (any count), Port channel interfaces (any count), CPU.
- **Petra Platform:** Ethernet interfaces (eight for Rx or Tx sessions; four for both ways).
- **Trident Platform:** Ethernet interfaces (one).
- **Trident II Platform:** Ethernet interfaces (one).

When there are multiple transmit (Tx) sources in a monitor session, mirrored frames use Tx properties of the lowest numbered Tx mirror source configured. Packets are modified based on properties.

Allowed VLANs on the ***ethernet8*** source interface are ***10***, ***20*** and ***30***. Allowed VLANs on ***ethernet9*** source interface are ***30***, ***40***, and ***50***. The frames going out of ***ethernet9*** tagged with ***10***, ***20***, and ***30*** appears at the mirrored destination as tagged frames. The tagged frames with ***40*** or ***50*** on ***ethernet9*** appears at the mirrored destination as untagged frames. Since ***ethernet8*** is the lowest numbered source interface, all Tx frames on ethernet8 are tagged in the mirrored destination.

Examples

- This command configures ***interface ethernet 8*** as the destination port for the ***redirect_1*** mirroring session.

```
switch(config)# monitor session redirect_1 destination ethernet 2
switch(config)# show monitor session

Session redirect_1
------------------------
Source Ports:

Destination Ports:

    Et2 :  active

switch(config)#
```

- This command configures a GRE tunnel with source and destination addresses as ***1.1.1.1*** and ***2.2.2.2*** respectively as the destination interface for the ***redirect_2*** mirroring.

```
switch(config)# monitor session redirect_2 destination tunnel mode gre source
1.1.1.1 destination 2.2.2.2
switch(config)# show monitor session

Session redirect_2
------------------------
Source Ports:

Destination Ports:

         status  source    dest    TTL  DSCP  proto    VRF     fwd-drop
 Gre1 :  active  1.1.1.1   2.2.2.2 128  0     0x88be   default no

switch(config)#
```

11.4.7.25 monitor session forwarding-drop

The **`monitor session forwarding-drop`** command configures a forwarding-drop session for mirroring ingress packets that are dropped during ASIC forwarding.

The **`no monitor session forwarding-drop`** and **`default monitor session forwarding-drop`** commands delete the current forwarding-drop configuration.

Command Mode

Global Configuration

Command Syntax

`monitor session` ***`session_name`*** **`forwarding-drop destination tunnel mode`**

`no monitor session` ***`session_name`*** **`forwarding-drop destination tunnel mode`**

`default monitor session` ***`session_name`*** **`forwarding-drop destination tunnel mode`**

Parameters

- **destination** specifies to mirror packets at destination.
- **tunnel mode** specifies to mirror packets that pass through a tunnel. Options include:
  - **gre** configures GRE-tunnel as the destination interface.

Related Commands

- monitor session destination
- monitor session destination cpu
- show monitor session

Guidelines

The forwarding-drop configuration is supported on select platforms only.

Example

This command configures a forwarding-drop session to ***1.1.1.1*** as the destination.

```
switch(config)# monitor session 1 forwarding-drop destination tunnel mode gre
 source 1.1.1.1 destination
2.2.2.2
switch(config)# show monitor session

Session 1
-------------------------
Programmed in HW: No
Source Ports:
Destination Ports:
            status   source    dest      TTL   DSCP  proto    VRF       fwd-
drop
    Gre1 :  active   1.1.1.1   2.2.2.2   128    0    0x88be   default   yes

switch(config)#
```

11.4.7.26 monitor session ip access-group

The **monitor session ip access-group** command configures an ACL to filter the traffic being mirrored to the destination port.ACLs applied to a source port affect the RX side of the interface, and do not impact the TX side of the interface. TX mirrored packets cannot be filtered, and will continue to be sent to the mirror destination.

The **no monitor session ip access-group** and **default monitor session ip access-group** commands remove the filter from the specified mirror session by deleting the corresponding **monitor session ip access-group** command from ***running-config***. The **no monitor session** command removes the entire mirror session.

Command Mode

Global Configuration

Command Syntax

monitor session ***session_name*** **ip access-group** ***acl_name***

no monitor session ***session_name*** **ip access-group**

default monitor session ***session_name*** **ip access-group**

Parameters

- ***session_name*** Label assigned to port mirroring session.
- ***acl_name*** The ACL to be applied to filter traffic for the specified session.

Examples

- These commands create an ACL and apply it to filter the traffic mirrored to the destination port by session ***redirect_1***.

```
switch(config)# ip access-list allow-host
switch(config-acl-allow-host)# 10 permit ip host 192.168.11.24 host
 10.0.215.23
switch(config-acl-allow-host)# 20 deny ip any any
switch(config-acl-allow-host)# exit
switch(config)#
switch(config)# monitor session redirect_1 ip access-group allow-host
switch(config)#
```

- Use the **`show monitor session`** command to verify the configuration.

```
switch# show monitor session
Session redirect_1
------------------------
Source Ports:
Both:        Et35(Acl:allow-host)
Destination Ports:
Cpu :  active (mirror0)
ip access-group: allow-host switch#
```

11.4.7.27 monitor session source

The **`monitor session source`** command configures the source port of a specified port mirroring session. The **`monitor session destination`** or **`monitor session destination cpu`** command configures the destination port of the mirroring session.

An interface cannot be used in more than one mirror session and cannot be simultaneously a source and a destination. An interface which is part of a port channel cannot be used as a source, but a port channel which is a member of an MLAG can be used. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from only one direction.

The **`no monitor session source`** and **`default monitor session source`** commands remove the mirroring session source assignment by deleting the corresponding **`monitor session source`** command from ***running-config***. The **`no monitor session`** removes entire the mirroring session.

Command Mode

Global Configuration

Command Syntax

`monitor session` ***session_name*** **`source`** **INT_NAME DIRECTION**

`no monitor session` ***session_name*** **`source`** **INT_NAME DIRECTION**

`default monitor session` ***session_name*** **`source`** **INT_NAME DIRECTION**

Parameters

- ***session_name*** Label assigned to port mirroring session.
- **INT_NAME** Source interface for the mirroring session.
  - **ethernet** ***e_range*** Ethernet interfaces specified by ***e_range***.
  - **port-channel** ***p_range*** Port channel interfaces specified by ***p_range***.
- **DIRECTION** transmission direction of traffic to be mirrored.
  - ***no parameter*** mirrors transmitted and received traffic.
  - **both** mirrors transmitted and received traffic.
  - **rx** mirrors received traffic only.
  - **tx** mirrors transmitted traffic only.

Guidelines

On DCS-7050, DCS-7050X, DCS-7250X, and DCS-7300X series, due to limitations of the switch ASIC, all frames mirrored on egress are prefixed with an 802.1Q VLAN tag, even when the egress port is configured as an access port. If the capture device is unable to process VLAN tags in a desirable manner mirroring should be configured exclusively for ingress traffic by specifying **rx**.

Restrictions

Port mirroring capacity varies by platform. Session source capacity for each platform is listed below:

- **FM6000 Platform:** Ethernet interfaces (any number), port channel interfaces (any number).
- **Arad Platform:** Ethernet interfaces (any number), port channel interfaces (any number).
- **Petra Platform:** Ethernet interfaces (eight for Rx or Tx sessions; four for both ways).
- **Trident Platform:** Ethernet interfaces (any number), port channel interfaces (any number).
- **Trident II Platform:** Ethernet interfaces (any number), port channel interfaces (any number).

  The number of interfaces that can be effectively mirrored is restricted by the destination port speed.

Example

This command configures ***interface ethernet 7*** as the source port for ***redirect_1*** mirroring session.

```
switch(config)# monitor session redirect_1 source ethernet 7
switch(config)#
```

11.4.7.28 monitor session source ip access-group

The **monitor session source ip access-group** command configures an ACL to filter the traffic being mirrored from a specific source port. This enables the ability to filter traffic using a different ACL on each source port and have the combined matched traffic sent to the destination port.

The **no monitor session source ip access-group** and **default monitor session source ip access-group** commands remove the filter from the specified mirror session by deleting the corresponding **monitor session source ip access-group** command from ***running-config***. The **no monitor session** command removes the entire mirror session.

Command Mode

Global Configuration

Command Syntax

monitor session ***s_name*** **source** **INT_NAME** [**DIRECT**] **ip access-group** ***acl_name***

no monitor session ***s_name*** **source** **INT_NAME** [**DIRECT**] **ip access-group** ***acl_name***

default monitor session ***s_name*** **source** **INT_NAME** [**DIRECT**] **ip access-group** ***acl_name***

Parameters

- ***s_name*** Label assigned to port mirroring session.
- **INT_NAME** Source interface for the mirroring session.
  - **ethernet** ***e_range*** Ethernet interfaces specified by ***e_range***.
  - **port-channel** ***p_range*** Port channel interfaces specified by ***p_range***.
- **DIRECT** transmission direction of traffic to be mirrored. Options include:
  - ***no parameter*** mirrors received traffic only.
  - **rx** mirrors received traffic only.
- ***acl_name*** The ACL to be applied to filter traffic for the specified session.

Example

These commands create ACLs and apply them to filter the traffic mirrored from two source ports by session ***redir_1***.

```
switch(config)# ip access-list allow-host-x
switch(config-acl-allow-host-x)# 10 permit ip host 192.168.11.24 host
 10.0.215.23
switch(config-acl-allow-host-x)# 20 deny ip any any
switch(config-acl-allow-host-x)# exit
switch(config)# ip access-list allow-host-y
switch(config-acl-allow-host-y)# 10 permit ip host 172.16.233.80 host
 10.0.215.23
switch(config-acl-allow-host-y)# 20 deny ip any any
switch(config-acl-allow-host-y)# exit
switch(config)# monitor session redir_1 source ethernet 5,9 rx
switch(config)# monitor session redir_1 source ethernet 5 ip access-group
 allow-host-x
switch(config)# monitor session redir_1 source ethernet 9 ip access-group
 allow-host-y
switch(config)#
```

11.4.7.29 monitor session truncate

The **`monitor session truncate`** command configures a port mirroring session to truncate mirrored packets, retaining only the first ***160*** bytes. Packet truncation can be used to prevent oversubscription of the session's destination port.

Packet truncation applies to the mirroring session as a whole, and cannot be applied to individual source ports.

The **`no monitor session truncate`** and **`default monitor session truncate`** commands restores mirroring of full packets by deleting the corresponding **`monitor session truncate`** command from ***running-config***. The **`no monitor session`** removes the entire mirroring session.

Command Mode

Global Configuration

Command Syntax

`monitor session` ***`session_name`*** **`truncate`**

`no monitor session` ***`session_name`*** **`truncate`**

`default monitor session` ***`session_name`*** **`truncate`**

Parameters

session_name Label assigned to port mirroring session.

Example

This command configures mirroring session ***redirect_1*** to truncate mirrored packets.

```
switch(config)# monitor session redirect_1 truncate
switch(config)#
```

11.4.7.30 mtu

The **`mtu`** command configures the IPv4 and IPv6 Maximum Transmission Unit (MTU) size for the configuration mode interface. The switch fragments IP packets that are larger than the MTU value for the outbound interface. An interface's MTU value is displayed with the show interfaces command.

MTU is independently configurable on all routable interfaces.

The **no mtu** and **default mtu** commands restore the interface's MTU to the default value by removing the corresponding mtu command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

mtu ***bytes***

no mtu

default mtu

Parameters

bytes MTU size (bytes). Values range from ***68*** to ***9214***. The default MTU size is ***1500*** bytes.

Example

This command sets the MTU size of ***1492*** bytes on ***interface vlan 20***.

```
switch(config)#interface vlan 20
switch(config-if-Vl20)#mtu 1492
switch(config-if-Vl20)#
```

11.4.7.31 network (server-failure configuration mode)

The **network** command specifies the IPv4 network space that Rapid Automated Indication of Link-loss (RAIL) monitors for failed links to connected servers. RAIL reduces the wait time for applications on directly connected servers that are blocked due to a failed link. ***Running-config*** supports simultaneous network command, allowing RAIL to monitor multiple disjoint network spaces.

When a server on the specified network is blocked because of a failed Ethernet or port channel link, the switch becomes a proxy for the unavailable server and responds with **TCP RST** or **ICMP Unreachable** segments to devices sending packets to the unavailable server.

The **no network** and **default network** commands terminate the RAIL monitoring of the specified IPv4 address space by deleting the corresponding **network** command from ***running-config***.

Command Mode

Server-failure Configuration

Command Syntax

network ***netv4_address***

no network ***netv4_address***

default network ***netv4_address***

Parameters

netv4_addr IPv4 subnet address to be monitored (CIDR or address-mask notation).

Related Command

monitor server-failure places the switch in server-failure configuration mode.

Example

This command specifies two IPv4 network spaces that RAIL monitors for server failures.

```
switch(config)# monitor server
switch(config-server-failure)# network 10.1.1.0/24
switch(config-server-failure)# network 10.2.1.96/28
switch(config-server-failure)# show active
monitor server-failure
   network 10.2.1.96/28
   network 10.1.1.0/24
switch(config-server-failure)#
```

11.4.7.32 no monitor session

The **no monitor session** and default monitor session commands remove the specified monitor session from the switch by deleting all corresponding monitor commands from ***running-config***. Commands that remove or alter individual commands within a session configuration are described in the monitor session destination and **monitor session source** commands.

Command Mode

Global Configuration

Command Syntax

no monitor session ***session_name***

default monitor session ***session_name***

Parameters

session_name Label assigned to port mirroring session.

Example

This command displays the configuration of the ***redirect_1*** mirroring session, deletes the session, then confirms that the session was removed.

```
switch(config)# show monitor session redirect_1
Session redirect_1
-----------------------
Source Ports
  Both:        Et7
Destination Port: Et8
switch(config)# no monitor session redirect_1
switch(config)# show monitor session redirect_1
Session not created

switch(config)#
```

11.4.7.33 phy diag

Use the **phy diag** command to configure a test pattern in the interface configuration mode. The **no** and **default** forms of the command disables the test pattern.

Command Mode

Interface configuration mode

Command Syntax

phy diag [transmitter | receiver] test pattern ***TestPattern***

no phy diag [transmitter | receiver] test pattern ***TestPattern***

default phy diag [transmitter | receiver] test pattern ***TestPattern***

Parameters

- **transmitter** Configures the physical transmitter.
- **receiver** Configures the physical receiver.
- **test pattern *TestPattern*** Configures the named test pattern.

Examples

- Enable a test pattern on an interface using the **phy diag** command. You can select the transmitter or the receiver. To display the available interfaces, select **test pattern ?**.

```
switch(config-if)# phy diag [ transmitter | receiver ] test pattern ?
  PRBS11  Configure the PRBS11 test pattern
  PRBS15  Configure the PRBS15 test pattern
  PRBS23  Configure the PRBS23 test pattern
  PRBS31  Configure the PRBS31 test pattern
  PRBS49  Configure the PRBS49 test pattern
  PRBS58  Configure the PRBS58 test pattern
  PRBS7   Configure the PRBS7 test pattern
  PRBS9   Configure the PRBS9 test pattern
```

- To disable a test pattern on an interface, enter the following command. You can select the tranmitter or the receiver, as well as the selected named test pattern.

```
switch(config-if)# no phy diag [ transmitter | receiver ] test
 pattern TestPattern
```

- By default, a test pattern is disabled.

```
switch(config-if)# default phy diag [ transmitter | receiver ] test pattern
```

- The following command clears the recorded test pattern status data for all the interfaces. Upon running the command, all the counter values are set to ***0*** and link states are marked as **not locked**.

```
switch# clear phy diag test pattern
```

11.4.7.34 platform sand monitor serdes error log

The **platform sand monitor serdes error log** command is used for enabling the serdes error log for fabric link monitoring.

Command Mode

Global Configuration

Command Syntax

platform sand monitor serdes error log

Example

This command enables the serdes error log for fabric link monitoring.

```
switch(config)# platform sand monitor serdes error log
switch(config)#
```

11.4.7.35 platform sand monitor serdes error threshold

The **platform sand monitor serdes error threshold** command is used for generating a fabric link monitoring serdes error threshold.

Command Mode

Global Configuration

Command Syntax

platform sand monitor serdes error threshold

Example

This command monitors serdes error thresholds over the specified number of received cells, resulting in the isolation of a fabric link between ***200*** and ***30000*** received cells.

```
switch(config)# platform sand monitor serdes error threshold 200 30000
switch(config)#
```

11.4.7.36 platform sand monitor serdes poll period

The **platform sand monitor serdes poll period** command is used to enable the serdes poll period.

Command Mode

Global Configuration

Command Syntax

platform sand monitor serdes poll period

Example

This command changes the serdes polling period for fabric link monitoring to ***6*** seconds.

```
switch(config)# platform sand monitor serdes poll period 6
switch(config)#
```

11.4.7.37 platform sand monitor serdes poll threshold isolation

The **platform sand monitor serdes poll threshold isolation** command is used to set and enables fabric link monitoring for serdes poll threshold isolation.

Command Mode

Global Configuration

Command Syntax

platform sand monitor serdes poll threshold isolation

Example

This command changes the number of consecutive polls in which the threshold needs to be detected to isolate a link. In this case the number is ***5*** consecutive polls.

```
switch(config)# platform sand monitor serdes poll threshold isolation 5
switch(config)#
```

11.4.7.38 platform sand monitor serdes poll threshold recovery

The **platform sand monitor serdes poll threshold recovery** command is used to set and enable fabric link monitoring for serdes poll threshold recovery.

Command Mode

Global Configuration

Command Syntax

`platform sand monitor serdes poll threshold recovery`

Example

This command changes the number of consecutive serdes polls used for threshold recovery to **`6`** seconds.

```
switch(config)# platform sand monitor serdes poll threshold recovery 6
switch(config)#
```

11.4.7.39 profile max-flaps (Link Flap Configuration)

The **`profile max-flaps`** command creates a link flap profile that, when assigned to an Ethernet interface, specifies the conditions that result in an error-disable action. Link flap profile parameters include:

- **flaps** Threshold number of interface state changes.
- **period** Interval when link flaps accumulate to trigger an error condition.
- **violations** Number of link flap errors (threshold exceeded over specified period).
- **intervals** Quantity of periods.

By default, **violations** and **intervals** are each set to one, resulting in a profile that triggers a link-flap error when the specified frequency is exceeded once. By configuring violations and intervals, link-flap errors are defined when the frequency is exceeded multiple times over a specified set of intervals.

Default is a reserved profile name that modifies the errdisable flap-setting cause link-flap statement in ***running-config***. When configuring the **default** profile, **violations** and **intervals** are disregarded.

The **`no profile max-flaps`** command removes the specified profile by deleting the corresponding **`profile max-flaps`** command from ***running-config***. The **`no profile max-flaps default`** command restores default errdisable flap-setting cause link-flap values by removing that command from ***running-config***.

Command Mode

Link-flap Configuration

Command Syntax

`profile` **PROFILE_NAME** **`max-flaps`** ***flap_max*** time ***period*** [**EXTENSIONS**]

`no profile` **LF_PROFILE**

Parameters

- **PROFILE_NAME** Name of link flap profile. Options include:
  - **default** command modifies default values (errdisable flap-setting cause link-flap).
  - ***profile_name*** command modifies specified link-flap profile.
- ***flap_max*** Threshold number of interface state changes. Value ranges from ***1*** to ***100***.
- ***period*** Interval when flaps accumulate toward threshold (seconds). Value ranges from ***1*** to ***1800***.
- **EXTENSIONS** Configures multi-flap triggers. Options include:
  - ***no parameter*** Sets errors and episodes to default values (one).
  - **violations *errors* intervals *episodes*** Link flap errors (***errors***) and number of periods (***episodes***).
    - ***Errors*** range is ***1*** to ***1000***. Default value is ***1***.
    - ***Episodes*** range is ***1*** to ***1000***. Default value is ***1***.

Related Command

monitor link-flap policy places the switch in ***link-flap*** configuration mode.

Example

These commands create two link flap profiles with various trigger settings.

```
switch(config)# monitor link-flap policy
switch(config-link-flap)# profile LF01 max-flaps 15 time 60
switch(config-link-flap)# profile LF02 max-flaps 10 time 30 violations 5
 intervals 10
switch(config-link-flap)# show active
monitor link-flap policy
   profile LF01 max-flaps 15 time 60 violations 1 intervals 1
   profile LF02 max-flaps 10 time 30 violations 5 intervals 10
switch(config-link-flap)#
```

11.4.7.40 proxy (server-failure configuration mode)

The **proxy** command enables the Rapid Automated Indication of Link-loss (RAIL) proxy setting and specifies the interval that RAIL responds to messages sent to servers on failed links, starting from when the switch detects the failed link. The RAIL state machine is in the proxying state during the timeout interval this command specifies. When RAIL proxy is not enabled, the switch maintains a list of unavailable servers without responding to messages sent the servers. The switch can enter RAIL proxy state only when this command is enabled.

The RAIL proxy setting is **disabled** by default. When RAIL proxy is enabled, the default period is three minutes.

The **no proxy** and **default proxy** commands return the RAIL proxy setting to disabled by removing the proxy statement from ***running-config***.

The **no proxy lifetime** and **default proxy lifetime** command sets the proxy time setting to its default value of three minutes if the RAIL proxy setting is **enabled**. These commands have no effect if the RAIL proxy setting is **disabled**.

Command Mode

Server-failure Configuration

Command Syntax

proxy [lifetime ***time_span***]

no proxy [lifetime]

default proxy [lifetime]

Parameters

timespan proxy timeout period (minutes). Value ranges from *1* to ***10080***. Default value is ***3***.

Related Command

monitor server-failure places the switch in server-failure configuration mode.

Examples

- These commands enable the RAIL proxy and sets the proxy timeout period of **10** minutes.

```
switch(config)# monitor server
switch(config-server-failure)# proxy lifetime 10
switch(config-server-failure)# show active
monitor server-failure
   proxy lifetime 10
switch(config-server-failure)#
```

- This command sets the proxy timeout period to its default value of **3** minutes.

```
switch(config-server-failure)# no proxy lifetime
```

```
switch(config-server-failure)# show active
monitor server-failure
   proxy
switch(config-server-failure)#
```

- This command disables the RAIL proxy.

```
switch(config-server-failure)# no proxy
switch(config-server-failure)# show active
monitor server-failure
switch(config-server-failure)#
```

11.4.7.41 show bridge mac-address-table aging timeout

The **show bridge mac-address-table aging timeout** command displays the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries that exceed the aging time.

Aging time ranges from ***10*** seconds to ***1000000*** seconds with a default of ***300*** seconds (five minutes).

Command Mode

EXEC

Command Syntax

show bridge mac-address-table aging timeout

Example

This command shows the MAC address table aging time.

```
switch> show bridge mac-address-table aging timeout
Global Aging Time:  120 switch>
```

11.4.7.42 show errdisable recovery

The **show errdisable recovery** command displays information about the recovery intervals and error disable causes.

Command Mode

EXEC

Command Syntax

show errdisable recovery

Parameters

- ***no parameter*** state of the system.

The following output is for a system where the causes are listed and interval timer for each cause is identified along with the timer status.

```
switch# show errdisable recovery
Errdisable Reason              Timer Status   Timer Interval
------------------------------ ---------------- --------------
   bpduguard                      Disabled                30
   hitless-reload-down            Disabled               300
   lacp-no-portid                 Disabled               N/A
   lacp-rate-limit                Disabled               300
```

```
    license-enforce                Disabled                  N/A
    link-flap                      Disabled                  300
    no-internal-vlan               Disabled                  300
    uplink-failure-detection       Disabled                  300
```

11.4.7.43 show fabric monitoring health

The **`platform sand monitor health`** command is used to display the fabric monitoring connected state status with isolated links.

Command Mode

Global Configuration

Command Syntax

`platform sand monitor health`

Example

This command displays the connected state status with isolated links.

```
switch(config)# show platform sand health
Fabric serdes isolated by fabric monitoring: (36 total)

Arad5/0 serdes [0-1, 10-19, 2, 20-29, 3, 30-35, 4-9]

Top fabric serdes list by number of times isolated by monitoring:
Arad5/0 serdes 0: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 1: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 10: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 11: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 12: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 13: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 14: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 15: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 16: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 17: 1 (last occurred: 0:01:04 ago)

switch(config)#
```

11.4.7.44 show interfaces

The **`show interfaces`** command displays operational status and configuration information of specified interfaces. The output includes speed, duplex, flow control information and basic interface statistics.

The input and output bit rates, as displayed, do not include framing bits that are part of the Ethernet standard, the inter-frame gap and preamble that total 20 bytes per packet. The percentage number includes those framing bits to provide a better link utilization estimate.

Command Mode

EXEC

Command Syntax

`show interfaces` [**INT_NAME**]

Parameters

INT_NAME Interface type and numbers. Options include:

- ***no parameter*** all interfaces.

- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **loopback *l_range*** Loopback interface specified by ***l_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
- **vlan *v_range*** VLAN interface range specified by ***v_range***.
- **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.

Valid range formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command display configuration and status information for Ethernet interface ***1*** and ***2***.

```
switch> show interfaces ethernet 1-2
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2481.7647 (bia 001c.2481.7647)
  Description: mkt.1
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 seconds input rate 33.5 Mbps (0.3% with framing), 846 packets/sec
  5 seconds output rate 180 kbps (0.0% with framing), 55 packets/sec
     76437268 packets input, 94280286608 bytes
     Received 2208 broadcasts, 73358 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     6184281 packets output, 4071319140 bytes
     Sent 2209 broadcasts, 345754 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output
Ethernet2 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2481.7648 (bia 001c.2481.7648)
  Description: mkt.2
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 seconds input rate 711 kbps (0.0% with framing), 271 packets/sec
  5 seconds output rate 239 kbps (0.0% with framing), 65 packets/sec
     73746370 packets input, 78455101010 bytes
     Received 11 broadcasts, 83914 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     5687714 packets output, 4325064454 bytes
     Sent 15 broadcasts, 107279 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output switch>
```

11.4.7.45 show interfaces description

The **`show interfaces description`** command displays the status and description text of the specified interfaces. The description command configures an interface's description parameter.

Command Mode

EXEC

Command Syntax

`show interfaces` [**INT_NAME**] **`description`**

Parameters

INT_NAME Interface type and labels. Options include:

- ***no parameter*** all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **loopback *l_range*** Loopback interface specified by ***l_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
- **vlan *v_range*** VLAN interface range specified by ***vx_range***.
- **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.

  Range formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays description text and status of ***interfaces ethernet 1-10***.

```
switch> show interfaces ethernet 1-10 description
Interface                      Status         Protocol Description
Et1                            up             up       ctar_01
Et2                            up             up       ctar_02
Et3                            up             up       ctar_03
Et4                            up             up       fobd_01
Et5                            up             up       fobd_02
Et6                            up             up       yzrq_01
Et7                            up             up       yzrq_02
Et8                            down           down     yzrq_03
Et9                            up             up       yzrq_04
Et10                           up             up       yzrq_05 switch>
```

11.4.7.46 show interfaces phy diag

Command Mode

EXEC

Command Syntax

`show interfaces` [interface type ***interface range***] **`phy diag`** [error-correction | test pattern]

Parameters

- **interface type *interface range*** Type of interface and range.
- **error-correction** Forwards error correction.
- **test pattern** Displays test patterns.

Guidelines

The user-configured test pattern is displayed under the **`Configured`** column, which is divided based on transmitter and receiver configuration. The currently operational test pattern is displayed under the **`Operational`** column. The **`Available`** column lists the test patterns available for the interface.

Example

- In this example, interfaces ethernet ***26/1*** and ***31/1*** in the **`show interfaces ethernet 26/1,31/1 phy diag test pattern`** command are selected to display the the configured and operational test pattern, and the available test patterns.

```
switch# show interfaces ethernet 26/1,31/1 phy diag test pattern
                 Configured       Operational
Interface     Transmit Receive Transmit Receive Available
------------- -------- ------- -------- ------- -----------------------
Ethernet26/1  PRBS15   PRBS15  PRBS15   PRBS15  PRBS 7,9,11,15,23,31,58
```

```
Ethernet31/1  PRBS7    PRBS31  PRBS7    PRBS31  PRBS 7,9,11,15,23,31,58
```

The user-configured test pattern is displayed under the **Configured** column, which is divided based on transmitter and receiver configuration. The currently operational test pattern is displayed under the **Operational** column. The **Available** column lists the test patterns available for the interface.

- In this example, the **show interfaces ethernet 26/1 phys detail | i Test pattern** command displays the operational test pattern for an interface. Here the **Test pattern** field will not be available, on disabling the test pattern.

```
switch# show interfaces ethernet 26/1 phy detail | i Test pattern
  Test pattern                 enabled
switch# show interfaces ethernet 31/1 phy detail | i Test pattern
  Test pattern                 enabled
```

- In this example, the **show interfaces ethernet 26/1,31/1 phy diag test pattern counters** command displays test pattern link state and error information.

  The following information is listed in the display output:

  - **Link state:** whether or not the checker locked on to the configured test pattern.
  - **Bit Errors:** the accumulated number of bit errors.
  - **Largest Burst:** the largest burst of errors that occurred.
  - **Burst Count:** the number of occurrences of errors.
  - **Last Error Time:** the last time an error has occurred, ‘never’ if no errors have occurred.

```
switch# show interfaces ethernet 26/1,31/1 phy diag test pattern counters
Current System Time: Wed May 30 22:24:32 2018
                                          Largest    Burst
Interface        Lane  Link State  Bit Errors   Burst      Count  Last Error Time
---------------- ----- ----------- ------------ ---------- ------ ----------------
Ethernet26/1     0     locked      409266       409266     1       0:21:27 ago
Ethernet26/1     1     locked      347084       347084     1       0:21:27 ago
Ethernet26/1     2     locked      420681       420681     1       0:21:27 ago
Ethernet26/1     3     locked      392969       392969     1       0:21:27 ago
Ethernet31/1     0     not locked  1417655      651822     3       0:03:20 ago
Ethernet31/1     1     not locked  1782238      736819     3       0:03:20 ago
Ethernet31/1     2     not locked  1760538      866185     3       0:03:20 ago
Ethernet31/1     3     not locked  1817413      923941     3       0:03:20 ago
```

- In this example, the **show interfaces ethernet 26/1,31/1 phy diag test pattern counters** command displays the lock state of an interface along with a detailed information on the recorded bit errors.

  The following information is listed in the display output:

  - **Last clear:** the time when the test pattern results were last cleared.
  - **Operational test pattern:** the test pattern operational at the receiver side.
  - **Bit rate:** the transmission bit rate.
  - **Lock state:** the current lock status, number of times it changed and the last time the lock status got changed.
    - **locked:** receiver is able to lock on to the incoming test pattern.
    - **not locked:** receiver is not able to lock on to the incoming test pattern.
  - **Largest burst:** the largest burst of errors that occurred.
  - **Bit errors*:** the accumulated number of errors, number of occurrences of errors, and last time errors were captured. The ***** suffix, indicating that data may not be accurate due to loss of lock, is applied if the current lock status is not locked or if the lock status has changed more than once. This suffix is cleared when the test pattern status data is cleared via the CLI listed above.
  - **Total Bits:** the total bits received.
  - **Bit error rate (BER)*:** the ratio of captured bit errors to the total bit received. The ***** suffix, indicating that data may not be accurate due to loss of lock, is applied if the current lock status is not locked or if

the lock status has changed more than once. This suffix is cleared when the test pattern status data is cleared via the CLI listed above.

- **Bit errors since last lock:** the accumulated number of errors since last time lock was gained.
- **Total bits since last lock:** the total bits received since last lock.
- **BER since last lock:** the ratio of captured bit errors to the total bit received since last lock.

```
switch# show interfaces ethernet 26/1,31/1 phy diag test pattern counters detail
*: Data may not be accurate due to loss of lock.

Current System Time:  Wed May 30 23:36:34 2018
Ethernet26/1
  Last clear                     1:33:29 ago
  Operational test pattern       PRBS15
                                 Current State     Changes      Last Change
                                 -------------     -------      -----------
  Lane 0
     Bit rate                    25.781 Gbps
     Lock state                  locked                 1       1:33:28 ago
     Largest burst               409266
     Bit errors                  409266                 1       1:33:28 ago
     Total bits                  144,607.648 Gb
     Bit error rate              2.83E-09
     Bit errors since last lock  409266
     Total bits since last lock  161,542.986 Gb
     BER since last lock         2.53E-09
  Lane 1
     Bit rate                    25.781 Gbps
     Lock state                  locked                 1       1:33:28 ago
     Largest burst               347084
     Bit errors                  347084                 1       1:33:28 ago
     Total bits                  144,607.668 Gb
     Bit error rate              2.40E-09
     Bit errors since last lock  347084
     Total bits since last lock  161,543.006 Gb
     BER since last lock         2.15E-09
  Lane 2
     Bit rate                    25.781 Gbps
     Lock state                  locked                 1       1:33:28 ago
     Largest burst               420681
     Bit errors                  420681                 1       1:33:28 ago
     Total bits                  144,607.658 Gb
     Bit error rate              2.91E-09
     Bit errors since last lock  420681
     Total bits since last lock  161,542.996 Gb
     BER since last lock         2.60E-09
  Lane 3
     Bit rate                    25.781 Gbps
     Lock state                  locked                 1       1:33:28 ago
     Largest burst               392969
     Bit errors                  392969                 1       1:33:28 ago
     Total bits                  144,607.678 Gb
     Bit error rate              2.72E-09
     Bit errors since last lock  392969
     Total bits since last lock  161,543.016 Gb
     BER since last lock         2.43E-09

Ethernet31/1
  Last clear                     1:33:29 ago
  Operational test pattern       PRBS31
                                 Current State     Changes      Last Change
                                 -------------     -------      -----------
  Lane 0
     Bit rate                    25.781 Gbps
     Lock state                  not locked             3       1:15:22 ago
     Largest burst               651822
     Bit errors                  1417655*               3       1:15:22 ago
     Total bits                  144,626.220 Gb
 Bit error rate              > 9.80E-09*
     Bit errors since last lock  765833*
     Total bits since last lock  144,471.763 Gb
     BER since last lock         > 5.30E-09*
  Lane 1
     Bit rate                    25.781 Gbps
     Lock state                  not locked             3       1:15:22 ago
     Largest burst               736819
     Bit errors                  1782238*               3       1:15:22 ago
```

```
    Total bits                  144,626.240 Gb
    Bit error rate              > 1.23E-08*
    Bit errors since last lock  1147126*
    Total bits since last lock  144,471.783 Gb
    BER since last lock         > 7.94E-09*
  Lane 2
    Bit rate                    25.781 Gbps
    Lock state                  not locked              3       1:15:22 ago
    Largest burst               866185
    Bit errors                  1760538*                3       1:15:22 ago
    Total bits                  144,626.230 Gb
    Bit error rate              > 1.22E-08*
    Bit errors since last lock  894353*
    Total bits since last lock  144,471.773 Gb
    BER since last lock         > 6.19E-09*
  Lane 3
    Bit rate                    25.781 Gbps
    Lock state                  not locked              3       1:15:22 ago
    Largest burst               923941
    Bit errors                  1817413*                3       1:15:22 ago
    Total bits                  144,626.250 Gb
    Bit error rate              > 1.26E-08*
    Bit errors since last lock  893472*
    Total bits since last lock  144,471.793 Gb
    BER since last lock         > 6.18E-09*
```

11.4.7.47 show link tracking group

The **show link tracking group** command displays information about a specified link-state group or about all groups.

Command Mode

EXEC

Command Syntax

show link tracking group [**DATA_LEVEL**][**GROUPS**]

Parameters

- **DATA_LEVEL** device for which the command provides data. Options include:
  - ***no parameter*** information about all groups in group list.
  - **detail** detailed information about all groups in group list.
- **GROUPS**
  - ***no parameter*** all link-state groups.
  - ***group_name*** link-state group name.

Example

This command displays all the link-state group information.

```
switch# show link tracking group detail
Link State Group: 1 Status: up
Upstream Interfaces : Vlan100
Downstream Interfaces : Vlan200
Number of times disabled : 2
Last disabled 0:10:29 ago

Link State Group: group3 Status: down
Upstream Interfaces : Ethernet24
Downstream Interfaces : Ethernet8
Number of times disabled : 2
Last disabled 0:30:35 ago

Link State Group: 2 Status: up
Upstream Interfaces : Ethernet2 Ethernet5
```

```
Downstream Interfaces : Ethernet12
Number of times disabled : 0
Last disabled never switch#
```

11.4.7.48 show mac address-table

The **show mac-address-table** command displays the specified MAC address table entries.

Command Mode

EXEC

Command Syntax

show mac address-table [**ENTRY_TYPE**][**MAC_ADDR**][**INTF_1 ... INTF_N**][**VLANS**]

Parameters

- **ENTRY_TYPE** command filters display by entry type. Entry types include mlag-peer, dynamic, static, unicast, multicast entries, and configured.
  - ***no parameter*** all table entries.
  - **configured** static entries; includes unconfigured VLAN entries.
  - **dynamic** entries learned by the switch.
  - **static** entries entered by CLI commands and include a configured VLAN.
  - **unicast** entries with unicast MAC address.
- **MAC_ADDR** command uses MAC address to filter displayed entries.
  - ***no parameter*** all MAC addresses table entries.
  - **address *mac_address*** displays entries with specified address (dotted hex notation – H.H.H).
- **INTF_X** command filters display by port list. When parameter lists multiple interfaces, command displays all entries containing at least one listed interface.
  - ***no parameter*** all Ethernet and port channel interfaces.
  - **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
  - **port-channel *p_range*** Port channel interfaces specified by ***p_range***.
- **VLANS** command filters display by VLAN.
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** VLANs specified by ***v_num***.

Related Commands

- show mac address-table mlag-peer
- show mac address-table multicast

Examples

- This command displays the MAC address table.

```
switch> show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
 101    001c.8224.36d7    DYNAMIC     Po2        1       9 days, 15:57:28 ago
 102    001c.8220.1319    STATIC      Po1
 102    001c.8229.a0f3    DYNAMIC     Po1        1       0:05:05 ago
 661    001c.8220.1319    STATIC      Po1
 661    001c.822f.6b22    DYNAMIC     Po7        1       0:20:10 ago
3000    001c.8220.1319    STATIC      Po1
3000    0050.56a8.0016    DYNAMIC     Po1        1       0:07:38 ago
3902    001c.8220.1319    STATIC      Po1
3902    001c.822b.a80e    DYNAMIC     Po4        2       9 days, 15:57:30 ago
3903    001c.8220.1319    STATIC      Po1
```

```
3903    001c.822c.3009    DYNAMIC     Po5        1       4 days, 15:13:03 ago
3908    001c.8220.1319    STATIC      Po1
3908    001c.822c.4e1d    DYNAMIC     Po1        1       0:07:26 ago
3908    001c.822c.55d9    DYNAMIC     Po1        1       0:04:33 ago
3909    001c.8220.1319    STATIC      Po1
3909    001c.822f.6a80    DYNAMIC     Po1        1       0:07:08 ago
3910    001c.730f.6a80    DYNAMIC     Et9        1       4 days, 15:13:07 ago
3911    001c.8220.1319    STATIC      Po1
3911    001c.8220.40fa    DYNAMIC     Po8        1       1:19:58 ago
3912    001c.822b.033e    DYNAMIC     Et11       1       9 days, 15:57:23 ago
3913    001c.8220.1319    STATIC      Po1
3913    001c.822b.033e    DYNAMIC     Po1        1       0:04:35 ago
3984    001c.8220.178f    DYNAMIC     Et8        1       4 days, 15:07:29 ago
3992    001c.8220.1319    STATIC      Po1
3992    001c.8221.07b9    DYNAMIC     Po6        1       4 days, 15:13:15 ago
Total Mac Addresses for this criterion: 25

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0 switch>
```

- This command displays the MAC address learning status on ***vlan 10***.

```
switch(config)# vlan 10
switch(config-vlan-10)# no mac address learning
switch(config-vlan-10)# show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
Total Mac Addresses for this criterion: 0

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

VLANs with disabled MAC learning: 10
```

11.4.7.49 show mac address-table count

The **`show mac-address-table count`** command displays the number of entries in the MAC address table for the specified VLAN or for all VLANs.

Command Mode

EXEC

Command Syntax

`show mac address-table count` [**VLANS**]

Parameters

VLANS The VLANs for which the command displays the entry count.

- ***no parameter*** all configured VLANs.
- **vlan *v_num*** VLAN interface specified by ***v_num***.

Example

This command displays the number of entries on ***VLAN 39***.

```
switch> show mac address-table count vlan 39

Mac Entries for Vlan 39:
---------------------------
Dynamic Address Count             : 1
Unicast Static  Address Count     : 1
Multicast Static  Address Count   : 0
Total Mac Addresses               : 2 switch>
```

11.4.7.50 show mac address-table mlag-peer

The **`show mac-address-table mlag-peer`** command displays the specified MAC address table entries learned from the MLAG peer switch.

Command Mode

EXEC

Command Syntax

`show mac address-table mlag-peer` [**ENTRY_TYPE**][**MAC_ADDR**][**INTF_1 ... INTF_N**][**VLANS**]

Parameters

- **ENTRY_TYPE** command filters display by entry type. Entry types include mlag-peer, dynamic, static, unicast, multicast entries, and configured.
  - ***no parameter*** all MLAG peer entries.
  - **configured** static entries on MLAG peer; includes unconfigured VLAN entries.
  - **dynamic** entries learned on MLAG peer.
  - **static** MLAG entries entered by CLI commands and include a configured VLAN.
  - **unicast** MLAG entries with unicast MAC address.
- **MAC_ADDR** command uses MAC address to filter displayed entries.
  - ***no parameter*** all MAC addresses table entries.
  - **address *mac_address*** displays entries with specified address (dotted hex notation – H.H.H).
- **INTF_X** command filters display by port list. When parameter lists multiple interfaces, command displays all entries containing at least one listed interface.
  - ***no parameter*** all Ethernet and port channel interfaces.
  - **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
  - **port-channel *p_range*** Port channel interfaces specified by ***p_range***.
- **VLANS** command filters display by VLAN.
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** VLANs specified by ***v_num***.

Related Commands

- show mac address-table
- show mac address-table multicast

11.4.7.51 show mac address-table multicast

The **`show mac-address-table`** command displays the specified multicast MAC address table entries.

Command Mode

EXEC

Command Syntax

show mac address-table multicast [**MAC_ADDR**][**INTF**][**VLANS**]

Parameters

- **MAC_ADDR** command uses MAC address to filter displayed entries.
  - ***no parameter*** all MAC addresses table entries.
  - **address *mac_address*** displays entries with specified address (dotted hex notation – H.H.H).
- **INTF** command filters display by port list. When parameter lists multiple interfaces, command displays all entries containing at least one listed interface.
  - ***no parameter*** all Ethernet and port channel interfaces.
  - **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
  - **port-channel *p_range*** Port channel interfaces specified by ***p_range***.
- **VLANS** command filters display by VLAN.
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** VLANs specified by ***v_num***.

Related Commands

- show mac address-table
- show mac address-table multicast brief

11.4.7.52 show mac address-table multicast brief

The **show mac-address-table** command displays a summary of multicast MAC address table entries.

Command Mode

EXEC

Command Syntax

show mac address-table multicast [**VLANS**] **brief**

Parameters

VLANS command filters display by VLAN.

- ***no parameter*** all VLANs.
- **vlan *v_num*** VLANs specified by ***v_num***.

Related Command

show mac address-table multicast.

11.4.7.53 show monitor server-failure

The **show monitor server-failure** command displays Rapid Automated Indication of Link-loss (RAIL) configuration settings and the number of servers on each monitored network.

Command Mode

EXEC

Command Syntax

show monitor server-failure

Example

This command displays RAIL configuration status and lists the number of servers that are on each monitored network.

```
switch> show monitor server-failure
Server-failure monitor is enabled
Proxy service: disabled
Networks being monitored: 3
   10.2.1.96/28      : 0 servers
   10.1.1.0/24       : 0 servers
   10.3.0.0/16       : 3 servers switch>
```

11.4.7.54 show monitor server-failure history

The **show monitor server-failure history** command displays the time of all link failures detected by Rapid Automated Indication of Link-loss (RAIL) and includes the interface name for each failure.

The history is cleared by removing RAIL from the switch (**no monitor server-failure**).

Command Mode

EXEC

Command Syntax

show monitor server-failure history

Related Command

clear server-failure servers inactive

Example

This command displays the Fast Server Failure link failure history from the time RAIL is instantiated on the switch.

```
switch> show monitor server-failure history
Total server failures: 4

Server IP   Server MAC          Interface       Last Failed
----------- -----------------   -----------     -------------------
10.1.67.92  01:22:ab:cd:ee:ff   Ethernet17      2013-02-02 11:26:22
44.11.11.7  ad:3e:5f:dd:64:cf   Ethernet23      2013-02-10 00:07:56
10.1.1.1    01:22:df:42:78:cd   Port-Channel6   2013-02-09 19:36:09
10.1.8.13   01:33:df:ee:39:91   Port-Channel5   2013-02-10 00:03:39 switch>
```

11.4.7.55 show monitor server-failure servers

The **show monitor server-failure servers** command displays status and configuration information about each server that RAIL is monitoring. The display format depends on the parameter specified by the command:

- **single IP address:** command displays information about the server at the specified address, including IP address, MAC address, RAIL state, the time of most recent entry of all RAIL states, and the number of failed, proxied, and inactive state entries.
- **no parameter, key specifying a server list:** command displays a table. Each row corresponds to a monitored server. Information that the command displays includes IP address, MAC address, RAIL state, the time of most recent link failure.

Command Mode

EXEC

Command Syntax

show monitor server-failure servers [**SERVER_LIST**]

Parameters

SERVER_LIST Servers for which command displays information. Valid options include:

- ***no parameter*** all servers in up, down, and proxying states.
- ***ipv4_addr*** individual server; command displays detailed information.
- **all** all servers on monitored networks.
- **inactive** all servers in inactive state.
- **proxying** all servers in proxying state.

Examples

- This command displays RAIL information for the server at IP address ***10.11.11.7***.

```
switch> show monitor server-failure servers 10.11.11.7
Server information:
Server Ip Address        : 10.11.11.7
MAC Address              : ad:3e:5f:dd:64:cf
Current state            : down
Interface                : Ethernet23
Last Discovered          : 2013-01-06 06:47:39
Last Failed              : 2013-02-10 00:07:56
Last Proxied             : 2013-02-10 00:08:33
Last Inactive            : 2013-02-09 23:52:21
Number of times failed   : 3
Number of times proxied  : 1
Number of times inactive : 18 switch>
```

- This command displays RAIL data for all servers in monitored networks that are in inactive state.

```
switch> show monitor server-failure servers inactive
Inactive servers: 1

Server IP   Server MAC         Interface    State     Last Failed
----------  -----------------  -----------  --------  -------------
10.1.67.92  01:22:ab:cd:ee:ff  Ethernet17   inactive  7 days, 12:48:06 ago
 switch>
```

- This command displays RAIL information for all servers in monitored networks that are in up, down, and proxying states.

```
switch> show monitor server-failure servers
Active servers: 4

Server IP   Server MAC            Interface         State        Last Failed
----------  -----------------     --------------    ---------    -----------
44.11.11.7  ad:3e:5f:dd:64:cf     Ethernet23        down         0:03:21 ago
10.1.1.1    01:22:df:42:78:cd     Port-Channel6     up           4:35:08 ago
10.1.8.13   01:33:df:ee:39:91     Port-Channel5     proxying     0:07:38 ago
132.23.23.1 00:11:aa:bb:32:ad     Ethernet1         up           never
 switch>
```

- This command displays RAIL information for all servers on configured interfaces.

```
switch >show monitor server-failure servers all
Total servers monitored: 5

Server IP   Server MAC         Interface       State Last Failed
----------  -----------------  --------------  ---------  -----------
10.1.67.92  01:22:ab:cd:ee:ff  Ethernet17      inactive   7 days, 12:47:48
 ago
```

```
44.11.11.7  ad:3e:5f:dd:64:cf  Ethernet23      down       0:06:14 ago
10.1.1.1    01:22:df:42:78:cd  Port-Channel6   up         4:38:01 ago
10.1.8.13   01:33:df:ee:39:91  Port-Channel5   proxying   0:10:31 ago
132.23.23.1 00:11:aa:bb:32:ad  Ethernet1       up         never switch>
```

11.4.7.56 show monitor session

The **show monitor session** command displays the configuration of the specified port mirroring session. The command displays the configuration of all mirroring sessions on the switch when the session name parameter is omitted.

Command Mode

EXEC

Command Syntax

show monitor session **SESSION_NAME**

Parameters

SESSION_NAME Port mirroring session identifier. Options include:

- ***no parameter*** displays configuration for all sessions.
- ***label*** command displays configuration of the specified session.

Example

This command displays the mirroring configuration of the specified monitor session.

```
switch> show monitor session redirect_1

Session redirect_1
------------------------

Source Ports

  Both:        Et7

Destination Port: Et8
switch(config)>
```

11.4.7.57 show platform trident mirroring

The **show platform trident mirroring** command displays current parameters of all configured mirroring sessions in Trident series platforms.

Command Mode

Privileged EXEC

Command Syntax

show platform trident mirroring [detail | session]

Parameters

- **detail** displays the detailed information of all configured mirroring sessions.
- **session *session_name*** displays the information of specified mirroring session.

Guidelines

This command is supported on DCS-7050/7050X, DCS-7250X, and DCS-7300X devices only.

Examples

- This command displays the detailed information of all configured mirroring sessions.

```
switch(config)# show platform trident mirroring detail

Session : 123
=========================

srcIntf(rx): Ethernet12/3
Hw Mirror Id: 0x1

IM_MTP_INDEX
------------
count: 1
Dest: Et15/1

EGR_IM_MTP_INDEX
----------------
DestPort[ 0 ]: Et15/1
 Encap Enable: 0

srcIntf(tx): Ethernet12/3
Hw Mirror Id: 0x2

EM_MTP_INDEX
------------
count: 1
Dest: Et15/1

EGR_EM_MTP_INDEX
----------------
DestPort[ 0 ]: Et15/1

Session : abc
=========================

srcIntf(rx): Ethernet24/2
Hw Mirror Id: 0x0

IM_MTP_INDEX
------------
count: 1
Dest: Et24/4

EGR_IM_MTP_INDEX
----------------
DestPort[ 0 ]: Et24/4
 Encap Enable: 0

switch(config)#
```

- This command displays the information of session ***123***.

```
switch(config)# show platform trident mirroring session 123

Session         SrcIntf      Acl             DestIntf NextHopMac
 OutIntf
=======         =======      ===             ======== ==========
 =======
123             Et12/3(rx)                   Et15/1
                Et12/3(tx)                   Et15/1

switch(config)#
```

11.4.7.58 show port-channel load-balance

The **show port-channel load-balance** command displays the traffic distribution between the member ports of the specified port channels. The command displays distribution for unicast, multicast, and broadcast streams.

The distribution values displayed are based on the total interface counters which start from zero at boot time or when the counters are cleared. For more current traffic distribution values, clear the interface counters of the member interfaces using the clear counters command.

Command Mode

EXEC

Command Syntax

show port-channel load-balance [**MEMBERS**]

Parameters

MEMBERS List of port channels for which information is displayed. Options include:

- ***no parameter*** All configured port channels.
- ***c_range*** Ports in specified channel list (number, number range, or list of numbers and ranges).

Example

This command displays traffic distribution for all configured port channels.

```
switch> show port-channel load-balance
ChanId Port      Rx-Ucst Tx-Ucst Rx-Mcst Tx-Mcst Rx-Bcst Tx-Bcst
------ --------- ------- ------- ------- ------- ------- -------
8      Et10      100.00% 100.00% 100.00% 100.00% 0.00%   100.00%
------ --------- ------- ------- ------- ------- ------- -------
1      Et1       13.97%  42.37%  47.71%  30.94%  0.43%   99.84%
1      Et2       86.03%  57.63%  52.29%  69.06%  99.57%  0.16%
------ --------- ------- ------- ------- ------- ------- -------
2      Et23      48.27%  50.71%  26.79%  73.22%  0.00%   100.00%
2      Et24      51.73%  49.29%  73.21%  26.78%  0.00%   0.00%
------ --------- ------- ------- ------- ------- ------- -------
4      Et3       55.97%  63.29%  51.32%  73.49%  0.00%   0.00%
4      Et4       44.03%  36.71%  48.68%  26.51%  0.00%   0.00%
------ --------- ------- ------- ------- ------- ------- -------
5      Et19      39.64%  37.71%  50.00%  90.71%  0.00%   0.00%
5      Et20      60.36%  62.29%  50.00%  9.29%   0.00%   100.00%
------ --------- ------- ------- ------- ------- ------- -------
6      Et6       100.00% 100.00% 100.00% 100.00% 0.00%   100.00%
------ --------- ------- ------- ------- ------- ------- -------
7      Et5       100.00% 0.00%   100.00% 100.00% 0.00%   0.00%
switch>
```

11.4.7.59 show port-security

The **show port-security** command displays a summary of MAC address port security configuration and status on each interface where switchport port security is enabled.

Command Mode

EXEC

Command Syntax

show port-security

Display Values

Each column corresponds to one physical interface. The table displays interfaces with port security enabled.

- **Secure Port:** Interface with switchport port-security enabled.
- **MaxSecureAddr:** Maximum quantity of MAC addresses that the specified port can process.

- **`CurrentAddr:`** Static MAC addresses assigned to the interface.
- **`SecurityViolation:`** Number of frames with unsecured addresses received by port.
- **`Security Action:`** Action triggered by a security violation.

These are the value displayed by the command.

- **`Aging Time:`** Age of Mac address.
- **`MAC Moveable:`** Mac address movement.
- **`Port Security:`** Enabled or disabled status

Examples

- This command displays switchport port security configuration and status data.

```
switch> show port-security
Secure Port    MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
               (Count)        (Count)      (Count)
--------------------------------------------------------------------------
     Et7             5             3            0            Shutdown
     Et10            1             0            0            Shutdown
--------------------------------------------------------------------------
Total Addresses in System: 3 switch>
```

- From ***EOS Release 4.26.0F***, **`show port-security`** command displays the settings for the new global port security configurations, including MAC aging, MAC moves, and persistent port security.

```
switch(config)# show port-security
Secure address moves: disabled
Secure address aging: disabled
Secure address reboot persistence: enabled
Secure address link down persistence: enabled
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)          (Count)
--------------------------------------------------------------------------
--------------------------------------------------------------------------
Total Addresses in System: 0
```

11.4.7.60 show port-security interface

The **`show port-security interface`** command displays the switchport port-security status of all specified interfaces.

Command Mode

EXEC

Command Syntax

`show port-security interface` [**INT_NAME**]

Parameters

INT_NAME Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **loopback *l_range*** Loopback interface specified by ***l_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
- **vlan *v_range*** VLAN interface range specified by ***v_range***.
- **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.

  Valid ***range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command display port-security configuration and status for the specified interfaces.

```
switch> show port-security interface ethernet 7-8
Interface                  : Ethernet7
Port Security              : Enabled
Port Status                : Secure-down
Violation Mode             : Shutdown
Maximum MAC Addresses      : 5
Aging Time                 : 5 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled
Total MAC Addresses        : 3
Configured MAC Addresses   : 3
Learn/Move/Age Events      : 5
Last Source Address:Vlan   : 164f.29ae.4e14:10
Last Address Change Time   : 0:39:47 ago
Security Violation Count   : 0
Interface                  : Ethernet8
Port Security              : Disabled
Port Status                : Secure-down
Violation Mode             : Shutdown
Maximum MAC Addresses      : 1
Aging Time                 : 5 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled switch>
```

11.4.7.61 show port-security mac-address

The **show port-security mac-address** command display static unicast MAC addresses assigned to interfaces where switchport port security is enabled.

Command Mode

EXEC

Command Syntax

show port-security mac-address

Example

This command displays MAC addresses assigned to port-security protected interfaces.

```
switch> show port-security mac-address
Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                Ports   Remaining Age
                                                              (mins)
----    -----------       ----------------    -----   -------------
  10    164f.29ae.4e14    SecureConfigured     Et7     N/A
  10    164f.29ae.4f11    SecureConfigured     Et7     N/A
  10    164f.320a.3a11    SecureConfigured     Et7     N/A
------------------------------------------------------------------------
Total Mac Addresses for this criterion: 3 switch>
```

11.4.7.62 show storm-control

The **show storm-control** command displays the storm-control level and interface inbound packet capacity for the specified interface.

The configured value (storm-control) differs from the programmed threshold in that the hardware accounts for Interframe Gaps (IFG) based on the minimum packet size. This command displays the broadcast or multicast rate after this adjustment.

Command Mode

Privileged EXEC

Command Syntax

`show storm-control` [**INT_NAME**]

Parameters

- ***no parameter*** Command returns data for all interfaces configured for storm control.
- **INT_NAME** interface type and port range. Settings include:
  - **ethernet *e_range*** Ethernet interfaces that ***e_range*** denotes.
  - **port-channel *p_range*** Port channel interfaces that ***p_range*** denotes.

    When storm control commands exist for a port-channel and an Ethernet port that is a member of the port channel, the command for the port-channel takes precedence.

    Valid ***range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the storm control configuration for ***ethernet port 2*** through ***ethernet port 4***.

```
switch# show storm-control
Port           Type  Level Rate(Mbps)   Status      Drops Reason
Et10/2          all     75       7500   active          0
Et10/3   multicast     55       5500   active          0
Et10/4   broadcast     50       5000   active          0 switch#
```

11.4.7.63 show switch forwarding-mode

The **`show switch forwarding-mode`** command displays the switch's current and available forwarding plane hardware modes.

Command Mode

EXEC

Command Syntax

`show switch forwarding-mode`

Related Command

switch forwarding-mode configures the switch's forwarding mode setting.

Example

This command changes the switch's forward mode to **store-and-forward**, then displays the forwarding mode.

```
switch(config)# switch forwarding-mode store-and-forward
switch(config)# show switch forwarding-mode
Current switching mode:    store and forward
Available switching modes: cut through, store and forward
```

11.4.7.64 show track

The **`show track`** command displays information about tracked objects configured on the switch.

Command Mode

EXEC

Command Syntax

show track [**OBJECT**][**INFO_LEVEL**]

Parameters

- **OBJECT** tracked object for which information is displayed. Options include:
  - ***no parameter*** displays information for all tracked objects configured on the switch.
  - ***object_name*** displays information for the specified object.
- **INFO_LEVEL** amount of information that is displayed. Options include:
  - ***no parameter*** displays complete information including object status, number of status changes, time since last change, and client process tracking the object (if any).
  - **brief** displays brief list of all tracked objects and their current status.

Examples

- This command displays all information for tracked object ***ETH8***.

```
switch# show track ETH8
Tracked object ETH8 is up
   Interface Ethernet8 line-protocol
      4 change, last change time was 0:36:12 ago
   Tracked by:
      Ethernet5/1 vrrp instance 50 switch#
```

- This command displays summary information for all tracked objects.

```
switch# show track brief
Tracked object ETH2 is up
Tracked object ETH4 is down
Tracked object ETH6 is up
Tracked object ETH8 is up switch#
```

11.4.7.65 shutdown (server-failure configuration mode)

The **shutdown** command disables Rapid Automated Indication of Link-Loss (RAIL). By default, RAIL is disabled.

After entering server-failure configuration mode, a **no shutdown** command is required to enable RAIL.

The **no shutdown** command enables RAIL on the switch. The **shutdown** and **default shutdown** commands disable RAIL by removing the shutdown command from ***running-config***.

Command Mode

Server-failure Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- This command enables RAIL on the switch.

```
switch(config)# monitor server
switch(config-server-failure)# no shutdown
switch(config-server-failure)#  show active
```

```
monitor server-failure
   no shutdown
switch(config-server-failure)#
```

- This command disables RAIL on the switch.

```
switch(config-server-failure)# shutdown
switch(config-server-failure)# show active
monitor server-failure
switch(config-server-failure)#
```

11.4.7.66 storm-control

The **storm-control** command configures and enables storm control on the configuration mode physical interface. The command provides three mode options:

- **storm-control all** unicast, multicast, and broadcast inbound packet control.
- **storm-control broadcast** broadcast inbound packet control.
- **storm-control multicast** multicast inbound packet control.

An interface configuration can contain three storm-control statements, one with each mode setting. The **storm-control all** threshold overrides broadcast and multicast thresholds.

The threshold is a percentage of the available port bandwidth and is configurable on each interface for each transmission mode.

The **no storm-control** and **default storm-control** commands remove the corresponding **storm-control** statement from ***running-config***, disabling storm control for the specified transmission type on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

storm-control MODE level ***threshold***

no storm-control MODE

default storm-control MODE

Parameters

- **MODE** packet transmission type. Options include:
  - **all**
  - **broadcast**
  - **multicast**
- ***threshold*** Inbound packet level that triggers storm control, as a percentage of port capacity. Value ranges from ***0.01*** to ***100***. Storm control is suppressed by a level of ***100***.

  The configured value differs from the programmed threshold in that the hardware accounts for InterFrame Gaps (IFG) based on the minimum packet size. The show storm-control command displays the broadcast or multicast rate after this adjustment.

Restrictions

The **storm-control all** option is not available on Arad platform switches.

Example

These commands enable multicast and broadcast storm control on ***Ethernet port 20*** and sets thresholds of ***65%*** (multicast) and ***50%*** (broadcast). During each one second interval, the interface drops inbound multicast traffic and broadcast traffic in excess of the specified thresholds.

```
switch(config)# interface ethernet 20
switch(config-if-Et20)# storm-control multicast level 65
switch(config-if-Et20)# storm-control broadcast level 50
switch(config-if-Et20)# show active
interface Ethernet20
   storm-control broadcast level 50
   storm-control multicast level 65
switch(config-if-Et20)#
```

11.4.7.67 switch forwarding-mode

The **`switch forwarding-mode`** command specifies the mode of the switch's forwarding plane hardware. The default forwarding mode is **cut through**.

The **`no switch forwarding-mode`** and **`default switch forwarding-mode`** commands restore the default forwarding mode by removing the **`switch forwarding-mode`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`switch forwarding-mode` **MODE_SETTING**

`no switch forwarding-mode`

`default switch forwarding-mode`

Parameters

MODE_SETTING Specifies the switch's forwarding plane hardware mode. Options include:

- **cut-through** the switch begins forwarding frames before their reception is complete.
- **store-and-forward** the switch accumulates entire packets before forwarding them.

Guidelines

The forwarding plane mode is **store-and-forward** on Petra and Arad platform switches.

Related Command

show switch forwarding-mode displays the current forwarding mode.

Example

This command changes the forwarding mode to **store-and-forward**.

```
switch(config)# switch forwarding-mode store-and-forward
switch(config)#
```

11.4.7.68 switchport

The **`switchport`** command places the configuration mode interface in ***switched port*** (Layer 2) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments.

The no switchport command places the configuration mode interface in ***routed port*** (Layer 3) mode. Routed ports are not members of any VLANs and do not switch or bridge packets. All IP level configuration commands, including IP address assignments, apply directly to the routed port interface.

By default, Ethernet and Port Channel interfaces are in switched port mode. The **default switchport** command also places the configuration mode interface in switched port mode by removing the corresponding **no switchport** command from ***running-config***.

These commands only toggle the interface between switched and routed modes. They have no effect on other configuration states.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport

no switchport

default switchport

Guidelines

When an interface is configured as a routed port, the switch transparently allocates an internal VLAN whose only member is the routed interface. Internal VLANs are created in the range from ***1006*** to ***4094***. VLANs that are allocated internally for a routed interface cannot be directly created or configured. The **vlan internal order** command specifies the method that VLANs are allocated.

All IP-level configuration commands, except **autostate** and **ip virtual-router**, can be used to configure a routed interface. Any IP-level configuration changes made to a routed interface are maintained when the interface is toggled to switched port mode.

A LAG that is created with the **channel-group** command inherits the mode of the member port. A LAG created from a routed port becomes a routed LAG. IP-level configuration statements are not propagated to the LAG from its component members.

Examples

- These commands put ***interface ethernet 5*** in routed port mode.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no switchport
switch(config-if-Et5)#
```

- These commands returns ***interface ethernet 5*** to switched port mode.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport
switch(config-if-Et5)#
```

11.4.7.69 switchport default mode access

The **switchport default mode access** command places the configuration mode interface in ***switched port default access*** (Layer 3) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments.

Command Mode

Global Configuration

Command Syntax

switchport default mode access

Related Command

switchport default mode routed puts a switch with all ports in routed port mode.

Example

This command puts a switch with all ports in access port mode.

```
switch(config)# switchport default mode access
```

11.4.7.70 switchport default mode routed

The **switchport default mode routed** command places the configuration mode interface in ***switched port default routed*** (Layer 3) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments.

By default, on a switch with default startup config or no config, all ports come up in access mode. By adding the CLI command **switchport default mode routed** to kickstart config, all ports will come up in routed mode after boot up. On boot up, Zero Touch Provisioning (ZTP) is enabled by default if the startup config (**/mnt/flash/startupconfig**) is deleted. ZTP can be disabled by setting DISABLE=True in ZTP config (**/mnt/flash/zerotouchconfig**). Kickstart config (/mnt/flash/kickstart-config) is used when startup config is missing and ZTP is disabled.

Command Mode

Global Configuration

Command Syntax

switchport default mode routed

Related Command

switchport default mode access puts a switch with all ports in access port mode.

Example

This command puts a switch with all ports in routed port mode.

```
switch(config)# switchport default mode routed
```

11.4.7.71 switchport mac address learning

The **switchport mac address learning** command enables MAC address learning for the configuration mode interface. MAC address learning is enabled by default on all Ethernet and port channel interfaces.

The switch maintains a MAC address table for switching frames between VLAN ports. When the switch receives a frame, it associates the MAC address of the transmitting interface with the recipient VLAN and port. When MAC address learning is enabled for the recipient port, the entry is added to the MAC address table. When MAC address learning is not enabled, the entry is not added to the table.

The **no switchport mac address learning** command disables MAC address learning for the configuration mode interface. The **switchport mac address learning** and **default switchport mac address learning** commands enable MAC address learning for the configuration mode interface by deleting the corresponding **no switchport mac address learning** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport mac address learning

no switchport mac address learning

default switchport mac address learning

Example

These commands disables MAC address learning for ***interface ethernet 8***, then displays the active configuration for the interface.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# no switchport mac address learning
switch(config-if-Et8)# show active
interface Ethernet8
   no switchport mac address learning
switch(config-if-Et8)#
```

11.4.7.72 switchport port-security

The **switchport port-security** command enables MAC address port security on the configuration mode interface. Ports with port security enables restrict traffic to a limited number of hosts, as determined by their MAC addresses. On enabling the **switchport port-security** command, the port-security mode would be 'shutdown', by default.

The switchport port-security mac-address maximum command specifies the maximum number of MAC addresses. The switchport port-security violation command enables port security in protect mode.

The **no switchport port-security** and **default switchport port-security** commands disable port security on the configuration mode interface by removing the corresponding **switchport port-security** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport port-security

no switchport port-security

default switchport port-security

Example

These commands enable port security on ***interface ethernet 7***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# switchport port-security
switch(config-if-Et7)#
```

11.4.7.73 switchport port-security mac-address maximum

The **switchport port-security mac-address maximum** command specifies the maximum MAC address limit for the configuration mode interface when configured as a secure port. When port security is enabled, the port accepts traffic and adds source addresses to the MAC table until the maximum is reached. Once the maximum is reached, if any traffic arrives from a source not already in the MAC table for the secure port, the port becomes errdisabled. The switchport port-security command configures an interface as a secure port.

The **no switchport port-security mac-address maximum** and **default switchport port-security mac-address maximum** commands restore the maximum MAC address limit of one on the

configuration mode interface by removing the corresponding **switchport port-security mac-address maximum** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport port-security mac-address maximum ***max_addr***

no switchport port-security mac-address maximum

default switchport port-security mac-address maximum

Parameters

max_addr maximum number of MAC addresses. Value ranges from ***1*** to ***1000***. Default value is ***1***.

Example

These commands configure a maximum of five incoming addresses for secure ***interface port-channel 14***.

```
switch(config)# interface port-channel 14
switch(config-if-Po14)# switchport port-security mac-address maximum 5
switch(config-if-Po14)#
```

11.4.7.74 switchport port-security violation

The **switchport port-security violation** command configures port security in protect mode (with the option of enabling logging) or the shutdown mode.

The **no switchport port-security** and **no switchport port-security violation protect log** commands disable port security protect mode and port security protect mode logging on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport port-security violation {protect [log]| shutdown}

no switchport port-security violation protect log

default switchport port-security violation protect log

Parameters

- **protect** configures the port security in the protect mode.
- **shutdown** configures the port security in the shutdown mode.
- **log** the log of new addresses seen after limit is reached in the protect mode.

Guidelines

When port security is enabled, the port accepts traffic and adds source addresses to the MAC table until the maximum is reached. The **switchport port-security** command configures an interface as a secure port.

In the protect mode, the ACLs are dynamically created to block incoming MAC addresses when the configured maximum MAC value is reached.

In the shutdown mode, once the maximum is reached, if any traffic arrives from a source not already in the MAC table for the secure port, the port is set to be errdisabled.

Examples

- These commands configure port security violation protect mode for ***secure port channel interface 14***.

```
switch(config)# interface port-channel 14
switch(config-if-Po14)# switchport port-security violation protect
switch(config-if-Po14)#
```

- These commands configure port security violation protect logging mode for ***secure port channel interface 14***.

```
switch(config)# interface port-channel 14
switch(config-if-Po14)# switchport port-security violation protect log
switch(config-if-Po14)#
```

- These commands configure port security violation shutdown mode for ***secure port channel interface 15***.

```
switch(config)# interface port-channel 15
switch(config-if-Po15)# switchport port-security violation shutdown
switch(config-if-Po15)#
```

11.4.7.75 system control-plane

The **`system control-plane`** command places the switch in ***control-plane*** configuration mode. Control-plane mode is used for assigning an ACL (access control list) to the control plane.

The ***control-plane*** configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. Exiting ***control-plane*** configuration mode does not affect the configuration.

The **`exit`** command returns the switch to ***global*** configuration mode.

Command Mode

Global Configuration

Command Syntax

`system control-plane`

Command Available in control-plane Configuration Mode

ip access-group (Control Plane mode)

Examples

- This command places the switch in the ***control plane*** mode.

```
switch(config)# system control-plane
switch(config-system-cp)#
```

- This command assigns the ***control-plane-2*** ACL to the control plane.

```
switch(config-system-cp)# ip access-group control-plane-2
switch(config-system-cp)#
```

- This command exits the ***control plane*** mode.

```
switch(config-system-cp)# exit
switch(config)#
```

11.4.7.76 track

The **track** command creates an object whose state changes to provide information to a client process. The client process must be separately configured for object tracking to have an effect on the switch.

The **no track** and **default track** commands remove the specified tracked object by removing the corresponding **track** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

track ***object_name*** interface **INTERFACE_NAME PROPERTY**

no track ***object_name***

default track ***object_name***

Parameters

- ***object_name*** User-created name for the tracked object.
- **INTERFACE_NAME** Interface associated with the tracked object. Options include:
  - **ethernet** ***e_num*** Ethernet interface specified by ***e_num***.
  - **loopback** ***l_num*** Loopback interface specified by ***l_num***.
  - **management** ***m_num*** Management interface specified by ***m_num***.
  - **port-channel** ***p_num*** Port-channel interface specified by ***p_num***.
  - **vlan** ***v_num*** VLAN interface specified by ***v_num***.
  - **vxlan** ***vx_num*** VXLAN interface specified by ***vx_num***.
- **PROPERTY** Tracked property. Options include:
  - **line-protocol** Object changes when the state of the associated interface changes.

Example

This command creates a tracked object which tracks the state of the line protocol on ***interface ethernet 8***.

```
switch(config)# track ETH8 interface ethernet 8 line-protocol
switch(config)#
```

11.4.7.77 traffic-loopback

The **traffic-loopback** command is used to create loopbacks to verify the functionality of interfaces and partner links. The **source** determines whether outgoing traffic is being looped back to the interface (**system**) to test the interface itself, or incoming traffic is being looped back to the link partner (**network**) to test the link between the systems. The **device** determines whether **system** traffic is looped on the physical level (**phy**) or Layer-2 level (**mac**). Only the **phy** level is available for **network** traffic.

The **no traffic-loopback** command deletes the loopback configuration.

Command Mode

Interface Configuration

Command Syntax

traffic-loopback source [system|network] device [phy|mac]

no traffic-loopback

Parameters

- **system** loops outgoing traffic back to the interface.
- **network** loops incoming traffic back to the link partner.

- **phy** implements loopback in the physical layer.
- **mac** implements loopback in the MAC layer (available only for **system** traffic).

Examples

- These commands cause outgoing traffic on ***interface ethernet 1*** to be looped back to the interface at the MAC level.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# traffic-loopback source system device mac
switch(config-if-Et1)#
```

- These commands cause incoming traffic on ***interface ethernet 1*** to be looped back to the link partner at the physical level.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# traffic-loopback source network device phy
switch(config-if-Et1)#
```

- These commands delete the loopback configuration from ***interface ethernet 1***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# no traffic-loopback
switch(config-if-Et1)#
```

11.5 Octal Port Renumber to Four Interfaces

For an octal port such as a QSFPDD or OSFP, this feature renumbers the ports on a system to have 4 configurable interfaces instead of eight. The original octal would have had eight interfaces named ***x/1*** to ***x/8*** where x is the front panel port number.

The four interfaces created by this feature are named ***x/1*** to ***x/4*** instead and each interface has two transceiver lanes attached to it. Thus each interface can be configured to two-lane, four-lane or eight-lanes speeds as appropriate. However, the interfaces are restricted from any single-lane speeds and are not capable of a full breakout.

The entire chassis must be renumbered. This feature is not supported on a per-line card or per-port basis. Any cards in the same chassis that do not support this feature are unaffected. For affected line cards, both the configurable speeds and the default speeds of all interfaces are affected.

Supported Speeds

Table 66: Supported Speeds on Octal Ports

Interface Name With Feature	Interface Name Without Feature	Supported Speeds with Feature	Default Speed
Ethernetx/1	Ethernetx/1	400G-8, 200G-4, 100G-2, 100G-4	400G-8
Nonexistent	Ethernetx/2	N/A	N/A
Ethernetx/2	Ethernetx/3	100G-2	100G-2
Nonexistent	Ethernetx/5	N/A	N/A
Ethernetx/3	Ethernetx/5	200G-4, 100G-2, 100G-4	100G-2
Nonexistent	Ethernetx/6	N/A	N/A
Ethernetx/4	Ethernetx/7	100G-2	100G-2
Nonexistent	Ethernetx/8	N/A	N/A

11.5.1 Configuring Octal Port Renumber to 4 Interfaces

In order to configure a chassis to octal port renumbering with four interfaces, a boot command is used.

The boot command is different from a standard CLI command since it takes effect very early in the boot process but can be configured in the same way. Therefore it requires a reboot after the CLI command in order to take effect.

Since the ports will be renumbered for all cards that support the feature, the current configuration of the switch may no longer make sense after the reboot. Please ensure all interface configurations reflect the new naming scheme. The following items should be addressed prior to enabling or disabling the feature and performing a reboot:

1. Change interface naming appropriately.
2. Review interface speeds to ensure they are supported with the feature.
3. Review change in interface defaults to ensure the feature's defaults are desired.
4. Review changes to SNMP OIDs for affected interfaces.

From the CLI configuration mode, use the following command to enable the feature:

```
switch(config)# boot port numbering octal interfaces 4
```

The command warns that the switch needs to be rebooted for the command to take effect. Reboot the box afterward.

From the CLI configuration mode, use the following command to disable the feature:

```
switch(config)# no boot port numbering octal interfaces 4
```

The command warns that the switch needs to be rebooted for the command to take effect. Reboot the box afterward.

11.5.2 Show Commands

To confirm the command has taken effect, expose inactive interfaces with the **`service interface inactive expose`** CLI command and confirm the affected linecards only have four interfaces for each front panel port.

```
switch#(config-if-Et16/1,16/3)# show interfaces et15/1-16/1 status
Port       Name   Status       Vlan     Duplex Speed  Type          Flags Encapsulation
Et15/1            connected    1        full   400G   400GBase-CR8
Et15/2            inactive     1        full   100G   Not Present
Et15/3            inactive     1        full   100G   Not Present
Et15/4            inactive     1        full   100G   Not Present
Et16/1            connected    1        full   400G   400GBase-CR8
```

11.5.3 Limitations

- A config replace of the startup config does not affect the configuration of this feature.
- Since the feature is stored in **`flash:boot-config`**, removing or overwriting the boot-config can affect the presence of the feature on the next reboot.

11.5.4 Commands

- **boot port numbering octal interfaces 4**

11.5.4.1 boot port numbering octal interfaces 4

The **`boot port numbering octal interfaces 4`** command configures a chassis to octal port renumbering with four interfaces.

The **`no boot port numbering octal interfaces 4`** command disable the ***boot port numbering octal interfaces 4*** configuration from the ***running-config***.

Command Mode

Global Configuration Mode

Command Syntax

`boot port numbering octal interfaces 4`

`no boot port numbering octal interfaces 4`

Example

To enable ***boot port numbering octal interfaces 4***, use the following command:

```
switch(config)#boot port numbering octal interfaces 4
```

Chapter 12

Layer 2 Configuration

This chapter contains the following sections:

- Spanning Tree Protocol
- Link Layer Discovery Protocol
- Virtual LANs (VLANs)
- DCBX and Flow Control
- IP Locking
- Layer 2 Protocol Forwarding
- Layer 2 Subinterfaces

12.1 Spanning Tree Protocol

Spanning Tree Protocols prevent bridging loops in Layer 2 Ethernet networks. Arista switches support Rapid Spanning Tree, Multiple Spanning Tree, and Rapid-Per VLAN Spanning Tree protocols.

These sections describe the Arista Spanning Tree Protocol implementation.

- Introduction to Spanning Tree Protocols
- Spanning Tree Overview
- Configuring a Spanning Tree
- STP Commands

12.1.1 Introduction to Spanning Tree Protocols

Arista Switches support the leading spanning tree protocols: RSTP, MST and Rapid-PVST. This variety of options simplifies integration into existing networks without compromising network reliability, scalability or performance.

12.1.2 Spanning Tree Overview

An Ethernet network functions properly when only one active path exists between any two stations. A spanning tree is a loop-free subset of a network topology. STP is a L2 network protocol that ensures a loop-free topology for any bridged Ethernet LAN. STP allows a network to include spare links as automatic backup paths that are available when an active link fails without creating loops or requiring manual intervention. The original STP is standardized as IEEE 802.1D.

Several variations to the original STP improve performance and add capacity. Arista switches support these STP versions:

- Rapid Spanning Tree (RSTP)
- Multiple Spanning Tree (MSTP)
- Rapid Per-VLAN Spanning Tree (Rapid-PVST)

The Overview contains the following sections:

- Spanning Tree Protocol Versions
- Structure of a Spanning Tree Instance
- Bridge Protocol Data Units (BPDUs)

12.1.2.1 Spanning Tree Protocol Versions

STP versions supported by Arista switches address two limitations of the original Spanning Tree protocol that was standardized as ***IEEE 802.1D***:

- Slow convergence to the new spanning tree topology after a network change.
- The entire network is covered by one spanning tree instance.

The following sections describe the supported STP versions, compatibility issues in networks containing switches running different STP versions, and supported alternatives to spanning tree.

12.1.2.1.1 Rapid Spanning Tree Protocol (RSTP)

RSTP is specified in ***802.1w*** and supersedes STP. RSTP provides rapid convergence after network topology changes. Similar to STP, RSTP provides a single instance of spanning tree for the entire network. Standard ***802.1D-2004*** incorporates RSTP and obsoletes STP.

The RSTP instance is the base unit of MST and Rapid-PVST spanning trees.

12.1.2.1.2 Rapid Per-VLAN Spanning Tree Protocol (Rapid-PVST)

Rapid Per-VLAN Spanning Tree (PVST) extends the original STP to support a spanning tree instance on each VLAN in the network. The maximum number of PVST instances that can be created on a switch depends on the hardware platform. In most of the cases, it is 510. PVST can load balance Layer-2 traffic without creating a loop because it handles each VLAN as a separate network. However, PVST does not address slow network convergence after a network topology change.

Arista switches support Rapid-PVST, which is a variation of PVST based on RSTP instances. Rapid-PVST provides rapid connectivity recovery after the failure of a bridge, port, or LAN. Rapid-PVST can be enabled or disabled on individual VLANs.

12.1.2.1.3 Multiple Spanning Tree Protocol (MSTP)

MST extends Rapid Spanning Tree Protocol (RSTP) to support multiple spanning tree instances on a network, but is still compatible with RSTP. By default, Arista switches use MSTP.

MST supports multiple spanning tree instances, similar to Rapid PVST. However, MST associates an instance with multiple VLANs. This architecture supports load balancing by providing numerous forwarding paths to data traffic. Network fault tolerance is improved because failures in one instance do not affect other instances.

MST Regions

An *MST region* is a group of connected switches with identical MST configuration. Each region can support a maximum of 65 spanning-tree instances. MST regions are identified by a version number, name, and VLAN-to-instance map. You must configure identical parameters on all switches in the regions. Only MST region members participate with the MST instances defined in the region. A VLAN can be simultaneously assigned to only one spanning-tree instance. MST does not specify the maximum number of regions that a network can contain.

MST Instances

Each MST instance is identified by an instance number that ranges from 0 to 4094 and is associated with a set of VLANs. An MST region contains two types of spanning tree instances: an Internal Spanning Tree Instance (IST) and Multiple Spanning Tree Instances (MSTI).

- IST is the default spanning tree instance in MST regions; and is always zero. It gives a root switch to the region that contains all VLANs configured across all switches in the region but not assigned to a MST instance. IST considers all interfaces regardless of the VLAN membership.

- **Multiple Spanning Tree instances (MSTIs)** consist of VLANs that are assigned through MST configuration statements. VLANs assigned to an MSTI are removed from the IST instance. VLANs in an MSTI operate as a part of a single Spanning Tree topology. Because each VLAN can belong to only one instance, MST instances (and the IST) are topologically independent.

MSTP-Rapid PVST+ Interoperation

MSTP-Rapid PVST+ interoperation allows protocol-level interaction between MSTP and Rapid PVST+ by consuming and transmitting PVST BPDUs through MSTP at the border ports. While transmitting from Rapid PVST+ to MSTP or vice-versa, you can deploy MSTP in certain regions and Rapid PVST+ in other regions based on their merits or other factors. MSTP-Rapid PVST+ interoperation allows MSTP to work with any Rapid PVST+ implementation.

MSTP inter-operation can be viewed as bi-directional BPDU conversion occurring at MSTP PVST+ border ports as following:

- The outgoing Common Internal Spanning Tree (CIST) BPDU is projected as Rapid PVST+ BPDUs across various VLANs.
- Incoming PVST+ topology BPDU is mapped to corresponding MSTI based on incoming VLAN.

The following restrictions are enforced based on port role. These are relevant only for border ports with roles designated or root (for CIST) that can potentially forward.

- Case (A): If the MSTP-Rapid PVST+ border port role is designated for CIST, it should not receive any superior PVST BPDU (with respect to CIST). Root of all VLANs in the PVST region should be towards the MSTP region.
- Case (B): If the MSTP-Rapid PVST+ border port role is root for CIST, it should not receive any inferior Rapid PVST+ BPDU (with respect to CIST). Root of all VLANs in the Rapid PVST+ region should be within the PVST+ region.



Note: If the restrictions are violated, the port enters blocking state. The BPDU translation occurs only on MSTP-Rapid PVST+ border ports with the forwarding state for CIST.

It is desirable to configure multiple MSTP-Rapid PVST+ border ports confirming to Case(A) for a specified Rapid PVST+ region as it allows VLAN load balancing across the ports. This can be achieved by adjusting the per-VLAN port cost towards the Rapid PVST+ region such that the per-VLAN root port is evenly distributed across the border ports.

12.1.2.1.4 Version Interoperability

A network can contain switches running different spanning tree versions. The Common Spanning Tree (CST) is a single forwarding path the switch calculates for STP, RSTP, MSTP, and Rapid-PVST topologies in networks containing multiple spanning tree variations.

In multi-instance topologies, the following instances correspond to the CST:

- **Rapid-PVST**: VLAN 1
- **MST**: IST (instance 0)

RSTP and MSTP are compatible with other spanning tree versions:

- An RSTP bridge sends 802.1D (original STP) BPDUs on ports connected to an STP bridge.
- RSTP bridges operating in 802.1D mode remain in 802.1D mode even after all STP bridges are removed from their links.
- An MST bridge detects a port is at a region boundary when it receives an STP BPDU or an MST BPDU from a different region.
- MST ports assume they are boundary ports when the bridges to which they connect join the same region.

The `clear spanning-tree detected-protocols` command forces MST ports to renegotiate with their neighbors.

12.1.2.1.5 Switchport Interface Pairs

Switchport interface pairs associate two interfaces in a primary-backup configuration. When the primary interface is functioning, the backup interface remains dormant in standby mode. When the primary interface stops functioning, the backup interface handles the traffic.

An alternative implementation balances traffic between the primary and backup interfaces. If either interface shuts down, the other handles traffic addressed to the pair.

The following guidelines apply to switchport interface pairs.

- Ethernet and Port Channels can be primary interfaces.
- Ethernet, Port Channel, Management, Loopback, and VLAN interfaces can be backup interfaces.
- The primary and backup interfaces can be different interface types.
- Interface pairs should be similarly configured to ensure consistent behavior.
- An interface can be associated with a maximum of one backup interface.
- An interface can back up a maximum of one interface.
- Any Ethernet interface configured in an interface pair cannot be a port channel member.
- STP is disabled on ports configured as primary or backup interfaces.
- Static MAC addresses should be configured after primary-backup pairs are established.

12.1.2.2 Structure of a Spanning Tree Instance

A Layer 2 network consists of bridges and network segments. A loop exists when multiple active paths connect two components. Spanning tree protocols allow only one active path between any two network components. Loops are removed by blocking selected ports that connect bridges to network segments.

Ports are assigned cost values that reflect their transmission speed and any other criteria selected by the administrator. Ports with faster transmission speeds and other desirable characteristics are assigned lower costs. High cost ports are blocked in deference to lower cost ports.

A network topology defines multiple possible spanning trees. Network bridges collectively compute and implement one spanning tree to maintain connectivity between all network components while blocking ports that could result in loops. Administrators improve network performance by adjusting parameter settings to select the most efficient spanning tree.

Spanning tree bridges continuously transmit topology information to notify all other bridges on the network when topology changes are required, such as when a link fails. Bridge Protocol Data Units (BPDUs) are STP information packets that bridges exchange.

The following sections describe spanning tree configuration parameters.

12.1.2.2.1 Root and Designated Bridges

The **root bridge** is the center of the STP topology. A spanning tree instance has one root bridge. Spanning tree bases path calculations on each network components distance from the root bridge.

All other network bridges calculate paths to the root bridge when selecting spanning tree links. STP calculates the distance to the root bridge to build a loop-free topology that features the shortest distance between devices among all possible paths.

Each switch is assigned a unique bridge ID number for each instance. All network switches collectively elect the root bridge by comparing bridge IDs. The root bridge is the switch with the lowest bridge ID.

The bridge ID contains the following eight bytes, in order of decreasing significance:

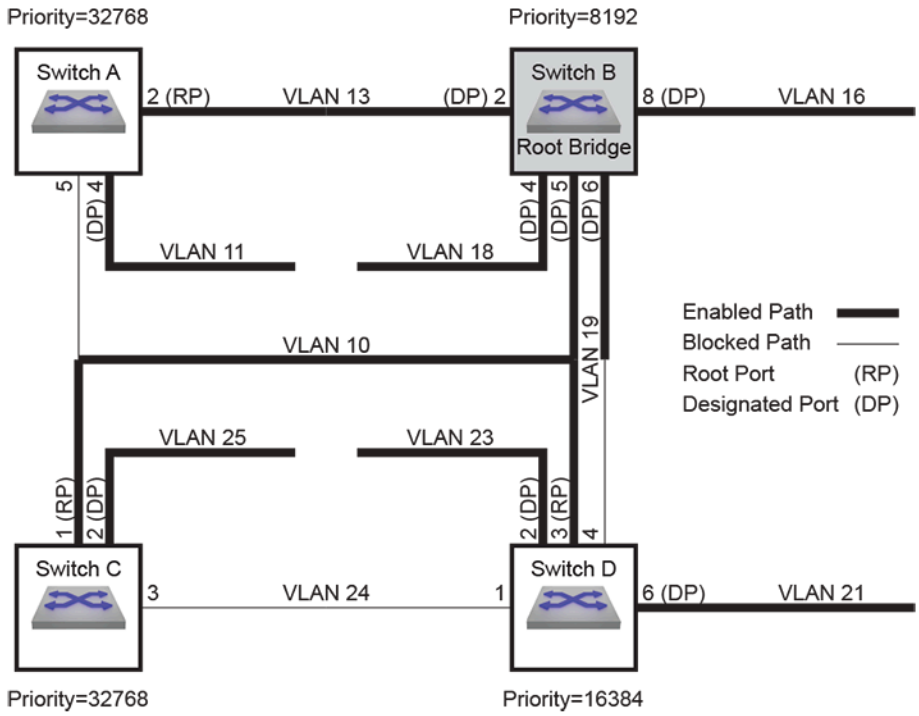
- Port priority (four bits)
- Instance number (12 bits): VLAN number (Rapid-PVST); instance number (MST); 0 (RST)
- MAC address of switch (six bytes)

A **designated bridge** is defined for each network segment as the switch that provides the segments shortest path to the root bridge. A designated bridge is selected for each segment after a root bridge is selected; a switch can be a designated bridge for multiple segments.

The following network calculations in Spanning Tree Network Example assume that each path has the same cost:

- ***Switch B*** is the root bridge its bridge ID is lowest because it has the smallest port priority.
- ***Switch A*** is the designated bridge for ***VLAN 11***.
- ***Switch B*** is the designated bridge for ***VLAN 10***, ***VLAN 13***, ***VLAN 16***, ***VLAN 18***, ***VLAN 19***.
- ***Switch C*** is the designated bridge for ***VLAN 25***.
- ***Switch D*** is the designated bridge for ***VLAN 21***, ***VLAN 23***.

Figure 12-1: Spanning Tree Network Example



12.1.2.2.2 Port Roles

Messages from connected devices to the root bridge traverse a least-cost path, which has the smallest cost among all possible paths to the root bridge. The cost of a path is the sum of the costs of all path segments, as defined through port cost settings.

Active ports in a least cost-path fulfill one of two possible roles: root port and designated port. STP blocks all other network ports. STP also defines alternate and backup ports to handle traffic when an active port is inaccessible.

- Root Port (RP) accesses the bridges least-cost path to the root bridge. Each bridge selects its root port after calculating the cost of each possible path to the root bridge.

  The following ports in Spanning Tree Network Example are root ports:

  - ***Switch A***: ***port 2***
  - ***Switch C***: ***port 1***
  - ***Switch D***: ***port 3***
- Designated Port (DP) accesses a network segments designated bridge. Each segment defines one DP. Switches can provide DPs for multiple segments. All ports on the root bridge are DPs.

The following ports in Spanning Tree Network Example are designated ports:

- ***Switch A***: ***port 4*** (***VLAN 11***)
- ***Switch B***: ***port 2*** (***VLAN 13***), ***port 4*** (***VLAN 18***), ***port 5*** (***VLAN 10***), ***port 6*** (***VLAN 19***), ***port 8*** (***VLAN 16***)
- ***Switch C***: ***port 2*** (***VLAN 25***)
- ***Switch D***: ***port 2*** (***VLAN 23***), ***port 6*** (***VLAN 21***)

- Alternate ports provide backup paths from their bridges to the root bridge. An alternate port is blocked until a network change transforms it into a root port.
- Backup ports provide alternative paths from VLANs to their designated bridges. A backup port is blocked until a network change transforms it into a designated port.

12.1.2.2.3 Port Activity States

A ports activity state defines its current STP activity level. STP monitors BPDUs for network changes that require an activity state transition.

STP defines three port activity states:

- **Forwarding:** The port receives and sends data. Root ports and designated ports are either in, or transitioning to, this state.
- **Discarding:** The port does not receive or send data. Blocked ports receive BPDU packets. All ports except RPs and DPs are blocked, including alternate and backup ports.
- **Learning:** The transitional post-discarding state where the port prepares to forward frames by adding source addresses from inbound data packets to the switching database.

12.1.2.2.4 Port Types

Port type is a configurable parameter that reflects the type of network segment that is connected to the port. Proper port type configuration results in rapid convergence after network topology changes. RSTP port types include normal, network, and edge ports. **Normal** is the default port type.

- **Normal** ports have an unspecified topology.
- **Network** ports connect only to switches or bridges.

  RSTP immediately transitions network ports to the discarding state.
- **Edge** ports connect directly to end stations.

  Edge ports transition directly to forwarding state because they do not create loops. An edge port becomes a normal port when it receives a BPDU.

12.1.2.2.5 Link Types

Link type is a configurable parameter that determines candidates for RSTP fast state transition.

- The default link type for full-duplex ports is **point-to-point**.
- The default link type for half-duplex ports is **shared**.

Fast state transitions are allowed on point-to-point links that connect bridges. Fast state transitions are not allowed on shared ports regardless of the duplex setting.

12.1.2.3 Bridge Protocol Data Units (BPDUs)

Spanning tree rules specify a root bridge, select designated bridges, and assign roles to ports. STP rule implementation requires that network topology information is available to each switch.

Switches exchange topology information through Bridge Protocol Data Units (BPDUs). Information provided by BPDU packets include bridge IDs and root path costs.

12.1.2.3.1 BPDU Types

STP defines three BPDU types:

- Configuration BPDU (CBPDU), used for computing the spanning tree.
- Topology Change Notification (TCN) BPDU, announces network topology changes.
- Topology Change Notification Acknowledgment (TCA), acknowledges topology changes.

Bridges enter the following addresses in outbound BPDU frames:

- Source address: outbound port's MAC address.
- Destination address: STP multicast address ***01:80:C2:00:00:00***.

Bridges regularly exchange BPDUs to track network changes that trigger STP recomputations and port activity state transitions. The **hello timer** specifies the period between consecutive BPDU messages; the default is two seconds.

12.1.2.3.2 Bridge Timers

Bridge timers specify parameter values that the switch includes in BPDU packets that it sends as a root bridge. Bridge timers include:

- **hello-time**: transmission interval between consecutive BPDU packets.
- **forward-time**: the period that ports remain in learning state.
- **max-age**: the period that BPDU data remains valid after it is received.
- **max-hop**: the number of bridges in an MST region that a BPDU can traverse before it is discarded.

The switch recomputes the spanning tree topology if it does not receive another BPDU before the max-age timer expires. When **edge** ports and **point-to-point** links are properly configured, RSTP network convergence does not require forward-delay and max-age timers.

12.1.2.3.3 MSTP BPDUs

MSTP BPDUs are targeted at a single instance and provide STP information for the entire region. MSTP encodes a standard BPDU for the IST, then adds region information and MST instance messages for all configured instances, where each message conveys spanning tree data for an instance. Frames assigned to VLANs operate in the instance to which the VLAN is assigned. Bridges enter an MD5 digest of the VLAN-to-instance map table in BPDUs to avoid including the entire table in each BPDU. Recipients use this digest and other administratively configured values to identify bridges in the same MST region.

MSTP BPDUs are compatible with RSTP. RSTP bridges view an MST region as a single-hop RSTP bridge regardless of the number of bridges inside the region because:

- RSTP bridges interpret MSTP BPDUs as RSTP BPDUs.
- RSTP bridges increment the **message age timer** only once while data flows through an MST region; MSTP measures **time to live** with a **remaining hops** variable, instead of the message age timer.

Ports at the edge of an MST region connecting to a bridge (RSTP or STP) or to an endpoint are *boundary ports*.

12.1.3 Configuring a Spanning Tree

These sections describe the following configuration processes:

- Version Configuration and Instance Creation
- Spanning Tree Instance Configuration
- Port Roles and Rapid Convergence
- Configuring BPDU Transmissions

12.1.3.1 Version Configuration and Instance Creation

The switch supports three STP versions and switchport backup interface pairs. Disabling spanning tree is also supported but not recommended.

The `spanning-tree mode` global configuration command specifies the spanning tree version the switch runs. This section describes command options that enable and configure STP versions.

12.1.3.1.1 Multiple Spanning Tree (MST)

Multiple Spanning Tree is enabled by the `spanning-tree mode` command with the **mstp** option. MSTP is the default STP version.

Example

This command enables Multiple Spanning Tree.

```
switch(config)# spanning-tree mode mstp
switch(config)#
```

Configuring MST Regions

All switches in an MST region must have the same name, revision, and VLAN-to-instance map. MST configuration mode commands sets the region parameters. The ***MST*** configuration mode is a group-change mode where changes are saved by exiting the mode.

Example

The `spanning-tree mst configuration` command places the switch in the ***MST*** configuration mode.

```
switch(config)# spanning-tree mst configuration
switch(config-mst)#
```

The `instance` command assigns VLANs to MST instances. The `name (mst-configuration mode)` and `revision (mst-configuration mode)` commands configure the MST region name and revision.

Examples

- These commands assign ***vlans 4-7*** and ***9*** to instance ***8*** and remove ***vlan 6*** from instance ***10***.

  ```
  switch(config-mst)# instance 8 vlans 4-7,9
  switch(config-mst)#no instance 10 vlans 6
  switch(config-mst)#
  ```

- These commands assign the **name** (***corporate_1***) and **revision** (***3***) to the switch.

  ```
  switch(config-mst)# name corporate_1
  switch(config-mst)# revision 3
  switch(config-mst)#
  ```

The `exit (mst-configuration mode)` command transitions the switch out of the ***MST*** configuration mode and saves all pending changes. The `abort (mst-configuration mode)` command exits the ***MST*** configuration mode without saving the pending changes.

Example

This command exits the ***MST*** configuration mode and saves all pending changes.

```
switch(config-mst)# exit
switch(config)#
```

Configuring MST Instances

These STP commands provide an optional MST instance parameter. These commands apply to instance ***0*** when the optional parameter is not included.

- `spanning-tree priority`
- `spanning-tree root`
- `spanning-tree port-priority`

Examples

- This command configures priority for MST instance ***4***.

```
switch(config)# spanning-tree mst 4 priority 4096
switch(config)#
```

- Each of these commands configure priority for MST instance ***0***.

```
switch(config)# spanning-tree mst 0 priority 4096
```

or

-
```
switch(config)# spanning-tree priority 4096
```

12.1.3.1.2 Rapid Spanning Tree (RST)

Rapid spanning tree is enabled through the `spanning-tree mode` command with the **rstp** option.

Example

- This command enables Rapid Spanning Tree.

```
switch(config)#spanning-tree mode rstp
switch(config)#
```

These STP commands, when they do not include an optional MST or VLAN parameter, apply to RSTP. Commands that configure MSTP instance ***0*** also apply to the RSTP instance.

- `spanning-tree priority`
- `spanning-tree root`
- `spanning-tree port-priority`

Examples

- These commands apply to the RST instance.

```
switch(config)# spanning-tree priority 4096
```

and

```
switch(config)# spanning-tree mst 0 priority 4096
```

- These commands do not apply to the RST instance.

```
switch(config)# spanning-tree mst 4 priority 4096
```

and

```
switch(config)# spanning-tree vlan-id 3 priority 4096
```

Show commands (such as `show spanning-tree`) displays the RSTP instance as MST0 (MST instance ***0***).

Example

- This command, while the switch is in RST mode, displays RST instance information.

```
switch(config)# show spanning-tree
MST0
  Spanning tree enabled protocol rstp<---RSTP mode indicator
  Root ID    Priority    32768
             Address     001c.730c.1867
             This bridge is the root

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.730c.1867
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- -----
Et51             designated forwarding 2000      128.51   P2p

switch(config)#
```

12.1.3.1.3 Rapid Per-VLAN Spanning Tree (Rapid-PVST)

Rapid-PVST mode is enabled by the `spanning-tree mode` command with the **rapid-pvst** option.

Example

- This command enables Rapid Per-VLAN Spanning Tree.

```
switch(config)# spanning-tree mode rapid-pvst
switch(config)#
```

These commands provide an optional VLAN parameter for configuring Rapid-PVST instances.

- `spanning-tree priority`
- `spanning-tree root`
- `spanning-tree port-priority`
- `spanning-tree mst pvst border`

Examples

- This command configures bridge priority for ***vlan 4***.

```
switch(config)# spanning-tree vlan-id 4 priority 4096
switch(config)#
```

- This command enables MSTP PVST+ border ports. The command is effective only if spanning-tree mode is MSTP.

```
switch(config)# spanning-tree mst pvst border
```

12.1.3.1.4 Switchport Backup Mode

Switchport backup interface pairs are enabled through the `spanning-tree mode` command with the **backup** option. Enabling switchport backup disables all spanning-tree modes. For loop avoidance under switchport backup mode, use the Loop Protection feature.

Example

This command enables switchport backup.

```
switch(config)# spanning-tree mode backup
switch(config)#
```

The `switchport backup-link` command establishes an interface pair between the command mode interface (primary) and the interface specified by the command (backup).

Example

These commands establish ***interface ethernet 7*** as the backup port for ***interface ethernet 1***.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#switchport backup-link ethernet 7
switch(config-if-Et1)#
```

The **prefer** option of the `switchport backup-link` command establishes a peer relationship between the primary and backup interfaces and specifies VLAN traffic that the backup interface normally carries. If either interface goes down, the other interface carries traffic normally handled by both interfaces.

Examples

These steps perform the following:

- configures ***interface ethernet 1*** as a trunk port that handles VLANs ***4*** through ***9*** traffic.
- configures ***interface ethernet 2*** as the backup interface.
- assigns ***ethernet 2*** as the preferred interface for VLANs ***7*** through ***9***.

1. Enter configuration mode for the primary interface.

```
switch(config)# interface ethernet 1
```

2. Configure the primary interface as a trunk port that services VLANs ***4-9***

```
switch(config-if-Et1)# switchport mode trunk
switch(config-if-Et1)# switchport trunk allowed vlan 4-9
```

3. Configure the backup interface and specify the VLANs that it normally services.

```
switch(config-if-Et1)# switchport backup-link Ethernet 2 prefer vlan 7-9
switch(config-if-Et1)#
```

12.1.3.1.5 Loop Protection

Loop protection is a loop detection and prevention method which is independent of STP and is not disabled when the switch is in switchport backup mode. When loop protection is active on an interface, that interface periodically sends out loop-detection frames; if one is received that originated on the switch, the receiving port is errdisabled until a timeout period has passed or it is manually reset.

Loop protection is configured and enabled per VLAN, but individual ports in a VLAN can be configured to disable loop protection.

Note: Loop protection cannot be enabled on an MLAG peer link.

This feature is disabled by default. To enable it, use the `monitor loop-protection` command to enter loop-protection configuration mode, then use the `no shutdown (Loop-protection)` command to enable the feature. To enable loop protection on a VLAN, use the `protect vlan` command. To exclude a port from loop protection, use the `no loop-protection` command.

The feature is configured with the following additional commands:

- `transmit-interval`

- disabled-time
- rate-limit

Examples

- This command enters loop protection configuration mode.

```
switch(config)# monitor loop-protection
switch(config-monitor-loop-protect)#
```

- These commands enable loop protection on VLANs ***1025-2000***.

```
switch(config)# monitor loop-protection
switch(config-monitor-loop-protect)# no shutdown
switch(config-monitor-loop-protect)# protect vlan 1025-2000
switch(config-monitor-loop-protect)#
```

- These commands exclude ***interface ethernet 38*** from loop protection.

```
switch(config)# interface ethernet 38
switch(config-if-Et38)# no loop-protection
switch(config-if-Et38)#
```

- These commands configure loop protection with a transmission interval of ***10*** seconds, a disabled time of two days, and a maximum rate of ***500*** loop detection frames per second.

```
switch(config-monitor-loop-protect)# transmit-interval 10
switch(config-monitor-loop-protect)# disabled-time 172800
switch(config-monitor-loop-protect)# rate-limit 500
switch(config-monitor-loop-protect)#
```

12.1.3.1.6 Disabling Spanning Tree

Spanning tree is disabled by the spanning-tree mode command with the **none** option. The switch does not generate STP packets. Switchport interfaces forward packets when connected to other ports. The switch forwards inbound STP packets as multicast data packets on the VLAN where they are received.

Example

This command disables all STP functions.

```
switch(config)# spanning-tree mode none
switch(config)#
```

12.1.3.2 Spanning Tree Instance Configuration

A network performs these steps to set up an STP instance:

1. The bridge with the lowest ID is elected root bridge.
2. Root Ports (RP) are selected on all other bridges.
3. Designated bridges are selected for each network segment.
4. Designated Ports (DP) are selected on each designated bridge.
5. Networks begin forwarding data through RPs and DPs. All other ports are blocked.

12.1.3.2.1 Root Bridge Parameters

STPs use bridge IDs for electing the root bridge. Switches denote a bridge ID for each configured Spanning Tree instance. The bridge ID composition is:

- Priority (four bits)

Priority is expressed as a multiple of 4096 because it is stored as the four most significant bits of a two-byte number.

- Protocol Dependent (twelve bits)
  - **Rapid-PVST:** VLAN number
  - **MST:** Instance number
  - **RST:** 0
- MAC address of switch (six bytes)

Example

The switch defines bridge IDs for three MST instances:

- **MST 0:** ***32768*** (Priority (***32768***)+Instance number(***0***)) and ***001c.7301.23de*** (MAC address)
- **MST101:** ***32869*** (Priority (***32768***)+Instance number(***101***)) and ***001c.7301.23de*** (MAC address)
- **MST102:** ***32870*** (Priority (***32768***)+Instance number(***102***)) and ***001c.7301.23de*** (MAC address)

This command displays a table of root bridge information.

```
switch> show spanning-tree root
                  Root ID          Root    Hello  Max  Fwd
Instance    Priority    MAC addr Cost    Time   Age  Dly  Root Port
----------  ------------------- ------ -----  ---  ---  ----------
MST0         32768 001c.7301.23de     0    2    20  15   Po937
MST101       32869 001c.7301.23de  3998    0     0   0   Po909
MST102       32870 001c.7301.23de  3998    0     0   0   Po911
```

The switch provides two commands that configure the switch priority: `spanning-tree priority` and `spanning-tree root`. The commands differ in the available parameter options:

- **spanning-tree priority** options are integer multiples of ***4096*** between ***0*** and ***61440***.
- **spanning-tree root** options are **primary** and **secondary**.
- **primary** assigns a priority of ***8192***.
- **secondary** assigns a priority of ***16384***.

The default priority value is ***32768***.

The following examples configure bridge IDs with both commands.

Examples

- These commands configure MST instance bridge priorities with the **root** command:

```
switch(config)# spanning-tree mst 0 root primary
switch(config)# spanning-tree mst 1 root secondary
switch> show spanning-tree root
                   Root ID          Root    Hello  Max  Fwd
Instance    Priority   MAC addr  Cost    Time   Age  Dly  Root Port
----------  ------------------- ------- -----  ---  ---  ----------
MST0          8192 001c.7301.6017      0    2    20  15   None
MST1         16385 001c.7301.6017      0    0     0   0   None
MST2         32770 001c.7301.6017      0    0     0   0   None
```

  - Instance ***0*** root priority is ***8192***: primary priority plus the instance number of ***0***.
  - Instance ***1*** root priority is ***16385***: secondary priority plus the instance number of ***1***.
  - Instance ***2*** root priority is ***32770***: default priority plus the instance number of ***2***.

    These priority settings normally program the switch to be the primary root bridge for instance 0, the secondary root bridge for instance 1, and a normal bridge for instance 2. Primary and secondary root bridge elections also depend on the configuration of other network bridges.

- These priority commands configure Rapid-PVST VLAN bridge priorities:

```
switch(config)# spanning-tree vlan-id 1 priority 8192
switch(config)# spanning-tree vlan-id 2 priority 16384
switch(config)# spanning-tree vlan-id 3 priority 8192
switch(config)# no spanning-tree vlan-id 4 priority
switch(config)# show spanning-tree root
                  Root ID          Root    Hello  Max  Fwd
Instance    Priority    MAC addr  Cost   Time   Age  Dly  Root Port
----------  ------------------- ------- -----  ---  ---  ----------
VL18193       001c.7301.6017       0        2     20   15   None
VL216386      001c.7301.6017       0        2     20   15   None
VL38195       001c.7301.6017       0        2     20   15   None
VL432788      001c.7301.6017       0        2     20   15   None
```

  - VLAN ***1*** root priority is ***8193***: configured priority plus the VLAN number of ***1***.
  - VLAN ***2*** root priority is ***16386***: configured priority plus the VLAN number of ***2***.
  - VLAN ***3*** root priority is ***8195***: configured priority plus the VLAN number of ***3***.

    These priority settings normally program the switch to be the primary root bridge for VLANs ***1*** and ***3***, the secondary root bridge for ***vlan 2***, and a normal bridge for ***vlan 4***. Primary and secondary root bridge elections also depend on the configuration of other network bridges.***vlan 4*** root priority is ***32788***: default priority plus the VLAN number of ***4***.

12.1.3.2.2 Path Cost

Spanning tree calculates the costs of all possible paths from each component to the root bridge. The path cost is equal to the sum of the cost assigned to each port in the path. Ports are assigned a cost by default or through CLI commands. Cost values range from *1* to ***200000000*** (200 million).

The default cost is a function of the interface speed:

- 1 gigabit interfaces have a default cost of ***20000***.
- 10 gigabit interfaces have a default cost of ***2000***.

The `spanning-tree cost` command configures the path cost of the configuration mode interface. Costs can be specified for Ethernet and port channel interfaces. The command provides a mode parameter for assigning multiple costs to a port for MST instances or Rapid-PVST VLANs.

Examples

- These commands configure a port cost of ***25000*** to ***interface ethernet 5***. This cost is valid for RSTP or MSTP instance ***0***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree cost 25000
switch(config-if-Et5)#
```

- This command configures a path cost of ***300000*** to ***interface ethernet 5*** in MST instance ***200***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree mst 200 cost 300000
switch(config-if-Et5)#
```

- This command configures a path cost of ***10000*** to ***interface ethernet 5*** in Rapid-PVST VLAN ***200-220***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree vlan-id 200-220 cost 10000
switch(config-if-Et5)#
```

12.1.3.2.3 Port Priority

STP uses the port priority interface parameter to select ports when resolving loops. The port with the lower port priority numerical value is placed in forwarding mode. When multiple ports are assigned equal port priority numbers, the port with the lower interface number is placed in forwarding mode. Valid port-priority numbers are multiples of 16 between ***0*** and ***240***; the default is ***128***.

The `spanning-tree port-priority` command configures the port-priority number for the configuration mode interface. The command provides a mode option for assigning different priority numbers to a port for multiple MST instances or Rapid-PVST VLANs. Port-priority can be specified for Ethernet and port channel interfaces.

Examples

- This command sets the access port priority of ***144*** for Ethernet ***5*** interface.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree port-priority 144
switch(config-if-Et5)#
```

- This command sets the access port priority of ***144*** for Ethernet ***5*** interface in MST instance ***10***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree mst 10 port-priority 144
switch(config-if-Et5)#
```

12.1.3.3 Port Roles and Rapid Convergence

Spanning Tree provides the following options for controlling port configuration and operation:

- **portfast**: Allows ports to skip learning state before entering the forwarding state.
- **port type** and **link type**: Designates ports for rapid transitions to the forwarding state.
- **root guard**: Ensures that a port will not become the root port.
- **loop guard**: Prevents loops resulting from unidirectional failure of links.
- **bridge assurance**: Prevents loops caused by unidirectional links or a malfunctioning switch.

12.1.3.3.1 PortFast

PortFast allows devices to gain immediate network access before convergence of the spanning tree. Enabling PortFast on ports connected to another switch can create loops.

A **portfast** port that receives a BPDU sets its operating state to **non-portfast** while remaining in **portfast** configured state. In this state, the port is subject to topology changes and can enter the discarding state.

The spanning-tree portfast command programs access ports to immediately enter the forwarding state. PortFast connects devices attached to an access port, such as a single workstation, to the network immediately without waiting for STP convergence. PortFast can also be enabled on trunk ports.

Example

This command unconditionally enables portfast on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree portfast
switch(config-if-Et5)#
```

12.1.3.3.2 Port Type and Link Type Configuration

RSTP only achieves rapid transition to forwarding state on edge ports and point-to-point links.

Port Type

Edge ports are directly connected to end stations. Because edge ports do not create loops, they transition directly to forwarding state when a link is established.

The spanning-tree portfast <port type> command sets the configuration mode interfaces port type. Spanning tree ports can be configured as **edge** ports, **network** ports, or **normal** ports. The default port type is **normal**.

- **edge ports** connect to a host (end station). Configuring a port that connects to a bridge as an edge port may create a loop. Edge ports that receive a BPDU become a normal spanning tree port.
- **network ports** connect only to a Layer 2 switch or bridge. Configuring a port connected to a host as a network port transitions the port to the discarding state.
- **normal ports** have an unspecified topology.

Examples

- This command configures ***interface ethernet 5*** as a network port.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree portfast network
switch(config-if-Et5)#
```

- Auto-edge detection converts ports into edge ports when they do not receive a new BPDU before the current BPDU expires, as measured by the max-age timer. The spanning-tree portfast auto command enables auto-edge detection on the configuration mode interface, superseding the **spanning-tree portfast** command. Auto-edge detection is enabled by default.

  This command enables auto-edge detection on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree portfast auto
switch(config-if-Et5)#
```

Link Type

The switch derives a ports default link type from its duplex mode:

- full-duplex ports are **point-to-point**.
- half-duplex ports are **shared link**.

The spanning-tree link-type command specifies the configuration mode interfaces link-type. RSTP fast transition is not allowed on **shared link** ports, regardless of their duplex setting. Because the ports are full-duplex by default, the default link-type setting is **point-to-point**.

Example

This command configures ***interface ethernet 5*** as a shared port.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree link-type shared
switch(config-if-Et5)#
```

12.1.3.3.3 Root Guard and Loop Guard

Root guard stops a port from becoming a root port, which stops connected switches from becoming root bridges. When a switch detects a new root bridge, its root-guard-enabled ports enter blocked (root-inconsistent) state. When the switch no longer detects a new root, these ports enter learning state.

Root guard is enabled on a per-port basis. The setting applies to all STP instances. Disabling root guard places the port in learning state.

The spanning-tree guard command, with the root option, enables root guard on the configuration mode interface.

Example

This command enables root guard on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree guard root
switch(config-if-Et5)#
```

Loop guard prevents loops resulting from unidirectional failure of point-to-point links by verifying that non-designated ports (root, blocked, and alternate) are receiving BPDUs from their designated ports. A loop-guard-enabled root or blocked port that stops receiving BPDUs transitions to the discarding (loop-inconsistent) state. The port recovers from this state when it receives a BPDU.

Loop guard, when enabled globally, applies to all point-to-point ports. Loop guard is configurable on individual ports and applies to all STP instances of an enabled port. Loop-inconsistent ports transition to learning state when loop guard is disabled.

If loop guard is enabled on a root switch, it takes effect only if the switch becomes a nonroot switch.

When using loop guard:

- Do not enable loop guard on portfast-enabled ports.
- Loop guard is not functional on ports not connected to point-to-point links.
- Loop guard has no effect on disabled spanning tree instances.

Loop guard aspects on port channels include:

- BPDUs are sent over the channels first operational port. Loop guard blocks the channel if that link becomes unidirectional even when other channel links function properly.
- Creating a new channel destroys state information for its component ports; new channels with loop-guard-enabled ports can enter forwarding state as a DP.
- Dissembling a channel destroys its state information; component ports from a blocked channel can enter the forwarding state as DPs, even if the channel contained unidirectional links.
- If a link on any port of the channel becomes unidirectional, the channel is blocked. Transmission resumes if the port is removed from the channel or the bidirectional communication is restored.

Loop guard configuration commands include:

- `spanning-tree guard loop default` command enables loop guard as a default on all switch ports.
- spanning-tree guard control the loop guard setting on the configuration mode interface. This command overrides the default command for the specified interface.

Examples

- This command enables loop guard as the default on all switch ports.

```
switch(config)# spanning-tree guard loop default
switch(config)#
```

- This command enables loop guard on ***interface ethernet 6***.

```
switch(config)# interface ethernet 6
switch(config-if-Et6)# spanning-tree guard loop
switch(config-if-Et6)#
```

12.1.3.3.4 Bridge Assurance

Bridge assurance protects against unidirectional link failures, other software failures, and devices that continue forwarding data traffic after they quit running spanning tree.

Bridge assurance programs the switch to send BPDUs at each hello time period through all bridge assurance-enabled ports (i.e., network ports). Bridge assurance operates only on **network** ports with **point-to-point** links, ideally with bridge assurance enabled on each side of the link. Bridge assurance-enabled ports will not necessarily be blocked when they link to a port where bridge assurance is not enabled.

Ports not receiving a BPDU packet within a hello time period enter inconsistent (blocking) state. In this case, the show spanning-tree transmit active command will show a bridge assurance status of inconsistent for the port. If the other side of the link has bridge assurance enabled, or if the other switch is the root bridge, it will send periodic BPDUs, preventing an inconsistent blocking state.

Bridge assurance is globally enabled by default, but must also be enabled on a per-port basis by designating the port as a network port with the `spanning-tree portfast <port type>` command. The `no spanning-tree transmit active` command disables bridge assurance globally.

Example

These commands enable bridge assurance on the switch, then enable bridge assurance on ***interface ethernet 5*** by designating it a network port.

```
switch(config)# spanning-tree transmit active
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree portfast network
switch(config-if-Et5)#
```

12.1.3.4 Configuring BPDU Transmissions

The following sections describe instructions that configure BPDU packet contents and transmissions.

12.1.3.4.1 Configuring Bridge Timers

Bridge timers specify parameter values that the switch includes in BPDU packets that it sends as a root bridge. Bridge timers include:

- **hello-time**: transmission interval between consecutive BPDU packets.
- **forward-time**: the period that ports remain in learning state.
- **max-age**: the period that BPDU data remains valid after it is received.
- **max-hop**: the number of bridges in an MST region that a BPDU can traverse before it is discarded.

In standard STP, ports passively wait for **forward_delay** and **max_age** periods before entering the forwarding state. RSTP achieves faster convergence by relying on edge port and link type definitions to start forwarding traffic. When edge ports and link types are properly configured, bridge timers are used in RSTP as backup or when interacting with networks running standard STP.

The `spanning-tree hello-time` command configures the hello time.

Examples

- This command configures a hello-time of ***1*** second (***1000*** ms).

  ```
  switch(config)# spanning-tree hello-time 1000
  switch(config)#
  ```

  The `spanning-tree max-hops` command specifies the max hop setting that the switch inserts into BPDUs that it sends out as the root bridge.
- This command sets the max hop value to ***40***.

  ```
  switch(config)# spanning-tree max-hops 40
  switch(config)#
  ```

  The `spanning-tree forward-time` command configures the forward delay setting that the switch inserts into BPDUs that it sends out as the root bridge.

- This command sets the forward delay timer value to ***25*** seconds.

```
switch(config)# spanning-tree forward-time 25
switch(config)#
```

The `spanning-tree max-age` command configures the max age setting that the switch inserts into BPDUs that it sends out as the root bridge.

- This command sets the max age timer value to ***25*** seconds.

```
switch(config)# spanning-tree max-age 25
switch(config)#
```

12.1.3.4.2 BPDU Transmit Hold-Count

The `spanning-tree bpdu tx hold-count` command specifies the maximum number of BPDUs per second that the switch can send from an interface. Valid settings range from ***1*** to ***10*** BPDUs with a default of ***6*** BPDUs.

Higher hold-count settings can significantly impact CPU utilization, especially in Rapid-PVST mode. Smaller values can slow convergence in some configurations.

Example

This command configures a transmit hold-count of ***8*** BPDUs.

```
switch(config)# spanning-tree bpdu tx hold-count 8
switch(config)#
```

12.1.3.4.3 BPDU Guard

PortFast interfaces do not receive BPDUs in a valid configuration. BPDU Guard provides a secure response to invalid configurations by disabling ports when they receive a BPDU. Disabled ports differ from blocked ports in that they are re-enabled only through manual intervention.

- When configured globally, BPDU Guard is enabled on ports in the operational portfast state.
- When configured on an individual interface, BPDU Guard disables the port when it receives a BPDU, regardless of the ports portfast state.

The `spanning-tree edge-port bpduguard default` global configuration command enables BPDU guard by default on all portfast ports. BPDU guard is disabled on all ports by default.

The `spanning-tree bpduguard` interface configuration command controls BPDU guard on the configuration mode interface. This command takes precedence over the default setting configured by `spanning-tree edge-port bpduguard default`.

- **`spanning-tree bpduguard`** enables BPDU guard on the configuration mode interface.
- **`spanning-tree bpduguard disable`** disables BPDU guard on the configuration mode interface.
- **`no spanning-tree bpduguard`** reverts the configuration mode interface to the default BPDU guard setting.

Example

These commands enable BPDU guard by default on all portfast ports, then disable BPDU guard on ***interface ethernet 5***.

```
switch(config)# spanning-tree edge-port bpduguard default
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree bpduguard disable
switch(config-if-Et5)
```

12.1.3.4.4 BPDU Filter

BPDU filtering prevents the switch from sending or receiving BPDUs on specified ports. BPDU filtering is configurable on Ethernet and port channel interfaces.

Ports with BPDU filtering enabled do not send BPDUs and drops inbound BPDUs. Enabling BPDU filtering on a port not connected to a host can result in loops as the port continues forwarding data while ignoring inbound BPDU packets.

The `spanning-tree bpdufilter` command controls BPDU filtering on the configuration mode interface. BPDU filtering is disabled by default.

Example

These commands enable BPDU filtering on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree bpdufilter enable
switch(config-if-Et5)#
```

12.1.3.4.5 BPDU Rate Limit

BPDU input rate limiting restricts the number of BPDUs that a port with BPDU guard and BPDU filter disabled, can process during a specified interval. The port discards all BPDUs that it receives in excess of the specified limit. Configuring the rate limiter requires two steps:

- Establishing the rate limit threshold
- Enabling rate limiting

Establishing the Rate Limit Threshold

The `spanning-tree bpduguard rate-limit count (interface)` command specifies the BPDU reception rate (quantity per interval) that triggers the discarding of BPDUs. The command is available in global and interface configuration modes.

- The **`spanning-tree bpduguard rate-limit count`** global command specifies the maximum reception rate for ports that are not covered by interface rate limit count commands. The global command specifies the default limit.

  Note: Arista Networks recommends to retain the default values of rate limit. In the PVST mode, when the VLAN membership of a port is changed by a significant margin, it is advisable to disable interface BPDU rate limit on both ends of a port. For example, if three VLANs are present on a port initially, the operator must first add 300 more VLANs on one side of the port and then add the same 300 VLANs on the other side of the port. In this case, if the VLANs are increased towards the root bridge first, then the other side can cross the rate-limit threshold.
- The **`spanning-tree bpduguard rate-limit count`** interface command defines the maximum BPDU reception rate for ports in the configuration mode interface.

Examples

- This command configures the global limit of ***5000*** BPDUs over a ***4*** second interval.

  ```
  switch(config)# spanning-tree bpduguard rate-limit count 5000 interval 4
  switch(config)#
  ```
- These commands configure a limit of ***7500*** BPDUs over an ***8*** second interval on the ***interface ethernet 2***.

  ```
  switch(config)# interface ethernet 2
  switch(config-if-Et2)# spanning-tree bpduguard rate-limit count 7500
   interval 8
  switch(config-if-Et2)#
  ```

Enabling Rate Limiting

BPDU rate limiting is enabled globally or on individual ports:

- `spanning-tree bpduguard rate-limit default` enables ratelimiting on all ports that are not covered by the interface rate limiting command. The default setting is **enabled**.
- `spanning-tree bpduguard rate-limit enable / disable` interface command enables or disables BPDU rate limiting on the configuration mode interface. This command has precedence over the global command.

Examples

- This command enables rate limiting on ports that are not covered by interface rate limit commands.

```
switch(config)# spanning-tree bpduguard rate-limit default
switch(config)#
```

- These commands enable rate limiting on the ***interface ethernet 15***.

```
switch(config)# interface ethernet 15
switch(config-if-Et15)# spanning-tree bpduguard rate-limit enable
switch(config-if-Et15)#
```

12.1.4 STP Commands

Spanning Tree Commands: Global Configuration

- spanning-tree bpdu tx hold-count
- spanning-tree bpduguard rate-limit count (global)
- spanning-tree bpduguard rate-limit default
- spanning-tree edge-port bpdufilter default
- spanning-tree edge-port bpduguard default
- spanning-tree forward-time
- spanning-tree guard loop default
- spanning-tree hello-time
- spanning-tree max-age
- spanning-tree max-hops
- spanning-tree mode
- spanning-tree mst configuration
- spanning-tree mst pvst border
- spanning-tree portchannel guard misconfig
- spanning-tree priority
- spanning-tree root
- spanning-tree transmit active
- spanning-tree vlan-id

Loop Protection Commands

- disabled-time
- loop-protection
- monitor loop-protection
- protect vlan
- rate-limit
- show loop-protection
- shutdown (Loop-protection)

- transmit-interval

Spanning Tree Commands: Interface Configuration Mode

- spanning-tree bpdufilter
- spanning-tree bpduguard
- spanning-tree bpduguard rate-limit count (interface)
- spanning-tree bpduguard rate-limit enable / disable
- spanning-tree cost
- spanning-tree guard
- spanning-tree link-type
- spanning-tree port-priority
- spanning-tree portfast
- spanning-tree portfast auto
- spanning-tree portfast <port type>
- switchport backup-link

MST Configuration Commands

- abort (mst-configuration mode)
- exit (mst-configuration mode)
- instance
- name (mst-configuration mode)
- revision (mst-configuration mode)
- show (mst-configuration mode)

Display Commands

- show spanning-tree
- show spanning-tree blockedports
- show spanning-tree counters
- show spanning-tree instance
- show spanning-tree instance detail
- show spanning-tree interface
- show spanning-tree mst
- show spanning-tree mst configuration
- show spanning-tree mst interface
- show spanning-tree mst test information
- show spanning-tree root
- show spanning-tree topology status
- show spanning-tree transmit active

Clear Commands

- clear spanning-tree counters
- clear spanning-tree counters session
- clear spanning-tree detected-protocols

12.1.4.1 abort (mst-configuration mode)

The **abort** command, in MST-configuration mode, discards pending changes to the MST region configuration, then returns the switch to global configuration mode.

The **exit (mst-configuration mode)** command saves MST region changes to ***running-config*** before returning the switch to global configuration mode.

Command Mode

MST-configuration

Command Syntax

abort

Example

This command discards changes to the MST region, then returns the switch to global configuration mode.

```
switch(config-mst)# abort
switch(config)#
```

12.1.4.2 clear spanning-tree counters

The **clear spanning-tree counters** command resets the BPDU counters for the specified interfaces to zero in all CLI sessions.

Command Mode

Privileged EXEC

Command Syntax

clear spanning-tree counters [**INT_NAME**]

Parameters

INT_NAME Interface type and number. Options include:

- ***no parameter*** resets counters for all interfaces.
- **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **interface loopback *l_num*** Loopback interface specified by ***l_num***.
- **interface management *m_num*** Management interface specified by ***m_num***.
- **interface port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **interface vlan *v_num*** VLAN interface specified by ***v_num***.

Example

This command resets the BPDU counters on ***interface ethernet 15***.

```
switch# show spanning-tree counters
            Port    Sent      Received    Tagged Error   Other Error
-------------------------------------------------------------------
      Ethernet15    32721          0           0              0
  Port-Channel10     8487          0           0              0

switch# clear spanning-tree counters interface ethernet 15<---Clear command
switch# show spanning-tree counters
            Port    Sent       Received   Tagged Error    Other Error
-------------------------------------------------------------------
      Ethernet15      11            0           0              0
  Port-Channel10    84942           6           0 switch#
```

12.1.4.3 clear spanning-tree counters session

The **clear spanning-tree counter session** command resets the BPDU counters to zero on all interfaces in the current CLI session. Counters in other CLI sessions are not affected.

Command Mode

Privileged EXEC

Command Syntax

clear spanning-tree counters session

Example

This command resets the BPDU counters in the current CLI session.

```
switch# show spanning-tree counters
            Port       Sent      Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15      32721             0                  0              0
  Port-Channel10       8487             0                  0              0

switch# clear spanning-tree counters session
switch# show spanning-tree counters
            Port       Sent      Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15         11             0                  0              0
  Port-Channel10          7             2                  6              0 switch#
```

12.1.4.4 clear spanning-tree detected-protocols

The **clear spanning-tree detected-protocols** command restarts the Spanning Tree Protocol (STP) migration state machine on the specified interfaces. The switch is reset to running rapid spanning tree protocol on an interface where it previously detected a bridge running an old version of the protocol.

Command Mode

Privileged EXEC

Command Syntax

clear spanning-tree detected-protocols [**INT_NAME**]

Parameters

INT_NAME Interface type and number. Values include:

- ***no parameter*** all interfaces.
- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.

Example

This command restarts the STP migration machine on all switch interfaces.

```
switch# clear spanning-tree detected-protocols
switch#
```

12.1.4.5 disabled-time

The **disabled-time** command sets the time for which the port remains disabled after a loop is detected by loop protection. The **no disabled-time** and **default disabled-time** commands reset the disabled time to the default of ***604800*** seconds (seven days).

Note: If this value is changed, interfaces that are already disabled by loop protection will remain disabled for the previously configured period.

Command Mode

Loop-protection Configuration

Command Syntax

`disabled-time` [***period***]

`no disabled-time` [***period***]

`default disabled-time` [***period***]

Parameters

period Time in seconds for which the port remains disabled. Values range from ***0*** to ***604800*** (seven days). Default is ***604800***. A value of ***0*** disables the interface until it is manually reset, even if the disabled time is later set to a non-zero value. To restore the port manually, shut it down and then re-enable it.

Example

This command configures loop protection to disable a port on which a loop is detected for a period of two days (***172800*** seconds).

```
switch(config-monitor-loop-protect)# disabled-time 172800
switch(config-monitor-loop-protect)#
```

12.1.4.6 exit (mst-configuration mode)

The **`exit`** command, in MST-configuration mode, saves changes to the MST region configuration, then returns the switch to global configuration mode. MST region configuration changes are also saved by entering a different configuration mode.

Command Mode

MST-configuration

Command Syntax

`exit`

Examples

- This command saves changes to the MST region, then returns the switch to global configuration mode.

```
switch(config-mst)# exit
switch(config)#
```

- This command saves changes to the MST region, then places the switch in Interface-Ethernet mode.

```
switch(config-mst)# interface ethernet 3
switch(config-if-Et3)#
```

12.1.4.7 instance

The **`instance`** command inserts an entry into the VLAN-to-instance map that associates a set of VLANs to an MST instance. In addition to defining the MST topology, the VLAN-to-instance map is one of three parameters, along with the MST name and revision number, that identifies the switchs MST region.

The **`no instance`** command removes specified entries from the VLAN-to-instance map. If the command does not provide a VLAN list, all entries are removed for the specified instance. The **`no instance`** and **`default instance`** commands function identically.

Command Mode

MST-Configuration

Command Syntax

`instance` ***mst_inst*** vlans ***v_range***

no instance ***mst_inst*** [vlans ***v_range***]

no default instance ***mst_inst*** [vlans ***v_range***]

Parameters

- ***mst_inst*** MST instance number. Value of ***mst_inst*** ranges from ***0*** to ***4094***.
- ***v_range*** VLAN list. Formats include a number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command maps VLANs ***20-39*** to MST instance ***2***.

```
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 2 vlans 20-39
switch(config-mst)#
```

- This command removes all VLAN mappings to MST instance ***10***.

```
switch(config-mst)# no instance 10
switch(config-mst)#
```

12.1.4.8 loop-protection

The **loop-protection** command enables loop protection on the configuration mode interface. All interfaces in a VLAN under loop protection have loop protection enabled by default. The **no loop-protection** and **default loop-protection** commands disable loop protection on the interface.

When loop protection is disabled (at the VLAN or interface level), the computed state of the interface is forgotten and packets queued to be sent are dropped. If an interface is err-disabled by loop protection, disabling loop protection removes the err-disable.

Command Mode

Interface-Ethernet Configuration

Command Syntax

loop-protection

no loop-protection

default loop-protection

Example

These commands disable loop protection on ***interface ethernet 2/4***. If the interface is currently errdisabled by loop protection, the errdisable is removed.

```
switch(config)# interface ethernet 2/4
switch(config-if-Et2/4)# no loop-protection
switch(config-if-Et2/4)#
```

12.1.4.9 monitor loop-protection

The **monitor loop-protection** command places the switch in loop-protection configuration mode.

Command Mode

Global Configuration

Command Syntax

monitor loop-protection

Commands available in loop-protection configuration mode:

- shutdown (Loop-protection)
- protect vlan
- transmit-interval
- disabled-time
- rate-limit

Example

This command places the switch in ***loop-protection*** configuration mode.

```
switch(config)# monitor loop-protection
switch(config-monitor-loop-protect)#
```

12.1.4.10 name (mst-configuration mode)

The **name** command configures the MST region name. The name is one of three parameters, along with the MST revision number and VLAN-to-instance map, that identifies the switchs MST region.

The name has up to 32 characters. The default name is an empty string. The name string accepts all characters except the space.

The **no name** and **default name** commands restore the default name by removing the **name** command from ***running-config***.

Command Mode

MST-Configuration

Command Syntax

name ***label_text***

no name

default name

Parameters

label_text character string assigned to name attribute. Maximum 32 characters. The space character is not permitted in the name string.

Example

This command assigns ***corporate_100*** as the MST region name.

```
switch(config)# spanning-tree mst configuration
switch(config-mst)# name corporate_100
switch(config-mst)# show pending
Active MST configuration
Name      [corporate_100]
Revision  0    Instances configured 1

Instance  Vlans mapped
-------------------------------------------------------
0         1-4094
-------------------------------------------------------
```

12.1.4.11 protect vlan

The **protect vlan** command specifies which VLANs participate in loop protection. The **no protect vlan** and **default protect vlan** commands remove loop protection from the specified VLANs.

Command Mode

Loop-protection Configuration

Command Syntax

`protect vlan` ***vlan-range***

Parameters

vlan-range List of VLANs (number, range, comma-delimited list of numbers and ranges). Numbers range from ***1*** to ***4094***.

Example

This command enables loop protection on VLANs ***1025-2000***.

```
switch(config-monitor-loop-protect)# protect vlan 1025-2000
switch(config-monitor-loop-protect)#
```

12.1.4.12 rate-limit

The **`rate-limit`** command sets the maximum number of loop detection frames which can be sent by the switch per second. The **`no rate-limit`** and **`default rate-limit`** commands return the rate limit to the default value of ***1000***.

Command Mode

Loop-protection Configuration

Command Syntax

`rate-limit` [***frames***]

`no rate-limit` [***frames***]

`default rate-limit` [***frames***]

Parameters

frames Maximum number of frames sent per second. Values range from ***0-1000***, default is ***1000***. A value of ***0*** disables throttling.

Example

This command sets the maximum number of loop detection frames to ***500*** per second.

```
switch(config-monitor-loop-protect)# rate-limit 500
switch(config-monitor-loop-protect)#
```

12.1.4.13 revision (mst-configuration mode)

The **`revision`** command configures the MST revision number. The revision number is one of three parameters, along with the MST name and VLAN-to-instance map, that identifies the switchs MST region. Revision numbers range from ***0*** to ***65535***. The default revision number is *0*.

The **`no revision`** and **`default revision`** commands restore the revision number to its default value by removing the revision command from ***running-config***.

Command Mode

MST-Configuration

Command Syntax

`revision` ***rev_number***

no revision

default revision

Parameters

rev_number revision number. Possible ranges from ***0*** to ***65535*** with a default of ***0***.

Example

This command sets the revision number to ***15***.

```
switch(config)# spanning-tree mst configuration
switch(config-mst)# revision 15
switch(config-mst)# show pending
Active MST configuration
Name      []
Revision 15Instances configured 1

Instance  Vlans mapped
--------  ----------------------------------------------------
0         1-4094
---------------------------------------------------------------
```

12.1.4.14 show (mst-configuration mode)

The **show** command displays the current and pending MST configuration:

Exiting MST configuration mode stores all pending configuration changes to ***running-config***.

Command Mode

MST-Configuration

Command Syntax

show [EDIT_VERSION]

Parameters

EDIT_VERSION specifies configuration version that the command displays. Options include:

- ***no parameter*** command displays pending MST configuration.
- **active** command displays MST configuration stored in ***running-config***.
- **current** command displays MST configuration stored in ***running-config***.
- **pending** command displays pending MST configuration.

Example

These commands contrast the difference between the active and pending configuration by adding MST configuration commands, then showing the configurations.

```
switch(config-mst)# show pending
Active MST configuration
Name      []
Revision  0    Instances configured 1

Instance  Vlans mapped
--------  ----------------------------------------------------
0         1-4094
---------------------------------------------------------------
switch(config-mst)# instance 2 vlan 20-29,102
switch(config-mst)# revision 2
switch(config-mst)# name baseline
switch(config-mst)# show pending
```

```
Pending MST configuration
Name      [baseline]
Revision  2    Instances configured 2

Instance  Vlans mapped
-------------------------------------------------------
0         1-19,30-101,103-4094
2         20-29,102
-------------------------------------------------------
switch(config-mst)# show active
Active MST configuration
Name      []
Revision  0    Instances configured 1

Instance  Vlans mapped
-------------------------------------------------------
0         1-4094
-------------------------------------------------------
```

12.1.4.15 show loop-protection

The **show loop-protection** command displays loop protection status.

Command Syntax

show loop-protection [**detail**]

Examples

- This command displays basic loop protection information.

```
switch# show loop-protection
Loop protection is enabled
Transmit interval: 5
Disable Time: 604800(or Permanent)
Packets Transmitted rate: 12/second(or Unthrottled)
Total: 3 Vlans enabled.
switch>
```

- This command displays detailed information about loop protection.

```
switch# show loop-protection detail
Loop protection is enabled

Transmit interval: 5
Disable Time: 604800
Packets Transmitted rate: 12/second
Total: 3 Vlans enabled.
Destination address: ffff.ffff.ffff
Ethernet type: 0x88b7
Receive action: Interface Disable

Vlan        Loop      Disabled Intfs Total Latest
Detected              Intfs Disabled Time
----------- --------- -------------- ----- -------------
1           Yes       Et1-2          20     18:01
2           No        -              20     -
3           No        -              20     -
switch#
```

- This command displays loop protection information for the interfaces in VLANS 3-4.

```
switch# show loop-protection vlan 3-4
```

```
Vlan Intf LP Enabled State     LP       Disabled Bring
                                 Disabled at       up at
----- ----- ---------- --------- -------  -------- ------
3     Et1   Yes        shutdown   Yes      17:21    18:21
3     Et2   Yes        shutdown   No       -        -
3     Et3   Yes        enabled    No       -        -
3     Et4   No         -          -        -        -
4     -     No         -          -        -        -
switch#
```

- This command displays the number of loop detection packets sent and received.

```
switch# show loop-protection counters
VLAN        Tx     Rx  Rx-Other
-------  -----   ----  --------
2           200     0     100
3           200     1       0

Intfs       Tx     Rx  Rx-Other
-------  -----   ----  --------
Et1        200     0       100
Et2        200     1         0 switch#
```

12.1.4.16 show spanning-tree

The **show spanning-tree command** displays spanning tree protocol (STP) data, organized by instance.

Command Mode

EXEC

Command Syntax

show spanning-tree [**VLAN_ID**][**INFO_LEVEL**]

Parameters

- **VLAN_ID** specifies the VLANs for which the command displays information. Formats include:
  - ***no parameter*** displays information for all VLANs.
  - **vlan** displays data for instances containing the first VLAN listed in running-config.
  - **vlan *v_range*** displays data for instances containing a VLAN in the specified range.
- **INFO_LEVEL** specifies level of information detail provided by the command.
  - ***no parameter*** displays table for each instance listing status, configuration, and history.
  - **detail** displays data blocks for each instance and all ports on each instance.

Display Values

- **Root ID** Displays information on the ROOT ID (elected spanning tree root bridge ID):
  - **Priority** Priority of the bridge. Default value is ***32768***.
  - **Address** MAC address of the bridge.
- **Bridge ID** bridge status and configuration information for the locally configured bridge:
  - **Priority** Priority of the bridge. The default priority is ***32768***.
  - **Address** MAC address of the bridge.
  - **Hello Time** Interval (seconds) between bridge protocol data units (BPDUs) transmissions.
  - **Max Age** Maximum time that a BPDU is saved.
  - **Forward Delay** Time (in seconds) that is spent in the learning state.
- **Interface** STP configuration participants. Link-down interfaces are not shown.
- **Role** Role of the port as one of the following:

    - **Root** The best port for a bridge to a root bridge used for forwarding.
    - **Designated** A forwarding port for a LAN segment.
    - **Alternate** A port acting as an alternate path to the root bridge.
    - **Backup** A port acting as a redundant path to another bridge port.
- **State** Displays the interface STP state as one of the following:
    - **Learning**
    - **Discarding**
    - **Forwarding**
- **Cost** STP port path cost value.
- **Prio. Nbr.** STP port priority. Values range from **0** to **240**. Default is **128**.
- **Type** The link type of the interface (automatically derived from the duplex mode of an interface):
    - **P2p Peer (STP)** Point to point full duplex port running standard STP.
    - **shr Peer (STP)** Shared half duplex port running standard STP.

Examples

- This command displays STP data, including a table of port parameters.

```
switch# show spanning-tree vlan 1000
MST0
  Spanning tree enabled protocol rstp
  Root ID    Priority    32768
             Address     001c.7301.07b9
             Cost        1999 (Ext) 0 (Int)
             Port        101 (Port-Channel2)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.7304.195b
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 20000     128.4    P2p
Et5              designated forwarding 20000     128.5    P2p
Et6              designated forwarding 20000     128.6    P2p
Et23             designated forwarding 20000     128.23   P2p
Et26             designated forwarding 20000     128.26   P2p
Et32             designated forwarding 2000      128.32   P2p switch>
```

- This command displays output from the show spanning-tree command:

```
Switch# show spanning-tree
MST0
  Spanning tree enabled protocol mstp
  Root ID    Priority    32768
             Address     0011.2201.0301
             This bridge is the root

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     0011.2201.0301
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 2000      128.4    P2p
Et5              designated forwarding 2000      128.5    P2p
...
PEt4             designated forwarding 2000      128.31   P2p
PEt5             designated forwarding 2000      128.44   P2p
...
Po3              designated forwarding 1999      128.1003 P2p
```

- This command displays STP data, including an information block for each interface running STP.

```
switch# show spanning-tree vlan 1000 detail
 MST0 is executing the rstp Spanning Tree protocol
  Bridge Identifier has priority 32768, sysid 0, address 001c.7304.195b
```

```
  Configured hello time 2.000, max age 20, forward delay 15, transmit hold-count 6
  Current root has priority 32768, address 001c.7301.07b9
  Root port is 101 (Port-Channel2), cost of root path is  1999 (Ext) 0 (Int)
  Number of topology changes 4109 last change occurred 1292651 seconds ago
          from Ethernet13

 Port 4 (Ethernet4) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.4.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.4, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 452252, received 0, taggedErr 0, otherErr 0, rateLimiterCount 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400

Port 5 (Ethernet5) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.5.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.5, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 1006266, received 0, taggedErr 0, otherErr 0, rateLimiterCount 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400 switch#
```

12.1.4.17 show spanning-tree blockedports

The **`show spanning-tree blockedports`** command displays the list of blocked (discarding) ports.

Command Mode

EXEC

Command Syntax

`show spanning-tree blockedports`

Example

This command shows the ports that are in discarding state.

```
switch# show spanning-tree blockedports
Name   Blocked Interfaces List
------------------------------------------------------------------------
MST0   Po903, Po905, Po907, Po909, Po911, Po913, Po915, Po917, Po919, Po921, Po923

       Po925, Po927, Po929, Po931, Po933, Po935, Po939, Po941, Po943, Po945, Po947


Number of blocked ports (segments) in the system : 22 switch#
```

12.1.4.18 show spanning-tree counters

The **`show spanning-tree counters`** command displays the number of BPDU transactions on each interface running spanning tree.

Command Mode

EXEC

Command Syntax

`show spanning-tree counters`

Example

This command displays the BPDU counter status on each interface running spanning tree.

```
switch# show spanning-tree counters
```

```
          Port      Sent   Received   Tagged Error   Other Error  sinceTimer
-------------------------------------------------------------------------------
    Ethernet2   1008399          0              0             0           0
    Ethernet3   1008554          0              0             0           0
    Ethernet4    454542          0              0             0           0
    Ethernet5   1008556          0              0             0           0
    Ethernet6    827133          0              0             0           0
    Ethernet8   1008566          0              0             0           0
   Ethernet10    390732          0              0             0           0
   Ethernet11   1008559          0              0             0           0
   Ethernet15    391379          0              0             0           0
   Ethernet17    621253          0              0             0           0
   Ethernet19    330855          0              0             0           0
   Ethernet23    245243          0              0             0           0
   Ethernet25    591695          0              0             0           0
   Ethernet26   1007903          0              0             0           0
   Ethernet32   1010429          8              0             0           0
   Ethernet33    510227          0              0             0           0
   Ethernet34    827136          0              0             0           0
   Ethernet38   1008397          0              0             0           0
   Ethernet39   1008564          0              0             0           0
   Ethernet40   1008185          0              0             0           0
   Ethernet41   1007467          0              0             0           0
   Ethernet42     82925          0              0             0           0
Port-Channel1   1008551          0              0             0           0
Port-Channel2    334854     678589              0             0           3
Port-Channel3   1010420          4              0             0           0 switch#
```

12.1.4.19 show spanning-tree instance

The **show spanning-tree instance** command displays spanning tree protocol bridge configuration settings for each instance on the switch. The display includes **Bridge ID**, **Hello Time**, **Max Age**, and **Forward Delay** times.

The command also displays the restartability of the STP agent when the **detail** option is selected. A switch can continue support of MLAG operation when its peer is offline and the STP agent is unavailable.

Command Mode

EXEC

Command Syntax

show spanning-tree instance [**INFO_LEVEL**]

Parameters

INFO_LEVEL specifies level of information detail provided by the command.

- ***no parameter*** command displays information in a data table.
- **detail** command displays bridge information in data blocks for each instance.

Examples

- This command displays a bridge data table.

```
switch# show spanning-tree instance
                          Bridge ID                      Hello  Max  Fwd
Instance     Priority                 MAC addr           Time   Age  Dly
----------   ------------------------------------     -----  ---  ---
MST0         32768(32768, sys-id 0   ) 001c.7302.2f98      2000   20   15
MST101       32869(32768, sys-id 101 ) 001c.7302.2f98      2000   20   15
MST102       32870(32768, sys-id 102 ) 001c.7302.2f98      2000   20   15 switch#
```

- This command displays bridge data blocks.

```
switch# show spanning-tree instance detail
Stp Detailed Status:
   Stp agent restartable                      :            True
   MST-PVST interoperation                    :        Disabled
   Stp heartbeat timeout                      :             2.0
   Last local heartbeat timeout               :     0:04:07 ago
   Local heartbeat timeout since reboot       :               1
```

```
MST0
  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST101
  Bridge ID  Priority    32869  (priority 32768 sys-id-ext 101)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST102
  Bridge ID  Priority    32870  (priority 32768 sys-id-ext 102)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec switch#
```

12.1.4.20 show spanning-tree instance detail

The **`show spanning-tree instance detail`** command displays detailed MST information including MSTP-Rapid PVST+ interoperation status.

Command Mode

EXEC

Command Syntax

`show spanning-tree instance detail`

Example

This command displays detailed MST information.

```
switch# show spanning-tree instance detail
Stp Detailed Status:
 Stp agent restartable : True
 MST-PVST interoperation : Enabled
 Stp heartbeat timeout : 2.0
 Last local heartbeat timeout : 36 days, 19:10:46 ago
 Local heartbeat timeout since reboot : 1
MST0
 Bridge ID Priority 32768 (priority 32768 sys-id-ext 0)
 Address 001c.7374.8572
 Hello Time 2.000 sec Max Age 20 sec Forward Delay 15 sec switch#
```

12.1.4.21 show spanning-tree interface

The **`show spanning-tree interface`** command displays spanning tree protocol information for the specified interface.

Command Mode

EXEC

Command Syntax

`show spanning-tree interface` **INT_NAME** [**INFO_LEVEL**]

Parameters

- **INT_NAME** Interface type and number. Values include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **peer ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **peerport-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **INFO_LEVEL** specifies level of detail provided by the output. Options include:
  - ***no parameter*** command displays a table of STP data for the specified interface.

- **detail** command displays a data block for the specified interface.

Examples

- This command displays an STP table for Ethernet interface 5.

```
switch# show spanning-tree interface ethernet 5
Instance         Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- ----
MST0             designated forwarding 20000     128.5    P2p switch#
```

- This command displays a data block for Ethernet interface 5.

```
switch# show spanning-tree interface ethernet 5 detail
 Port 5 (Ethernet5) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.5.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.5, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 1008766, received 0, taggedErr 0, otherErr 0, rateLimiterCount
 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400 switch#
```

12.1.4.22 show spanning-tree mst

The **show spanning-tree mst** command displays configuration and state information for Multiple Spanning Tree protocol (MST) instances.

Command Mode

EXEC

Command Syntax

show spanning-tree mst [**INSTANCE**] [**INFO_LEVEL**]

Parameters

- **INSTANCE** MST instance for which the command displays information. Options include:
  - ***no parameter*** all MST instances.
  - **mst_inst** MST instance number. Value of ***mst_inst*** ranges from ***0*** to ***4094***.
- **INFO_LEVEL** type and amount of information in the output. Options include:
  - ***no parameter*** output is interface data in tabular format.
  - **detail** output is a data block for each interface.

Examples

- This command displays interface data blocks for MST instance ***3***.

```
switch# show spanning-tree mst 3 detail
##### MST3vlans mapped:3
Bridgeaddress 0011.2233.4402priority32771(32768 sysid 3)
Rootaddress 0011.2233.4401priority32771(32768 sysid 3)

Ethernet1 of MST3 is root forwarding
Port infoport id128.1priority128cost2000
Designated rootaddress 0011.2233.4401priority32768cost0
Designated bridgeaddress 0011.2233.4401priority32768portid128.1
Ethernet2 of MST3 is alternate discarding
```

```
Port infoport id128.2 priority128cost2000
Designated rootaddress 0011.2233.4401 priority32768cost0
Designated bridgeaddress 0011.2233.4401 priority32768port id128.2

Ethernet3 of MST3 is designated forwarding
Port infoport id128.3 priority128cost2000
Designated rootaddress 0011.2233.4401 priority32768cost2000
Designated bridgeaddress 0011.2233.4402 priority32768port id128.3
```

- This command displays interface tables for all MST instances.

```
switch# show spanning-tree mst
##### MST0vlans mapped:1,4-4094
Bridgeaddress 0011.2233.4402priority32768 (32768 sysid 0)
Rootaddress 0011.2233.4401priority32768 (32768 sysid 0)
Regional Root address 0011.2233.4401priority32768 (32768 sysid 0)

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------
Et1              root       forwarding  2000     128.1    P2p
Et2              alternate  discarding  2000     128.2    P2p
Et3              designated forwarding  2000     128.3    P2p
Et4              designated forwarding  2000     128.4    P2p

##### MST2 vlans mapped: 2
Bridgeaddress 0011.2233.4402priority8194 (8192 sysid 2)
Rootthis switch for MST2

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- -----------
Et1              designated forwarding 2000      128.1    P2p
Et2              designated forwarding 2000      128.2    P2p
Et3              designated forwarding 2000      128.3    P2p
Et4              designated forwarding 2000      128.4    P2p

##### MST3 vlans mapped: 3
Bridgeaddress 0011.2233.4402priority32771 (32768 sysid 3)
Rootaddress 0011.2233.4401priority32771 (32768 sysid 3)

Interface        Role       State      Cost      Prio.Nbr  Type
---------------- ---------- ---------- --------- -------- ------------
Et1              root       forwarding 2000      128.1    P2p
Et2              alternate  discarding 2000      128.2    P2p
Et3              designated forwarding 2000      128.3    P2p
Et4              designated forwarding 2000      128.4    P2p
```

12.1.4.23 show spanning-tree mst configuration

The **show spanning-tree mst configuration** command displays information about the MST regions VLAN-to-instance mapping. The command provides two display options:

- **default** displays a table that lists the instance to VLAN map.
- **digest** displays the configuration digest.

The configuration digest is a 16-byte hex string calculated from the md5 encoding of the VLAN-to-instance mapping table. Switches with identical mappings have identical digests.

Command Mode

EXEC

Command Syntax

show spanning-tree mst configuration [**INFO_LEVEL**]

Parameters

INFO_LEVEL specifies data provided by the output. Options include:

- ***no parameter*** command displays VLAN-to-instance map.
- **digest** command displays the MST configuration digest.

Examples

- This command displays the MST regions VLAN-to-instance map.

```
switch# show spanning-tree mst configuration
Name      []
Revision  0    Instances configured 3

Instance  Vlans mapped
------------------------------------------------------------
01,4-4094
22
33
------------------------------------------------------------
switch#
```

- This command displays the MST regions configuration digest.

```
switch# show spanning-tree mst configuration digest
Name      []
Revision  0    Instances configured 1
Digest         0xAC36177F50283CD4B83821D8AB26DE62 switch#
```

12.1.4.24 show spanning-tree mst interface

The **show spanning-tree mst interface** command displays Multiple Spanning Tree Protocol (MSTP) information for a specified interface on the specified MST instances.

Command Mode

EXEC

Command Syntax

show spanning-tree mst [**INSTANCE**] **interface** **INT_NAME** [**INFO_LEVEL**]

Parameters

- **INSTANCE** MST instance for which the command displays information. Options include:
  - ***no parameter*** all MST instances.
  - **mst_inst** denotes a single MST instance. Value of ***mst_inst*** ranges from ***0*** to ***4094***.
- **INT_NAME** Interface type and number. Values include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **peerethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **peerport-channel *p_num*** Port-channel interface specified by ***p_num***.
- **INFO_LEVEL** specifies level of detail provided by the output. Options include:
  - ***no parameter*** command displays a table of STP instance data for the specified interface.
  - **detail** command displays a data block for all specified instance-interface combinations.

Examples

- This command displays an table of STP instance data for ***interface ethernet 1***:

```
switch# show spanning-tree mst interface ethernet 1
Ethernet1 of MST0 is root forwarding
Edge port: nobpdu guard: disabled
Link type: point-to-point
Boundary : Internal
Bpdus sent 2120, received 2164, taggedErr 0, otherErr 0

Instance Role Sts CostPrio.Nbr Vlans mapped
-------- ---- --- --------- -------- -------------------------
0RootFWD2000128.11,4-4094
2DesgFWD2000128.12
3RootFWD2000128.13
```

- This command displays blocks of STP instance information for ***interface ethernet 1***.

```
switch# show spanning-tree mst 3 interface ethernet 1 detail
Edge port: nobpdu guard: disabled
Link type: point-to-point
Boundary : Internal
Bpdus sent 2321, received 2365, taggedErr 0, otherErr 0

Ethernet1 of MST3 is root forwarding
Vlans mapped to MST3 3
Port infoport id128.1priority128 cost2000
Designated rootaddress 0011.2233.4401priority32768 cost0
Designated bridgeaddress 0011.2233.4401priority32768 port id128.1
```

12.1.4.25 show spanning-tree mst test information

The **`show spanning-tree mst test information`** displays diagnostic spanning tree protocol information.

Command Mode

EXEC

Command Syntax

`show spanning-tree mst test information`

Example

This command displays diagnostic STP information.

```
switch# show spanning-tree mst test information
bi = MstInfo.BridgeInfo( "dut" )
bi.stpVersion = "rstp"
bi.mstpRegionId = ""
bi.bridgeAddr = "00:1c:73:01:60:17"
si = MstInfo.BridgeStpiInfo( "Mst" )
bi.stpiInfoIs( "Mst", si )
si.cistRoot = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
si.cistPathCost = 0
bmi = MstInfo.BridgeMstiInfo( "Mst0" )
bmi.bridgeId = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
bmi.designatedRoot = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
si.mstiInfoIs( "Mst0", bmi )
bmii = MstInfo.BridgeMstiIntfInfo( "Mst0", "Ethernet15" )
bmii.portId = Tac.Value( "Stp::PortId",
```

```
                    portPriority=128, portNumber=15 )
bmii.role = "designated"
bmii.operIntPathCost = 2000
bmii.fdbFlush = 1
bmi.mstiIntfInfoIs( "Ethernet15", bmii )
bii = MstInfo.BridgeIntfInfo( "Ethernet15" )
bii.operExtPathCost = 2000
si.intfInfoIs( "Ethernet15", bii )
bmii = MstInfo.BridgeMstiIntfInfo( "Mst0", "Port-Channel10" )
bmii.portId = Tac.Value( "Stp::PortId",
                    portPriority=128, portNumber=101 )
bmii.role = "designated"
bmii.operIntPathCost = 1999
bmii.fdbFlush = 1
bmi.mstiIntfInfoIs( "Port-Channel10", bmii )
bii = MstInfo.BridgeIntfInfo( "Port-Channel10" )
bii.operExtPathCost = 1999
si.intfInfoIs( "Port-Channel10", bii )
switch>
```

12.1.4.26 show spanning-tree root

The **`show spanning-tree root`** command displays the Bridge-ID, cost to the root bridge, root port, and the root bridge timer settings for all instances.

Command Mode

EXEC

Command Syntax

`show spanning-tree root` [**INFO_LEVEL**]

Parameters

INFO_LEVEL specifies output format. Options include:

- ***no parameter*** output displays data in tabular format.
- **detail** output displays a data block for each instance.

Examples

- This command displays a table of root bridge information.

```
switch# show spanning-tree root
                  Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr   Cost    Time   Age  Dly  Root Port
----------   ------------------- --------- -----  ---  ---  ------------
MST0         32768 001c.7301.23de         0    2    20  15   Po937
MST101       32869 001c.7301.23de      3998    0     0   0   Po909
MST102       32870 001c.7301.23de      3998    0     0   0   Po911 switch>
```

- This command displays root bridge data blocks for each MSTP instance.

```
switch# show spanning-tree root detail
MST0
MST0
  Root ID    Priority    32768
             Address     001c.7301.23de
             Cost        0 (Ext) 3998 (Int)
             Port        100 (Port-Channel937)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST101
  Root ID    Priority    32869
             Address     001c.7301.23de
             Cost        3998
             Port        107 (Port-Channel909)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec
MST102
```

```
  Root ID    Priority    32870
             Address     001c.7301.23de
             Cost        3998
             Port        104 (Port-Channel911)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec switch>
```

12.1.4.27 show spanning-tree topology status

The **show spanning-tree topology status** command displays the forwarding state of ports on the specified VLANs.

Command Mode

EXEC

Command Syntax

show spanning-tree topology [**VLAN_NAME**] **status** [**INFO_LEVEL**]

Parameters

- **VLAN_NAME** specifies the VLANs that the output displays. Options include:
  - ***no parameter*** output includes all VLANs.
  - **vlan** output includes all VLANs.
  - **vlan *v_num*** command includes specified VLAN; ***v_num*** ranges from ***1*** to ***4094***.
- **INFO_LEVEL** specifies information provided by output. Options include:
  - ***no parameter*** output lists forwarding state of interfaces.
  - **detail** output lists forwarding state and change history of interfaces.

Examples

- This command displays forwarding state for ports mapped to all VLANs.

```
switch# show spanning-tree topology status
Topology: Cist
Mapped Vlans: 1-4,666,1000-1001,1004-1005
Cpu:               forwarding
Ethernet2:         forwarding
Ethernet3:         forwarding
Ethernet4:         forwarding
Ethernet5:         forwarding
Ethernet6:         forwarding
Ethernet8:         forwarding
Ethernet10:        forwarding
Port-Channel1:     forwarding
Port-Channel2:     forwarding
Port-Channel3:     forwarding switch>
```

- This command displays forwarding state and history for ports mapped to ***vlan 1000***.

```
switch# show spanning-tree topology vlan 1000 status detail
Topology: Cist
Mapped Vlans: 1000
Cpu:               forwarding (1 changes, last 23 days, 22:54:43 ago)
Ethernet2:         forwarding (3 changes, last 23 days, 22:48:59 ago)
Ethernet4:         forwarding (3 changes, last 10 days, 19:54:17 ago)
Ethernet5:         forwarding (3 changes, last 23 days, 22:54:38 ago)
Ethernet6:         forwarding (3 changes, last 19 days, 15:49:10 ago)
Ethernet10:        forwarding (3 changes, last 9 days, 7:37:05 ago)
Port-Channel1:     forwarding (3 changes, last 23 days, 22:54:34 ago)
Port-Channel3:     forwarding (5 changes, last 21 days, 4:56:41 ago)
```

```
switch>
```

12.1.4.28 show spanning-tree transmit active

The **show spanning-tree transmit active** command displays spanning tree protocol bridge assurance information for network ports or for all ports. Bridge assurance-enabled ports will not necessarily be blocked when they link to a port where bridge assurance is not enabled, but if they do not receive periodic BPDUs from the other side of the link the **show spanning-tree transmit active** command will show a bridge assurance status of inconsistent (blocking) for that port.

Command Mode

EXEC

Command Syntax

show spanning-tree transmit active **INFO_LEVEL**

Parameters

INFO_LEVEL specifies level of information detail provided by the command.

- ***no parameter*** command displays bridge assurance information for network ports.
- **detail** command displays bridge assurance information for all ports.

Example

This command displays the bridge assurance status of network ports.

```
switch# show spanning-tree transmit active
Name                 Bridge Assurance Status
-------------------------------------------
VL1        Et5/1       consistent

Number of bridge assurance inconsistent ports in the system : 0 switch>
```

12.1.4.29 shutdown (Loop-protection)

The **shutdown** command disables loop protection globally. The feature is disabled by default, and is enabled by using the **no shutdown** command.

Note: To function, loop protection must also be enabled on a per-VLAN basis using the **protect vlan** command.

Command Mode

Loop-protection Configuration

Command Syntax

shutdown

no shutdown

Example

This command enables loop protection globally on the switch.

```
switch(config-monitor-loop-protect)# no shutdown
switch(config-monitor-loop-protect)#
```

12.1.4.30 spanning-tree bpdu tx hold-count

The **spanning-tree bpdu tx hold-count** command specifies the maximum number of BPDUs per second that the switch can send from an interface. Valid settings range from ***1*** to ***10*** BPDUs with a default of ***6*** BPDUs.

The **no spanning-tree bpdu tx hold-count** and **default spanning-tree bpdu tx hold-count** commands restore the transmit hold count default of ***6*** BPDUs by removing the **spanning-tree bpdu tx hold-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree bpdu tx hold-count *max_bpdu*

no spanning-tree bpdu tx hold-count

default spanning-tree bpdu tx hold-count

Parameters

max_bpdu BPDU packets. Value ranges from ***1*** to ***10***. Default is ***6***.

Example

This command configures a transmit hold-count of ***8*** BPDUs.

```
switch(config)# spanning-tree bpdu tx hold-count 8
switch(config)#
```

12.1.4.31 spanning-tree bpdufilter

The **spanning-tree bpdufilter** command controls bridge protocol data unit (BPDU) filtering on the configuration mode interface. BPDU filtering is disabled by default.

- **spanning-tree bpdufilter enabled** enables BPDU filtering.
- **spanning-tree bpdufilter disabled** disables BPDU filtering by removing the **spanning-tree bpdufilter** command from ***running-config***.

The BPDU filter default setting for portfast ports is configured by the **spanning-tree edge-port bpdufilter default** command; BPDU filter is disabled by default on all non-portfast ports.

The **no spanning-tree bpdufilter** and **default spanning-tree bpdufilter** commands restore the global BPDU filter setting on the configuration mode interface by removing the corresponding **spanning-tree bpdufilter** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree bpdufilter FILTER_STATUS

no spanning-tree bpdufilter

default spanning-tree bpdufilter

Parameters

FILTER_STATUS BPDU filtering status. Options include:

- **enabled** BPDU filter is enabled on the interface.

- **disabled** BPDU filter is disabled on the interface.

Example

This command enables BPDU filtering on ***interface ethernet 5***.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpdufilter enabled
switch(config-if-Et5)#
```

12.1.4.32 spanning-tree bpduguard

The **spanning-tree bpduguard** command controls BPDU guard on the configuration mode interface. A BPDU guard-enabled port is disabled when it receives a BPDU packet.

The BPDU guard default setting for portfast ports is configured by the spanning-tree edge-port bpduguard default command; BPDU guard is disabled by default on all non-portfast ports.

The **no spanning-tree bpduguard** and **default spanning-tree bpduguard** commands restore the global BPDU guard setting on the configuration mode interface by removing the corresponding **spanning-tree bpduguard** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree bpduguard GUARD_ACTION

no spanning-tree bpduguard

default spanning-tree bpduguard

Parameters

GUARD_ACTION BPDU guard setting. Options include:

- **disable** Disable bpduguard.
- **enable** Enable bpduguard.
- **rate-limit** BPDU Input Rate Limiter options.

Example

These commands enable BPDU guard on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree bpduguard enabled
switch(config-if-Et5)
```

12.1.4.33 spanning-tree bpduguard rate-limit count (global)

The **spanning-tree bpduguard rate-limit count** command sets the maximum BPDU reception rate (quantity per interval) for ports that are not covered by a spanning-tree bpduguard rate-limit count (interface) command.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs they receive in excess of the specified limit. BPDU rate limiting is enabled or disabled by spanning-tree bpduguard rate-limit enable / disable commands.

The **no spanning-tree bpduguard rate-limit count** and **default spanning-tree bpduguard rate-limit count** commands restore the global setting to its default value by removing the **spanning-tree bpduguard rate-limit count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree bpduguard rate-limit count ***max_bpdu*** [interval period]

no spanning-tree bpduguard rate-limit count

default spanning-tree bpduguard rate-limit count

Parameters

- ***max_bpdu*** configures the maximum number of BPDUs per timer interval. The value of *BPDU quantity* ranges from ***1*** to ***20000***.
- **interval** ***period*** configures the timer interval in seconds. The value of ***period*** ranges from ***1*** to ***15***.

Guidelines

Arista Networks recommends to retain the default values of rate limit.

In the PVST mode, when the VLAN membership of a port is changed by a significant margin, it is advisable to disable interface BPDU rate limit on both ends of a port. For example, if three VLANs are present on a port initially, the operator must first add ***300*** more VLANs on one side of the port and then add the same 300 VLANs on the other side of the port. In this case, if the VLANs are increased towards the root bridge first, then the other side can cross the rate-limit threshold.

Example

This command configures the global rate limit as ***5000*** BPDUs per ***4*** second period.

```
switch(config)# spanning-tree bpduguard rate-limit count 5000 interval 4
switch(config)#
```

12.1.4.34 spanning-tree bpduguard rate-limit count (interface)

The **spanning-tree bpduguard rate-limit count** command configures the maximum BPDU reception rate for the configuration mode interface. The default rate limit is specified by the spanning-tree bpduguard rate-limit count (global) command.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limiting is enabled or disabled by spanning-tree bpduguard rate-limit enable / disable commands.

The **no spanning-tree bpduguard rate-limit count** and **default spanning-tree bpduguard rate-limit count** commands restore the interface value to the global setting by removing the corresponding **spanning-tree bpduguard rate-limit count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree bpduguard rate-limit count ***max_bpdu*** [**TIMER**]

no spanning-tree bpduguard rate-limit count

default spanning-tree bpduguard rate-limit count

Parameters

- ***max_bpdu*** BPDU quantity. Value ranges from ***1*** to ***20000***.
- **TIMER** BPDU reception interval (seconds). Options include:
  - ***period*** reception interval defaults to **hello-time**.
  - **interval** ***period*** Value of ***period*** ranges from ***1*** to ***15***.

Example

These commands configure rate limit as ***7500*** BPDUs per ***8*** second period on ***interface Ethernet 2***.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# spanning-tree bpduguard rate-limit count 7500 interval 8
switch(config-if-Et2)#
```

12.1.4.35 spanning-tree bpduguard rate-limit default

The **spanning-tree bpduguard rate-limit default** command configures the global BPDU rate limit setting. The global BPDU rate limit setting provides the default for individual ports whose configuration does not include a spanning-tree bpduguard rate-limit enable / disable command. The default global setting is enabled.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limits are established by spanning-tree bpduguard rate-limit count (global) commands.

The **no spanning-tree bpduguard rate-limit default** sets the global BPDU rate limit setting to disabled. The **spanning-tree bpduguard rate-limit default** and **default spanning-tree bpduguard rate-limit default** commands restore the default global rate limit setting to enabled by removing the **nospanning-tree bpduguard rate-limit default** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree bpduguard rate-limit default

no spanning-tree bpduguard rate-limit default

default spanning-tree bpduguard rate-limit default

Example

This command enables rate limiting on all ports not covered by an interface rate limit command.

```
switch(config)# spanning-tree bpduguard rate-limit default
switch(config)#
```

12.1.4.36 spanning-tree bpduguard rate-limit enable / disable

These commands enable and disable BPDU rate limiting on the configuration mode interface:

- **spanning-tree bpduguard rate-limit enable** enables BPDU rate limiting.
- **spanning-tree bpduguard rate-limit disable** disables BPDU rate limiting.

The spanning-tree bpduguard rate-limit default command enables BPDU rate limiting on all ports not configured with a **spanning-tree bpduguard rate-limit** command.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limits are established by spanning-tree bpduguard rate-limit count (interface) commands.

The **no spanning-tree bpduguard rate-limit** and **default spanning-tree bpduguard rate-limit** commands restore the global rate limit setting on the configuration mode interface by removing the corresponding **spanning-tree bpduguard rate-limit** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree bpduguard rate-limit enable

spanning-tree bpduguard rate-limit disable

no spanning-tree bpduguard rate-limit

default spanning-tree bpduguard rate-limit

Example

These commands enable rate limiting on ***interface ethernet 15***.

```
switch(config)# interface ethernet 15
switch(config-if-Et15)# spanning-tree bpduguard rate-limit enable
switch(config-if-Et15)#
```

12.1.4.37 spanning-tree cost

The **spanning-tree cost** command configures the path cost of the configuration mode interface. Cost values range from ***1*** to ***200000000*** (***200*** million). The default cost depends on the interface speed:

- ***1*** gigabit interface: cost = ***20000***
- ***10*** gigabit interface: cost = ***2000***

The **spanning-tree cost** command provides a mode option:

- RST instance cost is configured by not including a mode.
- MST instance 0 cost is configured by not including a mode or with the **mst** mode option.
- MST instance cost is configured with the **mst** mode option.
- Rapid-PVST VLAN cost is configured with the **vlan** mode option.

The **no spanning-tree cost** and **default spanning-tree cost** commands restore the default cost on the configuration mode interface by removing the corresponding **spanning-tree cost** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree **MODE** **cost** *value*

no spanning-tree **MODE** **cost**

default spanning-tree **MODE** **cost**

Parameters

- **MODE** specifies the spanning tree instances for which the cost is configured. Values include:
  - ***no parameter*** RST instance, MST instance ***0***, or all Rapid-PVST instances permitted on the interface.

- **mst *m_range*** specified MST instances. ***m_range*** formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from ***0*** to ***4094***.
- **vlan *v_range*** specified Rapid-PVST instances. ***v_range*** formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from ***1*** to ***4094***.***value*** path cost assigned to interface. Values range from ***1*** to ***200000000*** (***200*** million). Default values are ***20000*** (1 G interfaces) or ***2000*** (10 G interfaces).

Examples

- These commands configure a port cost of ***25000*** for ***interface Ethernet 5*** when configured as an RST port, as a port in MST instance ***0***, or all unconfigured Rapid-PVST instances that are not explicitly configured.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning tree cost 25000
```

- This command configures a port cost of ***30000*** for ***interface Ethernet 5*** when configured as a port in MST instance ***200***.

```
switch(config-if-Et5)# spanning tree mst 200 cost 30000
```

- This command configures a port cost of ***100000*** for ***interface Ethernet 5*** when configured as a port in VLANs ***200-220***.

```
switch(config-if-Et5)# spanning tree vlan 200-220 cost 100000
switch(config-if-Et5)#
```

12.1.4.38 spanning-tree edge-port bpdufilter default

The **`spanning-tree edge-port bpdufilter default`** command configures the global BPDU filter setting as **enabled**. Ports not covered by a spanning-tree bpdufilter command use the global BPDU filter setting.

Command Mode

Global Configuration

Command Syntax

`spanning-tree edge-port bpdufilter default`

`no spanning-tree edge-port bpdufilter default`

`default spanning-tree edge-port bpdufilter default`

Example

This command configures the BPDU filter global setting to **enabled**.

```
switch(config)# spanning-tree edge-port bpdufilter default
switch(config)#
```

12.1.4.39 spanning-tree edge-port bpduguard default

The **`spanning-tree edge-port bpduguard default`** command sets the global BPDU guard setting as **enabled**. Ports not covered by a spanning-tree bpduguard command use the global BPDU guard setting.

Command Mode

Global Configuration

Command Syntax

spanning-tree edge-port bpduguard default

no spanning-tree edge-port bpduguard default

default spanning-tree edge-port bpduguard default

Example

This command configures the global BPDU guard setting to enabled.

```
switch(config)# spanning-tree edge-port bpduguard default
switch(config)#
```

12.1.4.40 spanning-tree forward-time

The **spanning-tree forward-time** command configures the forward delay timer. Forward delay is the time that a port is in learning state before it begins forwarding data packets.

The switch inserts the forward delay timer value in BPDU packets it sends as the root bridge. The forward delay value ranges from ***4*** to ***30*** seconds with a default of ***15*** seconds.

The **no spanning-tree forward-time** and **default spanning-tree forward-time** commands restore the forward delay timer default of ***15*** seconds by removing the **spanning-tree forward-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree forward-time ***period***

no spanning-tree forward-time

default spanning-tree forward-time

Parameters

period forward delay timer (seconds). Value ranges from ***4*** to ***30***. Default is ***15***.

Example

This command sets the forward delay timer value to ***25*** seconds.

```
switch(config)# spanning-tree forward-time 25
switch(config)#
```

12.1.4.41 spanning-tree guard

The **spanning-tree guard** command enables root guard or loop guard on the configuration mode interface. The **spanning-tree guard loop default** command configures the global loop guard setting.

- Root guard prevents a port from becoming a root or blocked port. A root guard port that receives a superior BPDU transitions to the root-inconsistent (blocked) state.
- Loop guard protects against loops resulting from unidirectional link failures on point-to-point links by preventing non-designated ports from becoming designated ports. When loop guard is enabled, a root or blocked port transitions to loop-inconsistent (blocked) state if it stops receiving BPDUs from its designated port. The port returns to its prior state when it receives a BPDU.

The **no spanning-tree guard** and **default spanning-tree guard** commands sets the configuration mode interface to the global loop guard mode by removing the **spanning-tree guard** statement from ***running-config***. The **spanning-tree guard none** command disables loop guard and root guard on the interface, overriding the global setting.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

`spanning-tree guard` PORT_MODE

`no spanning-tree guard`

`default spanning-tree guard`

Parameters

PORT_MODE the port mode. Options include:

- **loop** enables loop guard on the interface.
- **root** enables root guard on the interface.
- **none** disables root guard and loop guard.

Example

This command enables root guard on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree guard root
switch(config-if-Et5)#
```

12.1.4.42 spanning-tree guard loop default

The **`spanning-tree guard loop default`** command configures the global loop guard setting as **enabled**. Ports not covered by a spanning-tree guard command use the global loop guard setting. Loop guard prevents blocked or root ports from becoming a designated port due to failures resulting in a unidirectional link. The spanning-tree guard interface configuration statement overrides the global setting for a specified interface. The default global loop guard setting is disabled.

The **`no spanning-tree guard loop default`** and **`default spanning-tree guard loop default`** commands restore the global loop guard setting of disabled by removing the **`spanning-tree guard loop default`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`spanning-tree guard loop default`

`no spanning-tree guard loop default`

`default spanning-tree guard loop default`

Example

This command enables loop guard as the default on all switch ports.

```
switch(config)# spanning-tree guard loop default
switch(config)#
```

12.1.4.43 spanning-tree hello-time

The **spanning-tree hello-time** command configures the hello time, which specifies the transmission interval between consecutive bridge protocol data units (BPDU) that the switch sends as a root bridge. The hello time is also inserted in outbound BPDUs.

This hello time ranges from ***0.2*** seconds to ***10*** seconds with a default of ***2*** seconds.

The **no spanning-tree hello-time** and **default spanning-tree hello-time** commands restore the hello time default of 2 seconds by removing the **spanning-tree hello-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree hello-time ***period***

no spanning-tree hello-time

default spanning-tree hello-time

Parameters

period hello-time (milliseconds). Value ranges from ***200*** to ***10000***. Default is ***2000***.

Example

This command configures a hello-time of ***1*** second.

```
switch(config)# spanning-tree hello-time 1000
switch(config)#
```

12.1.4.44 spanning-tree link-type

The **spanning-tree link-type** command specifies the configuration mode interfaces link type, which is normally derived from the ports duplex setting. The default setting depends on a ports duplex mode:

- full-duplex ports are **point-to-point**.
- half-duplex ports are **shared**.

The **no spanning-tree link-type** and **default spanning-tree link-type** commands restore the default link type on the configuration mode interface by removing the corresponding **spanning-tree link-type** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree link-type TYPE

no spanning-tree link-type

default spanning-tree link-type

Parameters

TYPE link type of the configuration mode interface. Options include:

- **point-to-point**
- **shared**

Example

This command configures ***interface ethernet 5*** as a shared port.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree link-type shared
switch(config-if-Et5)#
```

12.1.4.45 spanning-tree max-age

The **spanning-tree max-age** command configures the switchs max age timer, which specifies the max age value that the switch inserts in outbound BPDU packets it sends as a root bridge. The max-age time value ranges from ***6*** to ***40*** seconds with a default of ***20*** seconds.

Max age is the interval, specified in the BPDU, that BPDU data remains valid after its reception. The bridge recomputes the spanning tree topology if it does not receive a new BPDU before max age expiry.

The **no spanning-tree max-age** and **default spanning-tree max-age** commands restore the max-age default of ***20*** seconds by removing the **spanning-tree max-age** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree max-age ***period***

no spanning-tree max-age

default spanning-tree max-age

Parameters

period max age period (seconds). Value ranges from ***6*** to ***40***. Default is ***20***.

Example

This command sets the max age timer value to ***25*** seconds.

```
switch(config)# spanning-tree max-age 25
switch(config)#
```

12.1.4.46 spanning-tree max-hops

The **spanning-tree max-hops** command specifies the max hop setting that the switch inserts into BPDUs that it sends out as the root bridge. The max hop setting determines the number of bridges in an MST region that a BPDU can traverse before it is discarded. The max-hop value ranges from ***1*** to ***40*** with a default of ***20***.

The **no spanning-tree max-hops** and **default spanning-tree max-hops** commands restore the max-hops setting to its default value of ***20*** by removing the **spanning-tree max-hops** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree max-hops ***ports***

no spanning-tree max-hops

default spanning-tree max-hops

Parameters

ports max hops (bridges). Value ranges from ***1*** to ***40***. Default is ***20***.

Example

This command sets the max hop value to ***40***.

```
switch(config)# spanning-tree max-hop 40
switch(config)#
```

12.1.4.47 spanning-tree mode

The **`spanning-tree mode`** command specifies the spanning tree protocol version that the switch runs. The default mode is Multiple Spanning Tree Protocol (MSTP).

The **`no spanning-tree mode`** and **`default spanning-tree mode`** commands restore the default spanning tree protocol version.

Note: The **`spanning-tree mode`** command may disrupt user traffic. When the switch starts a different STP version, all spanning-tree instances are stopped, then restarted in the new mode.

Command Mode

Global Configuration

Command Syntax

`spanning-tree mode` **VERSION**

`no spanning-tree mode`

`default spanning-tree mode`

Parameters

VERSION spanning tree version that the switch runs. Options include:

- **mstp** multiple spanning tree protocol described in the ***IEEE 802.1Q-2005*** specification and originally specified in the ***IEEE 802.1s*** specification.
- **rstp** rapid spanning tree protocol described in the ***IEEE 802.1D-2004*** specification and originally specified in the ***IEEE 802.1w*** specification.
- **rapid-pvst** rapid per-VLAN spanning tree protocol described in the ***IEEE 802.1D-2004*** specification and originally specified in the ***IEEE 802.1w*** specification.
- **`backup`** disables STP and enables switchport interface pairs configured with the **`switchport backup-link`** command.
- **`none`** disables STP. The switch does not generate STP packets. Each switchport interface forwards data packets to all connected ports and forwards STP packets as multicast data packets on the VLAN where they are received.

Guidelines

Backup mode is not available on Trident platform switches.

Example

This command configures the switch to run multiple spanning tree protocol.

```
switch(config)# spanning-tree mode mstp
switch(config)#
```

12.1.4.48 spanning-tree mst configuration

The **`spanning-tree mst configuration`** command places the switch in MST-configuration mode, which is the group change mode where MST region parameters are configured.

Changes made in a group change mode are saved by leaving the mode through the **exit** command or by entering another configuration mode. To discard changes from the current edit session, leave the mode with the **abort** command.

These commands are available in MST-configuration mode:

- **abort (mst-configuration mode)**
- **exit (mst-configuration mode)**
- **instance**
- **name (mst-configuration mode)**
- **revision (mst-configuration mode)**
- **show (mst-configuration mode)**

The **no spanning-tree mst configuration** and **default spanning-tree mst configuration** commands restore the MST default configuration.

Command Mode

Global Configuration

Command Syntax

spanning-tree mst configuration

no spanning-tree mst configuration

default spanning-tree mst configuration

Examples

- This command enters MST configuration mode.

```
switch(config)# spanning-tree mst configuration
switch(config-mst)#
```

- This command exits MST configuration mode, saving MST region configuration changes to ***running-config***.

```
switch(config-mst)# exit
switch(config)#
```

- This command exits MST configuration mode without saving MST region configuration changes to ***running-config***.

```
switch(config-mst)# abort
switch(config)#
```

12.1.4.49 spanning-tree mst pvst border

The **spanning-tree mst pvst border** command configures MSTP PVST border feature to automatically detect border ports facing PVST+ regions. By default, spanning-tree mst pvst border is disabled.

The **no spanning-tree mst pvst border** and **default spanning-tree mst pvst border** commands restore the default MST configuration.

Command Mode

MST Configuration

Command Syntax

spanning-tree mst pvst border

no spanning-tree mst pvst border

default spanning-tree mst pvst border

Example

This command enters the MST configuration mode.

```
switch(config)# spanning-tree mst configuration
switch(config-mst)# spanning-tree mst pvst border
```

12.1.4.50 spanning-tree portchannel guard misconfig

The **spanning-tree portchannel guard misconfig** command enables the switch to detect misconfigured port channels that may cause network loops by monitoring inbound BPDUs. When a port channel receives ***75*** inconsistent BPDUs within ***30*** seconds, the switch error disables the port. When a port channel receives ***5*** BPDUs with the same source BPDU during the ***30*** second measurement interval, the error counter is reset and the port continues normal port channel operation. Misconfigured port channel detection is disabled by default.

The **no spanning-tree portchannel guard misconfig**and **default spanning-tree portchannel guard misconfig**commands disables the detection of misconfigured port channels by removing the **spanning-tree portchannel guard misconfig** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree portchannel guard misconfig

no spanning-tree portchannel guard misconfig

default spanning-tree portchannel guard misconfig

spanning-tree etherchannel guard misconfig

no spanning-tree etherchannel guard misconfig

default spanning-tree etherchannel guard misconfig

Guidelines

The **spanning-tree portchannel guard misconfig** and **spanning-tree etherchannel guard misconfig** commands are equivalent.

Examples

- This command enables port channel misconfiguration detection on the switch.

```
switch(config)# spanning-tree portchannel guard misconfig
switch(config)# show running-config
!
spanning-tree mode mstp
spanning-tree portchannel guard misconfig
!

!
end
switch(config)#
```

- This command disables port channel misconfiguration detection on the switch.

```
switch(config)# no spanning-tree portchannel guard misconfig
switch(config)# show running-config
!
```

```
spanning-tree mode mstp
!

!
end
switch(config)#
```

12.1.4.51 spanning-tree portfast

The **spanning-tree portfast** command programs configuration mode ports to immediately enter forwarding state when they establish a link. PortFast ports are included in spanning tree topology calculations and can enter discarding state. This command overrides the **spanning-tree portfast auto** command.

The **no spanning-tree portfast** and **default spanning-tree portfast** commands remove the corresponding **spanning-tree portfast** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree portfast

no spanning-tree portfast

default spanning-tree portfast

Example

This command unconditionally enables portfast on ***interface ethernet 5***.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast
switch(config-if-Et5)#
```

12.1.4.52 spanning-tree portfast auto

The **spanning-tree portfast auto** command enables auto-edge detection on the configuration mode interface. When auto-edge detection is enabled, the port is configured as an edge port if it does not receive a new BPDU before the current BPDU expires. Auto-edge detection is enabled by default. The **spanning-tree portfast** command, when configured, has priority over this command.

The **no spanning-tree portfast auto** command disables auto-edge port detection. This command is removed from ***running-config*** with the **spanning-tree portfast auto** and **default spanning-tree portfast auto** commands.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree portfast auto

no spanning-tree portfast auto

default spanning-tree portfast auto

Example

This command enables auto-edge detection on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree portfast auto
switch(config-if-Et5)#
```

12.1.4.53 spanning-tree portfast <port type>

The **spanning-tree portfast** command specifies the STP port mode for the configuration mode interface. Default port mode is **normal**.

Port modes include:

- **Edge**: Edge ports connect to hosts and transition to the forwarding state when the link is established. An edge port that receives a BPDU becomes a normal port.
- **Network**: Network ports connect only to switches or bridges and support bridge assurance. Network ports that connect to hosts or other edge devices transition to the discarding state.
- **Normal**: Normal ports function as normal STP ports and can connect to any type of device.

The **no spanning-tree portfast <port-type>** and **default spanning-tree portfast <port-type>** commands restore the default port mode of normal by removing the corresponding **spanning-tree portfast <port-type>** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree portfast PORT_MODE

no spanning-tree portfast PORT_MODE

default spanning-tree portfast PORT_MODE

Parameters

PORT_MODE STP port mode. Options include:

- **edge**
- **network**
- **normal**

  The **normal** option is not available for the **no** and **default** commands.

Related Commands

The **spanning-tree portfast** command also affects the **spanning-tree portfast auto** and **spanning-tree portfast** configuration for the configuration mode interface:

- **spanning-tree portfast normal**: **spanning-tree portfast auto** is enabled.
- **spanning-tree portfast edge**: **spanning-tree portfast** is enabled.
- **spanning-tree portfast network**: **spanning-tree portfast auto** is disabled.

Example

This command configures ***interface ethernet 5*** as a network port.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree portfast network
switch(config-if-Et5)#
```

12.1.4.54 spanning-tree port-priority

The **spanning-tree port-priority** command specifies the configuration mode interfaces port-priority number. The switch uses this number to determine which interface it places into forwarding mode when resolving a loop. Valid settings are all multiples of ***16*** between ***0*** and ***240***. Default value is ***128***. Ports with lower numerical priority values are selected over other ports.

The **no spanning-tree port-priority** and default spanning-tree port-priority commands restore the default of ***128*** for the configuration mode interface by removing the **spanning-tree port-priority** command from ***running-config***.

The **spanning-tree port-priority** command provides a mode option:

- RST instance port-priority is configured by not including a mode.
- MST instance ***0*** port-priority is configured by not including a mode or with the **mst** mode option.
- MST instance port-priority is configured with the **mst** mode option.
- Rapid-PVST VLAN port-priority is configured with the **vlan** mode option.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

spanning-tree [**MODE**] **port-priority** ***value***

no spanning-tree [**MODE**] **port-priority**

default spanning-tree [**MODE**] **port-priority**

Parameters

- **MODE** specifies the spanning tree instances for which the cost is configured. Values include:
  - ***no parameter*** RST instance or MST instance 0.
  - **mst** ***m_range*** specified MST instances. ***m_range*** formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from ***0*** to ***4094***.
  - **vlan** ***v_range*** specified Rapid-PVST instances. ***v_range*** formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from ***1*** to ***4094***.
- ***value*** bridge priority number. Values range from ***0*** to ***240*** and must be a multiple of ***16***.

Example

This command sets the port priority of ***interface ethernet 5*** to ***144***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# spanning-tree port-priority 144
switch(config-if-Et5)#
```

12.1.4.55 spanning-tree priority

The **spanning-tree priority** command configures the bridge priority number. The bridge priority is the four most significant digits of the bridge ID, which is used by spanning tree algorithms to select the root bridge and choose among redundant links. Bridge ID numbers range from ***0*** to ***65535*** (***16*** bits); bridges with smaller bridge IDs are elected over other bridges.

Because bridge priority sets the four most significant bits of the bridge ID, valid settings include all multiples of ***4096*** between ***0*** and ***61440***. Default value is ***32768***.

The **spanning-tree priority** command provides a mode option:

- RST instance priority is configured by not including a mode.

- MST instance ***0*** priority is configured by not including a mode or with the **mst** mode option.
- MST instance priority is configured with the **mst** mode option.
- Rapid-PVST VLAN priority is configured with the **vlan** mode option.

The **`no spanning-tree priority`** and **`default spanning-tree priority`** commands restore the bridge priority default of ***32768*** for the specified mode by removing the corresponding **`spanning-tree priority`** command from ***running-config***.

Another method of adding **`spanning-tree priority`** commands to the configuration is through the spanning-tree root command. Similarly, the **`no spanning-tree root`** command removes the corresponding **`spanning-tree priority`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`spanning-tree` [**MODE**] **`priority`** ***level***

`no spanning-tree` [**MODE**] **`priority`**

`default spanning-tree` [**MODE**] **`priority`**

Parameters

- **MODE** spanning tree instances for which the command configures priority. Options include:
  - ***no parameter*** RST instance or MST instance ***0***.
  - **mst** ***m_range*** specified MST instances. ***m_range*** formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from ***0*** to ***4094***.
  - **vlan** ***v_range*** specified Rapid-PVST instances. ***v_range*** formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from ***1*** to ***4094***.
- ***level*** priority number. Values include multiples of ***4096*** between ***0*** and ***61440***. Default is ***32768***.

Examples

- This command configures a bridge priority value of ***20480*** for Rapid-PVST VLANs ***20***, ***24***, ***28***, and ***32***.

```
switch(config)# spanning-tree vlan-id 20,24,28,32 priority 20480
switch(config)#
```

- This command configures a bridge priority value of ***36864*** for the RST instance. When MST is enabled, this command configures a priority of ***36864*** for MST instance ***0***.

```
switch(config)# spanning-tree priority 36864
switch(config)#
```

12.1.4.56 spanning-tree root

The **`spanning-tree root`** command configures the bridge priority number by adding a spanning-tree priority command to the configuration. Parameter settings set the following priority values:

- **primary** sets the bridge priority to ***8192***.
- **secondary** sets the bridge priority to ***16384***.

The bridge priority is the four most significant digits of the bridge ID, which is used by spanning tree algorithms to select the root bridge and choose among redundant links. Bridge ID numbers range from ***0*** to ***65535*** (***16*** bits); bridges with smaller bridge IDs are elected over other bridges.

When no other switch in the network is similarly configured, assigning the primary value to the switch facilitates its selection as the root switch. Assigning the secondary value to the switch facilitates its selection as the backup root in a network that contains one switch with a smaller priority number.

The **spanning-tree root** command provides a mode option:

- RST instance priority is configured by not including a mode.
- MST instance ***0*** priority is configured by not including a mode or with the **mst** mode option.
- MST instance priority is configured with the **mst** mode option.
- Rapid-PVST VLAN priority is configured with the **vlan** mode option.

The **no spanning-tree root** and **default spanning-tree root** commands restore the bridge priority default of ***32768*** by removing the corresponding **spanning-tree priority** command from ***running-config***. The **no spanning-tree root**, **no spanning-tree priority**, **default spanning-tree root** and **default spanning-tree priority** commands perform the same function.

Command Mode

Global Configuration

Command Syntax

spanning-tree [**MODE**] **root TYPE**

no spanning-tree [**MODE**] **root**

default spanning-tree [**MODE**] **root**

Parameters

- **MODE** specifies the spanning tree instances for which priority is configured. Values include:
  - ***no parameter*** RST instance or MST instance ***0***.
  - **mst** ***m_range*** specified MST instances. ***m_range*** formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from ***0*** to ***4094***.
  - **vlan** ***v_range*** specified Rapid-PVST instances. ***v_range*** formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from ***1*** to ***4094***.
- **TYPE** sets the bridge priority number. Values include:
  - **primary** sets the bridge priority to ***8192***.
  - **secondary** sets the bridge priority to ***16384***.

Examples

- This command configures a bridge priority value of ***8192*** for Rapid-PVST VLANs ***20*** to ***36***.

```
switch(config)# spanning-tree vlan-id 20-36 root primary
```

- This command configures a bridge priority value of ***16384*** for the RSTP instance and MST instance ***0***.

```
switch(config)# spanning-tree root secondary
```

12.1.4.57 spanning-tree transmit active

The **spanning-tree transmit active** command enables bridge assurance globally, which enables bridge assurance on all ports with a port type of **network**. Bridge assurance protects against unidirectional link failure, other software failure, and devices that quit running a spanning tree algorithm.

Bridge assurance is available only on point-to-point links on spanning tree **network** ports. Both ends of the link should ideally have bridge assurance enabled. Bridge assurance-enabled ports will not necessarily be blocked when they link to a port where bridge assurance is not enabled, but if they do not receive periodic BPDUs from the other side of the link the **show spanning-tree transmit active** command will show a bridge assurance status of inconsistent (blocking) for that port.

The **no spanning-tree transmit active** command disables bridge assurance.

The **spanning-tree transmit active** and **default spanning-tree transmit active** commands restore the default behavior by removing the **no spanning-tree transmit active** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

spanning-tree transmit active

no spanning-tree transmit active

default spanning-tree transmit active

Example

This command enables bridge assurance on the switch.

```
switch(config)# spanning-tree transmit active
switch(config)#
```

12.1.4.58 spanning-tree vlan-id

The **spanning-tree vlan-id** command enables Spanning Tree Protocol (STP) on specified VLANs by removing any corresponding **no spanning-tree vlan-id** statements from ***running-config***. Spanning-tree is enabled on all VLANs by default.

The **no spanning-tree vlan-id** command disables STP on the specified interfaces. The **default spanning-tree vlan-id** enables STP on the specified interfaces.

Note: Disabling STP is not recommended, even in topologies free of physical loops; STP guards against configuration mistakes and cabling errors. When disabling STP, ensure that there are no physical loops in the VLAN.

When disabling STP on a VLAN, ensure that all switches and bridges in the network disable STP for the same VLAN. Disabling STP on a subset of switches and bridges in a VLAN may have unexpected results because switches and bridges running STP will have incomplete information regarding the network's physical topology.

The following STP global configuration commands provide a **vlan** option for configuring Rapid-PVST VLAN instances:

- spanning-tree priority
- spanning-tree root

Command Mode

Global Configuration

Command Syntax

spanning-tree vlan-id *v_range*

no spanning-tree vlan-id *v_range*

default spanning-tree vlan-id *v_range*

Parameters

v_range VLAN list. Formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from ***1*** to ***4094***.

Examples

- This command disables STP on ***VLANs 200-205***.

```
switch(config)# no spanning-tree vlan-id 200-205
switch(config)#
```

- This command enables STP on ***vlan 203***.

```
switch(config)# spanning-tree vlan-id 203
switch(config)#
```

12.1.4.59 switchport backup-link

The **`switchport backup-link`** command establishes a primary-backup configuration for forwarding VLAN traffic between the command mode interface and a specified interface. The **`show interfaces switchport backup-link`** command displays the state of backup interface pairs on the switch:

- the primary interface is the command mode interface.
- the backup interface is the interface specified in the command.

The following guidelines apply to primary and backup interfaces:

- Ethernet and Port Channels can be primary interfaces.
- Ethernet, Port Channel, Management, Loopback, and VLANs can be backup interfaces.
- The primary and backup interfaces can be different interface types.
- Interface pairs should be similarly configured to ensure consistent behavior.
- An interface can be associated with a maximum of one backup interface.
- An interface can back up a maximum of one interface.
- Any Ethernet interface configured in an interface pair cannot be a port channel member.
- The STP mode is backup.
- Static MAC addresses should be configured after primary-backup pairs are established.

When load balancing is not enabled, the primary and backup interfaces cannot simultaneously forward VLAN traffic. When the primary interface is forwarding VLAN traffic, the backup interface drops all traffic. If the primary interface fails, the backup interface forwards VLAN traffic until the primary interface is functional.

The **prefer vlan** option balances the load across the primary and backup interfaces. When the command includes the **prefer vlan** option, each interface is the primary for a subset of the vlans carried by the pair. When both interfaces are up, prefer option vlans are forwarded on the backup interface and all other configured vlans are carried by the primary interface.

The **`no switchport backup-link`** and **`default switchport backup-link`** commands remove the primary-backup configuration for the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

`switchport backup-link` **INT_NAME** [**BALANCE**]

`no switchport backup-link`

`default switchport backup-link`

Parameters

- **INT_NAME** the backup interface. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.

- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Channel group interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.

- **BALANCE** VLANs whose traffic is normally handled on the backup interfaces. Values include:
  - ***no parameter*** backup interface handles no traffic if the primary interface is operating.
  - **prefer vlan *v_range*** list of VLANs whose traffic is handled by backup interface.

Examples

- These commands establish ***interface ethernet 7*** as the backup port for ***interface ethernet 1***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# switchport backup-link ethernet 7
switch(config-if-Et1)#
```

- These commands configure the following:
  - ***interface ethernet 1*** as a trunk port that handles VLAN ***4*** through ***9*** traffic.
  - ***interface ethernet 2*** as its backup interface.
  - ***interface ethernet 2*** as the preferred interface for VLANs ***7*** through ***9***.

```
switch(config-if-Et1)# switchport mode trunk
switch(config-if-Et1)# switchport trunk allowed vlan 4-9
switch(config-if-Et1)# switchport backup-link Ethernet 2 prefer vlan 7-9
switch(config-if-Et1)#
```

12.1.4.60 transmit-interval

The **`transmit-interval`** command sets the interval at which loop detection packets are transmitted. The **`no transmit-interval`** and **`default transmit-interval`**commands restore the transmission interval to the default of ***5*** seconds.

Command Mode

Loop-protection Configuration

Command Syntax

`transmit-interval` ***interval***

Parameters

interval Interval in seconds at which loop-detection packets are transmitted. Values range from ***1*** to ***10***; default is ***5***.

Example

This command sets the loop detection packet transmission interval to ***10*** seconds.

```
switch(config-monitor-loop-protect)# transmit-interval 10
switch(config-monitor-loop-protect)#
```

12.2 Link Layer Discovery Protocol

This section describes Link Layer Discovery Protocol (LLDP) configuration tasks. Refer to the command descriptions for information about commands used in this chapter.

Topics in this section include:

- LLDP Introduction
- LLDP Overview
- LLDP Configuration Procedures
- LLDP Configuration Commands

12.2.1 LLDP Introduction

Link Layer Discovery Protocol (LLDP) lets Ethernet network devices to advertise details about themselves, such as capabilities, identification, and device configurations to directly connected devices on the network that are also using LLDP.

12.2.2 LLDP Overview

LLDP is a discovery protocol that allows devices to advertise information about themselves to peer devices that are on the same physical LAN and store information about the network. LLDP allows a device to learn higher layer management reachability and connection endpoint information from adjacent devices.

Each switch with an active LLDP agent sends and receives messages on all physical interfaces enabled for LLDP transmission. These messages are sent periodically and are typically configured for short time intervals to ensure that accurate information is always available. These messages are then stored for a configurable period of time, and contained within the received packet. The message information expires and is discarded when the configured value is met. The only other time an advertisement is sent is when a relevant change takes place in the switch. If information changes for any reason, the LLDP agent is notified and sends out and update the new values.

12.2.2.1 LLDP Data Units

A single LLDP Data Unit (LLDPDU) is transmitted in a single 802.3 Ethernet frame. The basic LLDPDU includes a header and a series of Type-Length-Value elements (TLVs). Each TLV advertises different types of information, such as its device ID, type, or management addresses.

LLDP advertises the following TLVs by default:

- port-description
- system-capabilities
- system-description
- system-name
- management-address
- port-vlan

12.2.2.2 Transmission and Reception

Every device that uses LLDP has its own LLDP agent. The LLDP agent is responsible for the reception, transmission, and management of LLDP. When LLDP is enabled on a port, transmission and reception of LLDPDUs are both enabled by default, but the agent can be configured to only transmit or only receive.

Transmission

When LLDP transmission is enabled, the LLDP agent advertises information about the switch to neighbors at regular intervals. Each transmitted LLDPDU contains the mandatory TLVs, and any enabled optional TLVs.

Reception

When LLDP reception is enabled, the LLDP agent receives and stores advertised information from neighboring devices.

12.2.2.3 Storing LLDP Information

Whenever the switch receives a valid and current LLDP advertisement from a neighbor, it stores the information in a Simple Network Management Protocol (SNMP) Management Information Base (MIB).

12.2.2.4 Guidelines and Limitations

LLDP has the following configuration limitations:

- LLDP must be enabled globally before it can be enabled on an interface.
- LLDP is not supported on virtual interfaces.
- LLDP can discover only one device per port.

12.2.3 LLDP Configuration Procedures

These sections describe the following configuration processes:

- Enabling LLDP Globally
- Enabling LLDP on an Interface
- Optional LLDP Parameters
- Clearing LLDP Statistics
- Displaying LLDP Information

12.2.3.1 Enabling LLDP Globally

The lldp run command globally enables LLDP on the Arista switch. Once LLDP is enabled, the switch will transmit advertisements from the ports that are configured to send TLVs. The neighbor information table is populated as advertisements from the neighbors arrive on the ports.

Example

This command enables LLDP globally on the Arista switch.

```
switch(config)# lldp run
switch(config)#
```

12.2.3.2 Enabling LLDP on an Interface

When enabling LLDP, it is enabled on all interfaces by default. By using the lldp transmit and lldp receive commands, LLDP can be enabled or disabled on individual interfaces or configured to only send or only receive LLDP packets.

Example

- These commands enable ***interface ethernet port 3/1*** to transmit LLDP packets.

```
switch(config)# interface ethernet 3/1
switch(config-if-Et3/1)# lldp transmit
switch(config-if-Et3/1)#
```

- These commands enable ***interface ethernet port 3/1*** to receive LLDP packets.

```
switch(config)# interface ethernet 3/1
switch(config-if-Et3/1)# lldp receive
switch(config-if-Et3/1)#
```

12.2.3.3 Optional LLDP Parameters

The following sections describe these tasks:

- Setting the LLDP Timer
- Setting the LLDP Hold Time
- Setting the LLDP Re-initialization Timer
- Setting the IP Management Address to be used in the TLV
- Selecting the LLDP TLVs
- Configuring LLDP for Power over Ethernet

12.2.3.3.1 Setting the LLDP Timer

The lldp timer command specifies the time in seconds between LLDP updates sent by the switch.

Examples

- This command specifies that the LLDP updates should be sent every ***120*** seconds.

```
switch(config)# lldp timer 120
switch(config)#
```

- This command reverts the LLDP timer to its default value of ***30*** seconds.

```
switch(config)# no lldp timer 120
switch(config)#
```

12.2.3.3.2 Setting the LLDP Hold Time

The lldp hold-time command sets the amount of time a receiving device should retain the information sent by the device.

Examples

- This command specifies that the receiving device should retain the information for ***180*** seconds before discarding it.

```
switch(config)# lldp hold-time 180
switch(config)#
```

- This command reverts the LLDP hold time and to the default value of ***120*** seconds.

```
switch(config)# no lldp hold-time 180
switch(config)#
```

12.2.3.3.3 Setting the LLDP Re-initialization Timer

The lldp run command specifies the amount in time in seconds to delay the re-initialization attempt by the switch.

Example

This command specifies that the switch waits ***10*** seconds before attempting to re-initialize.

```
switch(config)# lldp timer reinitialization 10
switch(config)#
```

12.2.3.3.4 Setting the IP Management Address to be used in the TLV

The lldp management-address command specifies the IP management address or the IP address of the VRF interface in LLDP Type-Length-Value (TLV) triplets.

Example

This command specifies the IP management address to be used in the TLV.

```
switch(config)# lldp management-address ethernet 3/1
switch(config)#
```

12.2.3.3.5 Selecting the LLDP TLVs

The lldp tlv transmit command specifies which Type, Length, and Value elements (TLVs) are to be included in LLDP packets. The **no lldp tlv transmit** command removes the TLV configuration.

Example

This command enables the system descriptions to be included in the TLVs.

```
switch(config)# lldp tlv transmit system-description
switch(config)#
```

12.2.3.3.6 Configuring LLDP for Power over Ethernet

Initial Power over Ethernet (PoE) power-level negotiation with a Powered Device (PD) takes place in hardware (see Configuring Power over Ethernet (PoE)). Once hardware negotiation has taken place, IEEE 802.3at Power Via MDI Type-Length-Value elements (TLVs) are included by default in LLDP packets sent to connected PDs to allow LLDP to further negotiate power needs. LLDP allows the switch to deal with more granular power requests from PDs, and also allows dynamic power-level setting. TLVs received from connected Power-Sourcing Equipment (PSE) are ignored.

Note: Power Via MDI TLVs are not sent (even when enabled) under the following circumstances:

1. there is a user-configured power limit on the port, or
2. hardware negotiation sets the power to higher than class ***4*** because ***IEEE 802.3bt***, which increases the maximum power output for PoE, is not yet supported by LLDP.

To disable Power Via MDI TLVs globally, use the no lldp tlv transmit command and specify the Power Via MDI TLV. Hardware negotiation and manual power limits will remain in effect.

Example

This command disables the sending of Power Via MDI TLVs globally.

```
switch(config)# no lldp tlv transmit power-via-mdi
switch(config)#
```

To disable Power Via MDI TLVs on an individual interface, use the poe negotiation lldp disabled command. Hardware negotiation and manual power limits will remain in effect.

Example

These commands disable the sending of Power Via MDI TLVs on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# poe negotiation lldp disabled
switch(config-if-Et5)#
```

New LLDP Fields Defined by IEEE 802.3at-2009

Arista switches *do not* support the following new LLDP/SNMP fields defined in ***IEEE standard 802.3at-2009***:

- **Power type** a LldpXdot3RemPowerType
- **Power source** a LldpXdot3RemPowerSource
- **Power priority** a LldpXdot3RemPowerPriority

- **PD requested power value** a LldpXdot3RemPDRequestedPowerValue
- **PSE allocated power value** a LldpXdot3RemPSEAllocatedPowerValue

12.2.3.4 Clearing LLDP Statistics

- clear lldp counters
- clear lldp table

12.2.3.5 Displaying LLDP Information

- Viewing LLDP Global Information
- Viewing LLDP Local Information
- Viewing LLDP Neighbors
- Viewing LLDP Traffic

12.2.3.5.1 Viewing LLDP Global Information

The show lldp command displays LLDP information.

Examples

- This command displays global information about LLDP.

```
switch# show lldp
LLDP transmit interval      : 60 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : default
Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities
  Management Address (Management0)
  IEEE802.1 Port VLAN ID
  IEEE802.3 Link Aggregation
  IEEE802.3 Maximum Frame Size
Port       Tx Enabled  Rx Enabled
Et3/1      Yes         Yes switch#
```

- This command displays LLDP information.

```
switch# show lldp ethernet interface 3/1
LLDP transmit interval      : 30 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : default
Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities switch#
```

12.2.3.5.2 Viewing LLDP Local Information

The show lldp local-info command displays the information contained in the LLDP TLVs to be sent about the local system.

Example

This command displays information contained in the TLVS about the local systems.

```
switch# show lldp local-info management 1
Local System:
  - Chassis ID type: MAC address (4)
    Chassis ID     : 001c.730f.11a8
  - System Name: "switch.aristanetworks.com"
  - System Description: "Arista Networks EOS version 4.13.2F running on an
 Arista
Networks DCS-7150S-64-CL"
  - System Capabilities : Bridge, Router
    Enabled Capabilities: Bridge

Interface Management1:
  - Port ID type: Interface name (5)
    Port ID     : "Management1"
  - Port Description: ""
  - Management Address Subtype: IPv4 (1)
    Management Address         : 172.22.30.154
    Interface Number Subtype   : ifIndex (2)
    Interface Number           : 999001
    OID String                 :
  - IEEE802.1 Port VLAN ID: 0
  - IEEE802.1/IEEE802.3 Link Aggregation
    Link Aggregation Status: Not Capable (0x00)
    Port ID                : 0
  - IEEE802.3 Maximum Frame Size: 1518 bytes
switch(config)#
```

12.2.3.5.3 Viewing LLDP Neighbors

The show lldp neighbors command displays information about LLDP neighbors.

Examples

- This command shows information about LLDP neighbors.

```
switch# show lldp neighbor
Last table change time   : 0:12:33 ago
Number of table inserts  : 33
Number of table deletes  : 0
Number of table drops    : 0
Number of table age-outs : 0

Port      Neighbor Device ID              Neighbor Port ID           TTL
Et3/1     tg104.sjc.aristanetworks.com    Ethernet3/2                120

Ma1/1     dc1-rack11-tor1.sjc             1/1                        120
 switch#
```

- This command displays detailed information about the neighbor ***ethernet 3/1***.

```
switch# show lldp neighbor ethernet 3/1
Last table change time   : 0:16:24 ago
Number of table inserts  : 33
Number of table deletes  : 0
Number of table drops    : 0
Number of table age-outs : 0

Port      Neighbor Device ID              Neighbor Port ID           TTL
Et3/1     tg104.sjc.aristanetworks.com    Ethernet3/2                120
 switch#
```

12.2.3.5.4 Viewing LLDP Traffic

The show lldp counters command displays the LLDP traffic information for the switch.

Example

This command displays the LLDP counters on the switch.

```
switch# show lldp counters
Port          Tx Frames Tx Length Exceeded

Et20              69485                  0
Et21              69394                  0
Et22              69203                  0
Et23              57546                  0
Et24                  0                  0
Ma1               69665                  0
Port      Rx Frames     Rx Errors    Rx Discard  TLVs Discard  TLVs Unknown

Et20          69470             0             0             0             0
Et21          69383             0             0             0             0
Et22          69143             0             0             0             0
Et23          55370             0             0             0             0
Et24              0             0             0             0             0
Ma1           69078         69078             0         69078             0 switch#
```

12.2.4 LLDP Configuration Commands

Global Configuration Commands

- lldp hold-time
- lldp management-address
- lldp management-address vrf
- lldp receive packet tagged drop
- lldp run
- lldp timer
- lldp timer reinitialization
- lldp tlv transmit

Interface Configuration Commands – Ethernet Interface

- lldp receive
- lldp transmit
- poe negotiation lldp disabled

Privileged EXEC Commands

- clear lldp counters
- clear lldp table

EXEC Commands

- show lldp
- show lldp counters
- show lldp local-info
- show lldp neighbors

12.2.4.1 clear lldp counters

The **`clear lldp counters`** command resets the LLDP counters to zero.

Command Mode

Privileged EXEC

Command Syntax

`clear lldp counters` [**SCOPE**]

Parameters

SCOPE Session affected by command. Options include:

- ***no parameter*** command affects counters on all CLI sessions.
- **session** clears LLDP counters for the current CLI session only.

Examples

- This command resets all the LLDP counters to zero.

```
switch(config)# clear lldp counters
switch(config)#
```

- This command resets only the LLDP counters for the current CLI session.

```
switch(config)# clear lldp counters session
switch(config)#
```

12.2.4.2 clear lldp table

The **`clear lldp table`** command clears neighbor information from the LLDP table.

Command Mode

Privileged EXEC

Command Syntax

`clear lldp table`

Example

This command clears neighbor information from the LLDP table.

```
switch(config)# clear lldp table
switch(config)#
```

12.2.4.3 lldp hold-time

The **`lldp hold-time`** command specifies the amount of time a receiving device should maintain the information sent by the device before discarding it.

Command Mode

Global Configuration

Command Syntax

`lldp hold-time` ***period***

`no lldp hold-time`

`default lldp hold-time`

Parameters

period The amount of time a receiving device should hold LLDPDU information before discarding it. Value ranges from ***10*** to ***65535*** second; default value is ***120*** seconds.

Examples

- This command sets the amount of time before the receiving device discards LLDPDU information to ***180*** seconds.

```
switch(config)# lldp hold-time 180
switch(config)#
```

- This command restores the hold-time to its default value of ***120*** seconds.

```
switch(config)# no lldp hold-time 180
switch(config)#
```

12.2.4.4 lldp management-address

The **lldp management-address** command enables the user to add the IP management address used for LLDP Type-Length-Value (TLV).

Command Mode

Global Configuration

Command Syntax

lldp management-address [**INTERFACE**]

no lldp management-address [**INTERFACE**]

default lldp management-address [**INTERFACE**]

Parameters

INTERFACE Interface type and number. Options include:

- **all** all interfaces.
- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.

Examples

- This command specifies the IP management address to be used in the TLV.

```
switch(config)# lldp management-address ethernet 3/1
switch(config)#
```

- This command removes the IP management address used in the TLV.

```
switch(config)# no lldp management-address ethernet 3/1
switch(config)#
```

- This command specifies that ***vlan200*** is used in the TLV.

```
switch(config)# lldp management-address vlan 200
switch(config)#
```

- This command removes the VLAN ID used in the TLV.

```
switch(config)# no lldp management-address vlan 200
switch(config)#
```

12.2.4.5 lldp management-address vrf

The **lldp management-address vrf** command enables the user to add the IP address of the VRF interface used in LLDP Type-Length-Value (TLV).

Command Mode

Global Configuration

Command Syntax

lldp management-address vrf VRF_INSTANCE

no lldp management-address vrf VRF_INSTANCE

default lldp management-address vrf VRF_INSTANCE

Parameters

VRF_INSTANCE specifies the VRF instance.

Examples

- This command specifies the management address VRF to be used in the TLV.

```
switch(config)# lldp management-address vrf test 1
switch(config)#
```

- This command removes the management VRF used in the TLV.

```
switch(config)# no lldp management-address vrf test 1
switch(config)#
```

12.2.4.6 lldp receive

The **lldp receive** command enables LLDP packets on an interface. The **no lldp receive** command disables the acceptance of LLDP packets.

Command Mode

Interface-Ethernet configuration

Interface-Management configuration

Command Syntax

lldp receive

no lldp receive

default lldp receive

Examples

- These commands enable the reception of LLDP packets on ***interface ethernet 4/1***.

```
switch(config)# interface ethernet 4/1
switch(config-if-Et4/1)# lldp receive
switch(config-if-Et4/1)#
```

- These commands disable LLDP the reception of LLDP packets on ***interface ethernet 4/1***.

```
switch(config)# interface ethernet 4/1
switch(config-if-Et4/1)# no lldp receive
switch(config-if-Et4/1)#
```

12.2.4.7 lldp receive packet tagged drop

The **`lldp receive packet tagged drop`** command is a global configuration command and when configured the LLDP ignores all the packets with **VLAN-tag**. By default, this command is disabled.

Command Mode

Global Configuration

Command Syntax

`lldp receive packet tagged drop`

Example

This command when configured, the LLDP ignores all the packets with **VLAN-tag**.

```
switch(config)# lldp receive packet tagged drop
```

12.2.4.8 lldp run

The **`lldp run`** command enables LLDP on the switch.

Command Mode

Global Configuration

Command Syntax

`lldp run`

`no lldp run`

`default lldp run`

Examples

- This command enables LLDP globally on the switch.

```
switch(config)# lldp run
switch(config)#
```

- This command disables LLDP globally on the switch.

```
switch(config)# no lldp run
switch(config)#
```

12.2.4.9 lldp timer reinitialization

The **`lldp timer reinitialization`** command sets the time delay in seconds for LLDP to initialize.

Command Mode

Global Configuration

Command Syntax

`lldp timer reinitialization` ***delay***

`no lldp timer reinitialization`

`default lldp timer reinitialization`

Parameters

delay the amount of time the device should wait before re-initialization is attempted. Value ranges from ***1*** to ***20*** seconds; default value is ***2*** seconds.

Examples

- This command specifies that the switch should wait ***10*** seconds before attempting to re-initialize.

```
switch(config)# lldp timer reinitialization 10
switch(config)#
```

- This command restores the default initialization delay of ***2*** seconds.

```
switch(config)# no lldp timer reinitialization 10
switch(config)#
```

12.2.4.10 lldp timer

The **`lldp timer`** command specifies the amount of time a receiving device should maintain the information sent by the device before discarding it. The **`no lldp timer`** command removes the configured LLDP timer.

Command Mode

Global Configuration

Command Syntax

`lldp timer` ***transmission_time***

`no lldp timer`

`default lldp timer`

Parameters

transmission_time the period of time at which LLDPDUs are transmitted. Values range from ***5*** to ***32768*** seconds; the default is ***30*** seconds.

Examples

- This command configures a period of ***180*** seconds at which the LLDPDUs are transmitted.

```
switch(config)# lldp timer 180
switch(config)#
```

- This command removes the configured period of time at which the LLDPDUs are transmitted.

```
switch(config)# no lldp timer 180
switch(config)#
```

12.2.4.11 lldp tlv transmit

The **`lldp tlv transmit`** command allows the user to specify the Type-Length-Values (TLVs) to include in LLDP packets.

Command Mode

Global Configuration

Command Syntax

`lldp tlv transmit` **TLV_NAME**

`no lldp tlv transmit` **TLV_NAM**

`default lldp tlv transmit` **TLV_NAME**

Parameters

TLV_NAME Options include:

- **link-aggregation** specifies the link aggregation TLV.
- **management-address** specifies the management address TLV.
- **max-frame-size** specifies the Frame size TLV.
- **port-description** specifies the port description TLV.
- **port-vlan** specifies the port VLAN ID TLV.
- **power-via-mdi** specifies the power over Ethernet TLV.
- **system-capabilities** specifies the system capabilities TLV.
- **system-description** specifies the system description TLV.
- **system-name** specifies the system name TLV.

Examples

- This command enables the system description TLV:

```
switch(config)# lldp tlv transmit system-description
switch(config)#
```

- This command disables the system description TLV:

```
switch(config)# no lldp tlv transmit system-description
switch(config)#
```

- This command enables the max-frame-size TLV:

```
switch(config)# lldp tlv transmit max-frame-size
switch(config)#
```

- This command disables the max-frame-size TLV:

```
switch(config)# no lldp tlv transmit max-frame-size
switch(config)#
```

12.2.4.12 lldp transmit

The **lldp transmit** command enables the transit of LLDP packets on an interface.

Command Mode

Interface-Ethernet configuration

Interface-Management configuration

Command Syntax

lldp transmit

no lldp transmit

default lldp transmit

Examples

- These commands enable the transmission of LLDP packets.

```
switch(config)# interface ethernet 4/1
switch(config-if-Et4/1)# lldp transmit
switch(config-if-Et4/1)#
```

- These commands disable the transmission of LLDP packets.

```
switch(config)# interface ethernet 4/1
switch(config-if-Et4/1)# no lldp transmit
```

```
switch(config-if-Et4/1)#
```

12.2.4.13 poe negotiation lldp disabled

Power Via MDI TLVs are included by default in LLDP packets sent to Power over Ethernet (PoE) Powered Devices (PDs) to allow dynamic negotiation of power levels. The **poe negotiation lldp disabled** command disables the sending of Power Via MDI TLVs from the configuration-mode interface.

The **no poe negotiation lldp disabled** and **default poe negotiation lldp disabled** commands restore the default behavior (sending Power Via MDI TLVs) by removing the corresponding **poe negotiation lldp disabled** command from ***running-config***.

To disable Power Via MDI TLVs globally, use the no lldp tlv transmit command and specify the Power Via MDI TLV.

Command Mode

Interface-Ethernet configuration

Command Syntax

poe negotiation lldp disabled

no poe negotiation lldp disabled

default poe negotiation lldp disabled

Example

These commands disable the sending of power via MDI TLVs on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# poe negotiation lldp disabled
switch(config-if-Et5)#
```

12.2.4.14 show lldp counters

The **show lldp counters** command displays LLDP traffic information for the switch.

Command Mode

EXEC

Command Syntax

show lldp counters [**INTERFACE**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.

  Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the LLDP counters on the switch.

```
switch# show lldp counters

Port          Tx Frames Tx Length Exceeded

Et20              69485                  0
```

```
Et21              69394                 0
Et22              69203                 0
Et23              57546                 0
Et24                  0                 0
Ma1               69665                 0

Port          Rx Frames     Rx Errors    Rx Discard  TLVs Discard  TLVs Unknown

Et20              69470             0             0             0             0
Et21              69383             0             0             0             0
Et22              69143             0             0             0             0
Et23              55370             0             0             0             0
Et24                  0             0             0             0             0
Ma1               69078         69078             0         69078             0
```

12.2.4.15 show lldp local-info

The **`show lldp local-info`** command displays LLDP errors and overflows.

Command Mode

EXEC

Command Syntax

`show lldp local-info` [**INTERFACE**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.

  Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the specific LLDP errors and overflows on ***management interface 1***.

```
switch# show lldp local-info management 1
Local System:
  - Chassis ID type: MAC address (4)
    Chassis ID     : 001c.730f.11a8qqq
  - System Name: "switch.aristanetworks.com"
  - System Description: "Arista Networks EOS version 4.13.2F running on an
 Arista
Networks DCS-7150S-64-CL"
  - System Capabilities : Bridge, Router
    Enabled Capabilities: Bridge

Interface Management1:
  - Port ID type: Interface name (5)
    Port ID     : "Management1"
  - Port Description: ""
  - Management Address Subtype: IPv4 (1)
    Management Address        : 172.22.30.154
    Interface Number Subtype  : ifIndex (2)
    Interface Number          : 999001
    OID String                :
  - IEEE802.1 Port VLAN ID: 0
  - IEEE802.1/IEEE802.3 Link Aggregation
    Link Aggregation Status: Not Capable (0x00)
    Port ID                : 0
  - IEEE802.3 Maximum Frame Size: 1518 bytes
```

```
se505.16:01:44#
switch#
```

12.2.4.16 show lldp neighbors

The **show llpd neighbors** command displays information about the switch's LLDP neighbors.

Command Mode

EXEC

Command Syntax

show lldp neighbors [**INTERFACE**][**INFO_LEVEL**]

Parameters

- **INTERFACE** Interface type and numbers. Options include:
  - ***no parameter*** displays information for all interfaces.
  - **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
  - **management *m_range*** Management interface range specified by ***m_range***.
  - Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **INFO_LEVEL** amount of information that is displayed. Options include:
  - ***no parameter*** Displays information for all interfaces.
  - **detailed** LLDP information for all the adjacent LLDP devices.

Examples

- This command displays the neighbor's information about LLDP.

```
switch(config)# show lldp neighbors
Last table change time   : 0:12:33 ago
Number of table inserts  : 33
Number of table deletes  : 0
Number of table drops    : 0
Number of table age-outs : 0

Port      Neighbor Device ID             Neighbor Port ID      TTL
Et3/1     tg104.sjc.aristanetworks.com   Ethernet3/2           120

Ma1/1     dc1-rack11-tor1.sjc            1/1                   120 switch#
```

- This command displays LLDP neighbor information for ***interface ethernet 3/1***.

```
switch# show lldp neighbors ethernet 3/1
Last table change time   : 0:16:24 ago
Number of table inserts  : 33
Number of table deletes  : 0
Number of table drops    : 0
Number of table age-outs : 0

Port      Neighbor Device ID             Neighbor Port ID       TTL
Et3/1     tg104.sjc.aristanetworks.com   Ethernet3/2            120 switch#
```

- This command displays detailed LLDP neighbor information for ***interface ethernet 3/1***.

```
switch# show lldp neighbors 3/1 detail

Interface Ethernet 3/1 detected 1 LLDP neighbors:

  Neighbor 001c.7300.1506/Ethernet6/25, age 8 seconds
```

```
  Discovered 5 days, 3:58:58 ago; Last changed 5 days, 3:56:57 ago
    - Chassis ID type: MAC address (4)
      Chassis ID     : 001c.7300.1506
    - Port ID type: Interface name (5)
      Port ID     : "Ethernet6/25"
    - Time To Live: 120 seconds
 - Port Description: "Ethernet6/25"
 - IEEE802.3 Power Via MDI
    Port Class               : PD
    PSE MDI Power Support    : Not Supported
    PSE MDI Power State      : Disabled
    - System Name: "Leaf-Switch1.aristanetworks.com"
    - System Description: "Arista Networks EOS version 4.10.1-SSO running on
 an Arista Networks DCS-7504"
    - System Capabilities : Bridge, Router
      Enabled Capabilities: Bridge
    - Management Address Subtype: IPv4 (1)
      Management Address        : 172.22.30.116
      Interface Number Subtype  : ifIndex (2)
      Interface Number          : 999999
      OID String                :
    - IEEE802.1 Port VLAN ID: 1
    - IEEE802.1/IEEE802.3 Link Aggregation
      Link Aggregation Status: Capable, Disabled (0x01)
      Port ID                : 0
    - IEEE802.3 Maximum Frame Size: 9236 bytes
switch#
```

12.2.4.17 show lldp

The **show lldp** command displays LLDP information.

Command Mode

EXEC

Command Syntax

show lldp [**INTERFACE**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.

  Valid ***e_range*** and ***m_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays all LLDP information.

```
switch# show lldp
LLDP transmit interval      : 60 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : test

Enabled optional TLVs:
  Port Description
  System Name
  System Description
```

```
  System Capabilities
  Management Address (Management0)
  IEEE802.1 Port VLAN ID
  IEEE802.3 Link Aggregation
  IEEE802.3 Maximum Frame Size

Port       Tx Enabled  Rx Enabled
Et3/1      Yes         Yes

switch#
```

- This command displays specific information about LLDP for ***interface ethernet 3/1***.

```
switch# show lldp ethernet 3/1
LLDP transmit interval      : 30 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : default

Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities switch#
```

- This command displays specific information about LLDP for ***management interface 1/1***.

```
switch# show lldp management 1/1
LLDP transmit interval      : 60 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : default

Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities
  Management Address (Management0)
  IEEE802.1 Port VLAN ID
  IEEE802.3 Link Aggregation
  IEEE802.3 Maximum Frame Size

Port       Tx Enabled  Rx Enabled
Ma1/1      Yes         Yes switch#
```

12.3 Virtual LANs (VLANs)

This chapter describes Arista's Virtual LANs (VLANs) implementation and MAC address tables.

Sections in this chapter include:

- VLAN Introduction
- VLAN Conceptual Overview
- VLAN Configuration Procedures
- VLAN Configuration Commands

12.3.1 VLAN Introduction

Arista switches support industry standard 802.1q VLANs. Arista EOS provides tools to manage and extend VLANs throughout the data center network.

12.3.2 VLAN Conceptual Overview

12.3.2.1 VLAN Definition

A Virtual Local Area Network (VLAN) allows a group of devices to communicate as if they were in the same network regardless of their physical location. VLANs are Layer 2 structures based on the 802.1Q standard.

These parameters are associated with a VLAN:

- VLAN number (***1-4094***): VLAN numbers uniquely identify the VLAN within a network. `VLAN 1` exists by default; all other VLANs only exist after they are configured.
- VLAN name (optional): The VLAN name is a text string that describes the VLAN.
- VLAN state (***active*** or ***suspended***): The state specifies the VLAN transmission status within the switch. In the ***suspended*** state, VLAN traffic is blocked on all switch ports. The default state is ***active***.

VLANs define Layer 2 broadcast domains in a Layer 2 network, in which each device can receive broadcast frames sent by any other within the domain. Switches accommodating multiple broadcast domains serve as multi-port bridges where each broadcast domain is a distinct virtual bridge. Traffic does not pass directly between different VLANs within a switch or between two switches.

12.3.2.2 VLAN Switching

Ethernet and port channel interfaces are configured as switched ports by default. Switched ports are configurable as members of one or more VLANs. Switched ports ignore all IP-level configuration commands, including IP address assignments.

12.3.2.2.1 VLAN Trunking and Trunk Groups

Trunking extends multiple VLANs beyond the switch through a common interface or port channel.

A trunk group is the set of physical interfaces that comprise the trunk and the collection of VLANs whose traffic is carried on the trunk. The traffic of a VLAN that belongs to one or more trunk groups is carried only on ports that are members of trunk groups to which the VLAN belongs, i.e., VLANs configured in a trunk group are pruned of all ports that are not associated with the trunk group. See the Trunk Ports example section for further details.



Note: Be cautious when using allowed VLAN lists or trunk groups to ensure that the VLAN topology is consistent with any Layer-2 control protocol topology, or unpredictable results can occur.

VLAN traffic is carried through Ethernet or LAG ports. A port's switchport mode defines the number of VLANs for which the port can carry traffic.

- Access ports carry traffic for one VLAN – the access VLAN. Access ports associate untagged frames with the access VLAN. Access ports drop tagged frames that are not tagged with the access VLAN.
- Trunk ports carry traffic for multiple VLANs. Tag frames specify the VLAN for which trunk ports process packets.

12.3.2.2.2 Q-in-Q Trunking

A Q-in-Q network is a multi-tier layer 2 VLAN network. A typical Q-in-Q network is composed of a service provider network (tier 1) where each node connects to a customer network (tier 2).

802.1ad is a networking standard that supports Q-in-Q networks by allowing multiple 802.1Q tags in an Ethernet frame.

Each interface in a customer network is assigned to a customer-VLAN (c-VLAN). Packets in c-VLANs contain 802.1q tags that switch traffic within the network. c-VLANs access the service provider VLAN (s-VLAN) through a provider switch. Customer switch ports connect to an s-VLAN through provider switch edge ports, which are configured as dot1q ports and operate as follows:

- **Inbound traffic (from customer switches)**: adds an s-VLAN tag, then forwards packets to the provider network.
- **Outbound traffic (to customer switches)**: removes the s-VLAN tag, then forwards packets to the customer network.

12.3.2.2.3 TPID (Configurable Ethertypes)

By default, VLAN-tagged packets carry a Tag Protocol Identifier (TPID) of 0x8100. On some Arista platforms, however, the TPID of a switchport can be modified in accordance with IEEE 802.1ad to allow for the use of 802.1q TPIDs other than 0x8100. Well known and standard tags include:

0x8100 customer VLAN.

- **0x88a8** service VLAN tag used in provider bridging.
- **0x9100** service VLAN tag used in provider bridging (common, but not standardized).

Other non-standard TPID values may also be configured for interoperability with legacy equipment or non-standard systems. Values range from 0x600 (1536) through 0xFFFF (65535).

Non-default TPID values are most commonly used for provider bridging on a network-to-network interface.

12.3.2.3 VLAN Routing

Each VLAN can be associated with a Switch Virtual Interface (SVI), also called a VLAN interface. The VLAN interface functions in a routed network (Layer 3) with an assigned IP subnet address. Connecting different VLANs requires Layer 3 networking.

12.3.2.3.1 VLAN Interfaces

A Switched Virtual Interface (SVI) connects to the VLAN segment on the switch to provide Layer 3 processing for packets from the VLAN. An SVI can be activated only after it is connected to a VLAN. SVIs are typically configured for a VLAN to a default gateway for a subnet to facilitate traffic routing with other subnets.

In a Layer 3 network, each VLAN SVI is associated with an IP subnet, with all stations in the subnet members of the VLAN. Traffic between different VLANs is routed when IP routing is enabled.

12.3.2.3.2 Internal VLANs

A routed port is an Ethernet or port channel interface that functions as a Layer 3 interface. Routed ports do not bridge frames nor switch VLAN traffic. Routed ports have IP addresses assigned to them and packets are routed directly to and from the port.

The switch allocates an internal VLAN for an interface when it is configured as a routed port. The internal VLAN is assigned a previously unused VLAN ID. The switch prohibits the subsequent configuration of VLANs and VLAN interfaces with IDs corresponding to allocated internal VLANs.

12.3.2.3.3 Support for Private VLAN

Private VLAN is a feature that segregates a regular VLAN broadcast domain while maintaining all ports in the same IP subnet. There are three types of VLAN within a private VLAN:

1. **Primary VLAN**: Ports in the primary VLAN can send and or receive traffic from ports in all the corresponding PVLANs. There is only one primary VLAN in a private VLAN.
2. **Community VLAN**: This is a secondary VLAN. Hosts in a community VLAN forward traffic to each other as well as ports in the primary VLAN. There are multiple community VLANs in a private VLAN.

3. **Isolated VLAN**: This is a secondary VLAN. Hosts in an isolated VLAN only forward traffic to ports in the primary VLAN. Hosts within an isolated VLAN can not communicate with each other useing bridging. There are multiple isolated VLANs in a private VLAN.

12.3.2.3.3.1Limitations

On DCS-7280R, DCS-7280R2, DCS-7500R, DCS-7500R2, DCS-7020R

- Private VLAN and Algomatch features are mutually exclusive. Disable algomatch with the **`hardware access-list mechanism tcam`** command . Note that this requires a reload of the system to take effect.
- L2 and L3 multicast traffic is not supported.

On All Platforms except 7300X3, CCS-720XP, DCS-7050X3

Private VLAN and IPv4/IPv6 uRPF features are mutually exclusive.

On All Platforms

- Tunnel termination on PVLAN ports is not supported.
- Ingress IPv4/IPv6 RACLs on the primary VLAN are not honored for packets ingressing through ports in secondary VLANs.
- Only isolated private VLAN trunks and normal trunk ports are supported. It allows trunk ports to forward and receive traffic for all primary and or secondary VLANs. An isolated trunk translates traffic coming in on a primary VLAN to the lowest valued secondary VLAN on the trunk port.
- Private VLAN is not supported on L2 subinterfaces.
- Hardware accelerated Sflow is not supported on Private VLAN ports.
- VLAN Mapping and or Translation is not supported with Private VLAN.

12.3.2.3.3.2Show Commands

- Use the **`show vlan private-vlan`** command to display the primary and secondary defined VLANs:

```
switch# show vlan private-vlan
Primary Secondary Type         Ports
------- --------- ----------- ------------------------
100      101      community    Et1, Et6
100      102      isolated     Et1, Et7, Et8
200      201      community    Et10, Et9
```

- Use the **`show vlan 100,101,102,200,201`** command to diplay which interfaces are member of which VLANs:

```
# show vlan 100,101,102,200,201
VLAN  Name          Status    Ports
----- ------------- --------- --------------------------------
100   VLAN0100      active    Et1, Et6+, Et7+, Et8+
101   VLAN0101      active    Et1+, Et6
102   VLAN0102      active    Et1+, Et7, Et8
200   VLAN0200      active    Et10
201   VLAN0201      active    Et10+, Et9

+ indicates a private VLAN promoted port
```

Promoted ports are displayed to indicate they are part of the same broadcast domain as the indicated VLAN. Interfaces in a primary VLAN are included in the display of all its associated secondary VLANs. Interfaces in secondary VLANs are included in the display of both its primary VLAN and its own domain.

On DCS-7280R, DCS-7280R2, DCS-7500R, DCS-7500R2, DCS-7020R

Use the **show platform sand pvlan interfaces** command to display the status of the interfaces to configure a private VLAN.

```
switch# show platform sand pvlan interfaces
Interface           Secondary      Primary     State
                         VLAN         VLAN
-----------------  ----------  ----------  ---------
Ethernet6                 101         100    enabled
Ethernet7                 102         100    enabled
Ethernet8                 102         100    enabled
Po1                       102         100    enabled
Ethernet9                 201         200     failed
Po2                       202         200     failed
```

In this output, the **Secondary VLAN** column indicates the VLAN which is configured on the interface. The **Primary VLAN** column indicates the primary VLAN to which the secondary VLAN belongs to. The **State** field has three possible values - enabled, failed, configured. The **enabled** state indicates that the private VLAN is configured and enabled on that interface. The **failed** state indicates that the private VLAN configuration has failed for that interface The **configured** state indicates that private VLAN is configured on that interface but has not taken effect. When port channels are configured in a private VLAN, it is enabled only if entries for all the member interfaces are successfully programmed in the hardware. If the hardware entries for any one of the member interfaces fails, the entries for other member interfaces are also removed from the hardware and the state is marked as failed.

12.3.2.3.4 VLAN Translation

VLAN translation allows you to map packets from one VLAN to another. This can be carried out only on packets having a dot1q header (tagged frames). The translation rewrites the Vlan ID field (VID) in dot1q headers on packets passing through a switched port without changing any other fields.

VLAN translation also supports the ability to translate packets with a dot1q header to the internal VLAN for a routed port. The VLAN in the incoming packets is mapped to the internal VLAN of the routed port and packets egressing the routed port are encapsulated with a dot1q header for the specified VLAN. For egress packets, no priority information is added to the dot1q header and the priority from the incoming encapsulation will be retained.

When configuring the VLAN translation mode, consider the following:

- VLAN translation is only supported for tagged packets.
- BPDUs from STP, LLDP and other protocols are not affected by this mapping.
- VLAN translation is not applicable for access ports.
- Untagged packets entering the switch on the trunk native VLAN are not mapped.
- TPID and VLAN priority does not get re-written during the translation.

12.3.3 VLAN Configuration Procedures

These sections describe basic VLAN configuration tasks.

- Creating and Configuring VLANs
- Configuring VLAN Switching
- Creating and Configuring VLAN Interfaces
- Allocating Internal VLANs
- Private VLAN Configuration
- Configuring VLAN Translation

12.3.3.1 Creating and Configuring VLANs

The CLI provides two methods of creating VLANs.

- Explicitly through the vlan command.
- Implicitly through the switchport access vlan command.

The switchport access vlan command generates a warning message when it creates a VLAN.

To create a VLAN, use the **vlan** command in global configuration mode. Valid VLAN numbers range between ***1*** and ***4094***. To create multiple VLANs, specify a range of VLAN numbers.

To edit an existing VLAN, enter the **vlan** command with the number of the existing VLAN.

Examples

- This command creates ***VLAN 45*** and enters VLAN configuration mode for the new VLAN.

```
switch(config)# vlan 45
switch(config-vlan-45)#
```

- Use the name (VLAN configuration mode) command to assign a name to a VLAN.

  These commands assign the name Marketing to ***VLAN 45***.

```
switch(config)# vlan 45
switch(config-vlan-45)# name Marketing
switch(config-vlan-45)# show vlan 45

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
45   Marketing                        active    Et1

switch(config-vlan-45)#
```

- To change a VLAN's state, use the state command in VLAN configuration mode.

  These commands suspend ***VLAN 45***. VLAN traffic is blocked on all switch ports.

```
switch(config)# vlan 45
switch(config-vlan-45)# state suspend
switch(config-vlan-45)# show vlan 45

VLAN Name                             Status    Ports
---- -------------------------------- --------- ------
45   Marketing                        suspended

switch(config-vlan-45)#
```

- These commands activate ***VLAN 45***.

```
switch(config)# vlan 45
switch(config-vlan-45)# state active
switch(config-vlan-45)# show vlan 45

VLAN Name                             Status    Ports
---- -------------------------------- --------- ------
45   Marketing                        active    Et1

switch(config-vlan-45)#
```

12.3.3.1.1 VLAN Policy

The VLAN policy configuration command enables a switch to configure a VLAN policy when it receives a packet with unknown destination MAC address on a VLAN. The mac address forwarding command provides three options to configure a VLAN policy:

- Flood the Layer 2 miss packets on the VLAN
- Drop the Layer 2 miss packets
- Log the Layer 2 miss packets to the CPU (while still flooding them on the VLAN)

The default behavior is to flood the L2 miss packets on all ports of the VLAN.

VLAN policy configuration is supported on the Arista 7010, 7050 (excluding 7050SX3-48YC12, 7050CX3-32S, 7050QX2-32S, 7050SX2-72Q, 7050SX2-128, 7050TX2-128), 7060, 7250, and the 7300 series platforms.

VLAN policy is not supported in the following cases:

- STP, LLDP, and LACP packets
- VLAN policy configurations on VXLAN-enabled VLAN
- On a VLAN if IGMP snooping is configured with Multicast miss action is set to drop, then all multicast packets received on that VLAN are dropped.

Examples

- These commands create a ***vlan 333*** and then set the unicast policy to 'drop' and the multicast policy to 'log' for the specific ***vlan 333***.

```
switch(config)# vlan 333
switch(config-vlan-333)# mac address forwarding unicast miss action drop
switch(config-vlan-333)# mac address forwarding multicast miss action log
```

- These commands display the VLAN policy that was defined when ***vlan 333*** is created.

```
switch(config)# show vlan 333 mac address forwarding

VLAN  UcMissAction  McMissAction
----  ------------  ------------
 333  flood         flood
```

- These commands display the VLAN policy type that was defined when ***vlan 333*** is configured with the 'drop' unicast policy and the 'log' multicast policy.

```
switch(config)# show vlan 333 mac address forwarding

VLAN  UcMissAction  McMissAction
----  ------------  ------------
 333  drop          log

switch(config)# show vlan mac address forwarding

VLAN  UcMissAction  McMissAction
----  ------------  ------------
   1  flood         flood
 333  drop          log
```

12.3.3.2 Configuring VLAN Switching

The following describe the configuration of VLAN ports.

12.3.3.2.1 Access Ports

Access ports carry traffic for one VLAN, as designated by a switchport access vlan command. Access ports associate untagged frames with the access VLAN. Tagged frames received by the interface are dropped unless they are tagged with the access VLAN.

To configure an interface group as an access port, use the switchport mode command.

Examples

- These commands configure ***interface ethernet 1*** as an access port.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# switchport mode access
switch(config-if-Et1)#
```

- To specify the port's access VLAN, use the **`switchport access vlan`** command.

  These commands configure ***vlan 15*** as the access VLAN for ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport access vlan 15
switch(config-if-Et5)#
```

- These commands configure interface Ethernet ***1*** through ***3*** as access ports that process untagged frames as ***vlan 5*** traffic.

```
switch(config)# interface Ethernet 1-3
switch(config-if-Et1-3)# switchport mode access
switch(config-if-Et1-3)# switchport access vlan 5
switch(config-if-Et1-3)# show interfaces ethernet 1-3 vlans
Port       Untagged Tagged
Et1        None     23,25
Et2        18       -
Et3        None     14
switch(config-if-Et1-3)#
```

12.3.3.2.2 Trunk Ports

Trunk ports carry traffic for multiple VLANs. Messages use tagged frames to specify the VLAN for which trunk ports process traffic.

- The **`vlan trunk list`** specifies the VLANs for which the port handles tagged frames. The port drops any packets tagged for VLANs not in the VLAN list.
- The **`native vlan`** is the VLAN where the port switches untagged frames.

To configure an interface group as a trunk port, use the switchport mode command.

Example

These commands configure ***interface ethernet 8*** as a trunk port.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# switchport mode trunk
switch(config-if-Et8)#
```

By default all VLANs are permitted on a port configured with 'switchport mode trunk'. To limit the port's VLAN trunk list, use the **`switchport trunk allowed vlan`** command. Only VLANs in the allowed list will be permitted.

Examples

- These commands configure VLAN ***15***, ***20***, ***21***, ***22***, ***40***, and ***75*** as the explicitly permitted VLAN trunk list for ethernet interface ***12***-***16***.

```
switch(config)# interface ethernet 12-16
switch(config-if-Et12-16)# switchport trunk allowed vlan 15,20-22,40,75
switch(config-if-Et12-16)#
```

- These commands explicitly permit VLAN ***100*** through ***120*** to the VLAN trunk list for ***interface ethernet 14***.

```
switch(config)# interface ethernet 14
switch(config-if-Et14)# switchport trunk allowed vlan add 100-120
switch(config-if-Et14)#
```

- To specify the port's native VLAN, use the **`switchport trunk native vlan`** command.

  These commands configure ***vlan 12*** as the native VLAN trunk for ***interface ethernet 10***.

```
switch(config)# interface ethernet 10
switch(config-if-Et10)# switchport trunk native vlan 12
switch(config-if-Et10)#
```

- By default, ports send native VLAN traffic with untagged frames. The **`switchport trunk native vlan`** command can also configure the port to send native VLAN traffic with tag frames.

  These commands configure ***interface ethernet 10*** to send native VLAN traffic as tagged.

```
switch(config)# interface ethernet 10
switch(config-if-Et10)# switchport trunk native vlan tag
switch(config-if-Et10)#
```

- These commands configure ***interface ethernet 12*** as a trunk with ***vlan 15*** as the native VLAN. The port's trunk list includes all VLANs except ***201***-***300***.

```
switch(config)# interface ethernet 12
switch(config-if-Et12)# switchport mode trunk
switch(config-if-Et12)# switchport trunk native vlan 15
switch(config-if-Et12)# switchport trunk allowed vlan except 201-300
switch(config-if-Et12)#
```

- Assume that all ports on the switch are configured with switchport mode trunk similar to Ethernet ***1*** and ***2*** shown below:

```
!
interface ethernet 1
    switchport mode trunk
!
interface ethernet 2
    switchport mode trunk
!
```

- Further assume that ***vlan 30*** is not configured as part of a trunk group.

```
switch# show vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -----------
1     default                          active    Et1, Et2
30    vlan30                           active    Et1, Et2
```

- Now configure ***vlan 30*** as part of ***trunk group 30***:

```
switch(config)# vlan 30
switch(config-vlan-30)# trunk group 30
```

- This updates the VLAN membership for ***vlan 30***.

```
switch#show vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -----------
1     default                          active    Et1, Et2
30    vlan30                           active
```

Note: ***Vlan 30*** is no longer on Et1, Et2 i.e. it has been 'pruned' due to the trunk group command in the vlan configuration.

- To permit ***vlan 30*** on ***Et1***, you need to associate the interface with the trunk group as follows:

```
switch(config-if-Et1)# switchport trunk group 30

Now we see Et1 included in the vlan 30 list

switch# show vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- ----------
1     default                          active    Et1, Et2
30    vlan30                           active    Et1
```

- The trunk group command is not additive to the allowed VLAN command.

```
interface ethernet 1
        switchport mode trunk
        switchport trunk allowed vlan 10
        switchport trunk group trunk30

Vlan 30 will not be permitted on the interface as it is not listed in the
 allowed
vlan list.
```

12.3.3.2.3 Dot1q Tunnel Ports

Dot1q (802.1Q) is a tunneling protocol that encapsulates traffic from multiple customer (c-tag) VLANs in an additional single outer service provider (s-tag) VLAN for transit across a larger network structure that includes traffic from all customers. Tunneling eliminates the service provider requirement that every VLAN be configured from multiple customers, avoiding overlapping address space issues.

Tunneling preserves the inner VLANs through the tunneled network; these inner VLANs are ignored by intermediate devices that make forwarding decisions based only on the outermost VLAN tag (S-Tag)

A dot1q-tunnel port sits at the edge of the tunneled network. Unlike regular access ports, a dot1q-tunnel port does not drop traffic that arrives with ***802.1Q*** tags in place; it ignores existing 802.1Q information and associates arriving traffic (with or without ***802.1Q*** headers) with a new tunnel VLAN ID.

Packets arriving at a tunnel port are encapsulated with an additional ***802.1Q*** tag that can be trunked between multiple devices like any traditional VLAN. When exiting a dot1-tunnel port, the S-Tag is removed to revert the customer traffic to its original tagged or untagged state.

To configure an interface group as a dot1q tunnel port, use the **`switchport mode`** command.

Example

These commands configure ***interface ethernet 12*** as a dot1q tunnel port.

```
switch(config)# interface ethernet 12
switch(config-if-Et12)# switchport mode dot1q-tunnel
switch(config-if-Et12)#
```

To specify the dot1q-tunnel port's access VLAN, use the **switchport access vlan** command. The port then handles all inbound traffic as untagged VLAN traffic.

Example

These commands configure ***vlan 60*** as the access VLAN for ***interface ethernet 12***.

```
switch(config)# interface ethernet 12
switch(config-if-Et12)# switchport access vlan 60
switch(config-if-Et12)#
```

12.3.3.2.4 TPID Configuration

The default Tag Protocol IDentifier (TPID, also called dot1q ethertype) on all switch ports is 0x8100. To configure a different TPID on a port, use the switchport dot1q ethertype command. This feature is available only on 7280E and 7500E platforms.

Note: If dot1q tunneling is enabled on the interface, a TPID configured on the interface becomes irrelevant.

Example

In this provider bridging example, ***interface ethernet 1*** is the user network interface and ***interface ethernet 2*** is the network-to-network interface. These commands configure dot1q tunneling on ***interface ethernet 1*** and set the TPID of ***interface ethernet 2*** to ***0x9100***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# switchport mode dot1q-tunnel
switch(config-if-Et1)# interface ethernet 2
switch(config-if-Et2)# switchport mode trunk
switch(config-if-Et2)# switchport dot1q ethertype 0x9100
switch(config-if-Et2)#
```

In the above configuration, packets from ***Et1*** to ***Et2*** will undergo dot1q-tunneling (stacking of an additional dot1q tag), with an outer TPID of 0x9100 at egress, while packets with outer TPID 0x9100 going from ***Et2*** to ***Et1*** will have the outer tag removed at egress.

12.3.3.2.5 Layer 2 802.1Q Encapsulation

Layer 2 traffic encapsulation is enabled on the configuration mode interface for a specified VLAN through l2-protocol encapsulation dot1q vlan.

Example

These commands enable traffic encapsulation for ***vlan 200*** traffic passing through ***interface ethernet 5/2***.

```
switch(config)# interface ethernet 5/2
switch(config-if-Et5/2)# l2-protocol encapsulation dot1q vlan 200
```

12.3.3.2.6 Port VLAN Scaling on DCS-7160

Port VLAN scaling allows the user to configure a subset of ports in the scale mode. The switchport vlan forwarding command forwards packets between the ports belonging to VLAN in the interface configuration mode. Port-VLAN table is used for storing the configuration on a per port/VLAN combination. The scaling configuration is applicable on a per-port basis and supports a maximum of ***128*** ports.

Note: The configuration is applicable to trunk ports only.

Example

- This command enables VLAN scaling on a port with an ***interface ethernet 2***.

```
switch# config terminal
switch(config)# interface ethernet 2
switch(config-if-Et2)# switchport vlan forwarding accept all
```

- This command disables VLAN scaling on a port.

```
switch# config
switch(config)# interface ethernet 2
switch(config-if-Et2)# no switchport vlan forwarding accept all
```

12.3.3.3 Creating and Configuring VLAN Interfaces

The interface vlan command places the switch in ***VLAN-interface*** configuration mode for modifying an SVI. An SVI provides a management address point and Layer 3 processing for packets from all VLAN ports.

Example

This command enters ***VLAN-interface*** configuration mode for ***vlan 12***. The command also creates ***vlan 12*** interface if it was not previously created.

```
switch# config t
switch(config)# interface vlan 12
switch(config-if-Vl12)#
```

12.3.3.4 Allocating Internal VLANs

The vlan internal order command specifies the VLANs that the switch allocates as internal VLANs when configuring routed ports and the order of their allocation. By default, the switch allocates VLANs in ascending order. The default allocation range is between VLAN ***1006*** and VLAN ***4094***.

The **`no switchport`** command converts an Ethernet or port channel interface into a routed port, disabling Layer 2 switching for the interface.

Examples

- This command configures the switch to allocate internal VLANs in ascending order starting with ***1006***.

```
switch(config)# vlan internal order ascending
switch(config)#
```

- This command configures the switch to allocate internal VLANs in descending order starting with ***4094***.

```
switch(config)# vlan internal order descending
switch(config)#
```

- This command configures the switch to allocate internal VLANs in descending order from ***4094*** through ***4000***.

```
switch(config)# vlan internal order descending range 4000 4094
switch(config)#
```

12.3.3.5 Private VLAN Configuration

On DCS-7280R, DCS-7280R2, DCS-7500R, DCS-7500R2, DCS-7020R

- On systems with algomatch hardware, the access-list mechanism must explicitly be set to TCAM using the following command. Ignore this step when on non-algomatch hardware based systems.

```
switch(config)# hardware access-list mechanism tcam
```

- To enable the private VLAN feature, you must also enable the forwarding-ID feature.

```
switch(config)# platform sand l2 forwarding-id sharing
```

On All Platforms

- Any regular VLAN can act as a primary without any extra configuration. The only requirement is that the VLAN must be active. Use the following command to configure a VLAN as active or inactive:

```
switch(config)# vlan 100
```

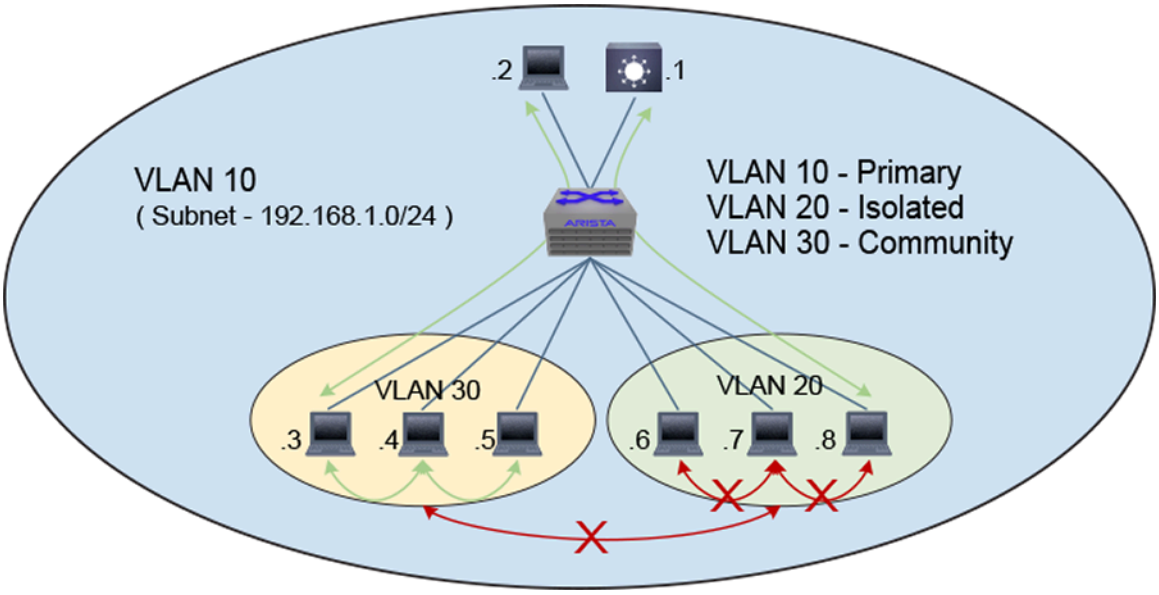
```
switch(config)# no vlan 100
```

```
switch(config)# default vlan 100
```

- Configure VLANs as secondary inside the ***VLAN*** configuration mode. Use the configuration to specify the primary VLAN as isolated and the type of secondary VLAN:

```
switch(config)# vlan 20
switch(config-vlan-20)# private-vlan isolated primary vlan 10
switch(config)# vlan 30
switch(config-vlan-30)# private-vlan community primary vlan 10
```

Figure 12-2: Support for Private VLAN



- Interfaces are assigned to primary or secondary VLANs in the same way as regular VLANs. It works with both access and trunk ports. The following shows the standard switchport command configuring an access interface to the secondary VLAN configured before:

```
switch(config)# interface ethernet 1/1
switch(config-if-Et1/1)# switchport access vlan 20
```

- Trunk ports forward any traffic within the allowed VLANs configured on the interface, whether they are primary or secondary VLANs. To configure trunk ports to translate traffic from primary VLAN to secondary (this maps to the lowest secondary VLAN if multiple are allowed) configure the following on the trunk port:

```
switch(config)# interface ethernet 1/1
switch(config-if-Et1/1)# switchport trunk private-vlan secondary
```

Steps to Unconfigure

On all Platforms

- To unconfigure a private VLAN, use the following command. This reverts the VLAN back to a regular VLAN. At this point, the broadcast domain for this VLAN adjusts and all hosts start to be learned in the regular VLAN, as opposed to the primary VLAN. The MAC table entries previously learned on the primary VLAN are not used anymore for forwarding.

```
switch(config-vlan-20)# no private-vlan
```

- To restore trunk port behavior to allow traffic on all primary and secondary VLANs:

```
switch(config)# interface ethernet1/1
switch(config-if-Et1/1)# no switchport trunk private-vlan secondary
```

On DCS-7280R, DCS-7280R2, DCS-7500R, DCS-7500R2, DCS-7020R

To unconfigure forwarding-id sharing:

```
switch(config)# no platform sand l2 forwarding-id sharing
Note: This configuration needs the device to be rebooted to take effect.
```

12.3.3.6 Configuring VLAN Translation

VLAN translation changes the VLAN ID of specified packets entering or leaving a port. The following sections describe the configuration of VLAN translation.

12.3.3.6.1 Per-port VLAN Translation on Switched Ports

The switchport vlan translation command allows translation of the VLAN tag of traffic entering or exiting a switched port.

To use VLAN translation on a switched port, the port must be configured as a trunk port using the switchport mode command.

Examples

- This command configures ***interface ethernet 5*** as a trunk port.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport mode trunk
switch(config-if-Et5)#
```

- By default, the translation is bidirectional: packets ingressing an interface through ***vlan A*** are internally mapped to ***vlan B***; ***vlan B*** packets egressing the same interface are mapped to ***vlan A***.
  - These commands map ***interface ethernet 5*** traffic with dot1q tag ***50*** to bridging ***vlan 60***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport vlan translation 50 60
```

```
switch(config-if-Et5)#
```

- These commands provides multiple 1:1 VLAN mappings under an interface.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport vlan translation 50 60
switch(config-if-Et5)# switchport vlan translation 61 71
switch(config-if-Et5)# switchport vlan translation 62 72
switch(config-if-Et5)#
```

- These commands translate only incoming packets.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport vlan translation in 50 60
switch(config-if-Et5)#
```

- These commands translate only egress packets.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport vlan translation out 60 50
switch(config-if-Et5)#
```

12.3.3.6.2 Per-port VLAN Translation on Routed Ports

On routed ports, the encapsulation dot1q vlan command (permitted only on routed ports) configures the VLAN on the interface to act as the native VLAN. This command will map packets ingressing with the specified VLAN ID to the internal VLAN ID of the routed port. All traffic egressing out of the routed port will be tagged with the VLAN ID specified in the command.

Example

These commands translate between ***vlan 50*** and the internal VLAN for ***interface ethernet 5*** (a routed port).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no switchport
switch(config-if-Et5)#encapsulation dot1q vlan 50
switch(config-if-Et5)#
```

12.3.3.6.3 Double VLAN Translation

Double VLAN translation creates mappings between an inner and outer VLAN ID pair of a double-tagged packet and a single bridging VLAN. On ingress, specified double-tagged packets are mapped to the bridging VLAN, and on egress packets with the ID of the bridging VLAN are double tagged as specified. By default, the translation is bidirectional, but it can be applied only on ingress or egress.

Example

These commands causes packets entering ***interface ethernet 3/1*** with an outer VLAN ID of ***1000*** and an inner VLAN ID of ***100*** to be processed in bridging ***vlan 200***.

```
switch(config)# interface ethernet 3/1
switch(config-if-Et3/1)# switchport vlan translation in 1000 inner 100 200
switch(config-if-Et3/1)#
```

12.3.4 VLAN Configuration Commands

Global VLAN Configuration Commands

- interface vlan

- vlan
- vlan internal order

VLAN Configuration Mode Commands

- mac address forwarding
- name (VLAN configuration mode)
- state
- trunk group

Layer 2 Interface (Ethernet and Port Channel) Configuration Commands

- switchport access vlan
- switchport dot1q ethertype
- switchport mode
- switchport trunk allowed vlan
- switchport trunk group
- switchport trunk native vlan
- switchport vlan forwarding
- switchport vlan translation

VLAN Interface Configuration Mode Commands

- autostate
- encapsulation dot1q vlan
- l2-protocol encapsulation dot1q vlan
- pvlan mapping

Show Commands

- show dot1q-tunnel
- show interfaces switchport
- show interfaces switchport backup-link
- show interfaces switchport vlan mapping
- show interfaces trunk
- show interfaces vlans
- show pvlan mapping interfaces
- show vlan
- show vlan brief count
- show vlan dynamic
- show vlan internal allocation policy
- show vlan internal usage
- show vlan trunk group

12.3.4.1 autostate

When **`autostate`** is **`enabled`**, the VLAN interface will be up when:

- the corresponding VLAN exists and is in the active state.
- one or more Layer 2 ports in the VLAN are up and in spanning-tree forwarding state.
- the VLAN interface exists and is not in a shutdown state.

Autostate is **enabled** by default. When autostate is **disabled**, the VLAN interface is forced to be active.

- The **no autostate** command disables autostate on the configuration mode interface. The **no autostate** command is stored to ***running-config***.
- The **autostate** command enables the autostate function on the configuration mode VLAN SVI by removing the corresponding **no autostate** statement from ***running-config***.
- The **default autostate** command restores the autostate default state of **enabled** by removing the corresponding **no autostate** statement from ***running-config***.

Command Mode

Interface-VLAN Configuration

Command Syntax

autostate

no autostate

default autostate

Guidelines

Autostate should be disabled on SVIs configured as an MLAG local interface.

Examples

- These commands disable autostate on ***vlan 100***.

```
switch(config)# interface vlan 100
switch(config-if-Vl100)# no autostate
switch(config-if-Vl100)#
```

- These commands enable autostate on ***vlan 100***.

```
switch(config)# interface vlan 100
switch(config-if-Vl100)# autostate
switch(config-if-Vl100)#
```

12.3.4.2 encapsulation dot1q vlan

Routed Port VLAN Translation

In the configuration mode for an Ethernet or port channel interface, the **encapsulation dot1q vlan** translates packets with a dot1q header to the internal VLAN for a routed port. The VLAN in the incoming packets is mapped to the internal VLAN of the routed port, and packets egressing the routed port are encapsulated with a dot1q header for the specified VLAN. For egress packets, no priority information is added to the dot1q header and the priority from the incoming encapsulation will be retained.

Subinterface VLAN Assignment

When used in the configuration mode for an Ethernet or port channel subinterface, however, the **encapsulation dot1q vlan** command assigns a dot1q tag to the subinterface. Traffic ingressing on the parent interface with that dot1q tag will then be sent to the configured subinterface. See **Subinterfaces** and **Subinterface Configuration** for details.

The **no encapsulation dot1q vlan** and **default encapsulation dot1q vlan** commands restore the default VLAN to the configuration mode interface by removing the corresponding **encapsulation dot1q vlan** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-port-channel Configuration

Subinterface-Ethernet Configuration

Subinterface-port-channel Configuration

Command Syntax

`encapsulation dot1q vlan` ***vlan_id***

`no encapsulation dot1q vlan`

`default encapsulation dot1q vlan`

Parameters

vlan_id For VLAN translation, the ID of the external VLAN to be translated; for subinterface configuration, the VLAN of the subinterface. Values range from ***1*** to ***4094***.

Examples

- These commands translate between ***vlan 50*** and the internal VLAN for ***interface ethernet 5*** (a routed port).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no switchport
switch(config-if-Et5)# encapsulation dot1q vlan 50
switch(config-if-Et5)#
```

- These commands assign packets ingressing on ***interface ethernet 1/1*** with ***vlan ID 100*** to subinterface ***ethernet 1/1.1***.

```
switch(config)# interface ethernet1/1.1
switch(config-if-Et1/1.1)# no switchport
switch(config-if-Et1/1.1)# encapsulation dot1q vlan 100
switch(config-if-Et1/1.1)#
```

12.3.4.3 interface vlan

The **`interface vlan`** command places the switch in VLAN-interface configuration mode for modifying parameters of the Switch Virtual Interface (SVI). An SVI provides Layer 3 processing for packets from all ports associated with the VLAN. There is no physical interface for the VLAN.

When entering configuration mode to modify existing SVIs, the command can specify multiple interfaces. The command creates an SVI if the specified interface does not exist prior to issuing the command. When creating an SVI, the command can only specify a single interface.

The **`no interface vlan`** command deletes the specified SVI interfaces from **`running-config`**. The **`default interface vlan`** commands remove all configuration statements for the specified SVI interfaces from ***running-config*** without deleting the interfaces.

Command Mode

Global Configuration

Command Syntax

`interface vlan` ***v_range***

`no interface vlan` ***v_range***

`default interface vlan` ***v_range***

Parameter

v_range VLAN interfaces (number, range, or comma-delimited list of numbers and ranges). VLAN number ranges from ***1*** to ***4094***.

Restrictions

Internal VLANs: A VLAN interface cannot be created or configured for internal VLAN IDs. The switch rejects any interface vlan command that specifies an internal VLAN ID.

Example

This example creates an SVI for ***vlan 12***:

```
switch# config
switch(config)# interface vlan 12
switch(config-if-Vl12)#
```

12.3.4.4 l2-protocol encapsulation dot1q vlan

The **`l2-protocol encapsulation dot1q vlan`** command enables Layer 2 802.1Q traffic encapsulation on the configuration mode interface for a specified VLAN. The default VLAN for all interfaces is VLAN 1.

The **`no l2-protocol encapsulation dot1q vlan`** and **`default l2-protocol encapsulation dot1q vlan`** commands disable the specified encapsulation on the configuration mode interface by removing the corresponding **`l2-protocol encapsulation dot1q vlan`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Command Syntax

`l2-protocol encapsulation dot1q vlan` ***vlan_id***

`no l2-protocol encapsulation dot1q vlan`

`default l2-protocol encapsulation dot1q vlan`

Parameters

vlan_id the ID of the native VLAN. Values range from ***1*** to ***4094***.

Example

These commands enable 802.1Q encapsulation of traffic on ***vlan 200***.

```
switch(config)# interface ethernet 5/2
switch(config-if-Et5/2)# l2-protocol encapsulation dot1q vlan 200
switch(config-if-Et5/2)# show active
interface Ethernet5/2
   l2-protocol encapsulation dot1q vlan 200
switch(config-if-Et5/2)#
```

12.3.4.5 mac address forwarding

The **`mac address forwarding`** command enables a switch to configure a VLAN policy when it receives a packet with an unknown destination MAC address on a VLAN. The command provides three options to configure a VLAN policy:

- Flood the Layer 2 miss packets on the VLAN
- Drop the Layer 2 miss packets
- Log the Layer 2 miss packets to the CPU (while still flooding them on the VLAN)

The default state is to flood the L2 miss packets on all ports of the VLAN.

The **`show vlan`** command displays information about the VLAN policy that is being configured.

The **no** form and the **default** form of the command removes the previously configured VLAN policy on the VLAN.

Command Mode

VLAN Configuration

Command Syntax

`mac address forwarding` [unicast | multicast] **miss action** [drop | flood | log]

`no mac address forwarding` [unicast | multicast] **miss action** [drop | flood | log]

`default mac address forwarding` [unicast | multicast] **miss action** [drop | flood | log]

Parameters

- **unicast** the unicast type of transmission.
- **multicast** the multicast type of transmission.
- **drop** the selected packets are dropped.
- **flood** the selected packets are flooded in the specific VLAN.
- **log** the selected packets are sent to the CPU for logging purpose.

Guidelines

VLAN policy configuration is supported on the Arista 7010, 7050 (excluding 7050SX3-48YC12, 7050CX3-32S, 7050QX2-32S, 7050SX2-72Q, 7050SX2-128, 7050TX2-128), 7060, 7250, and the 7300 series platforms.

VLAN policy is not supported in the following cases:

- STP, LLDP, and LACP packets
- VLAN policy configurations on VXLAN-enabled VLAN
- On a VLAN if IGMP snooping is configured with Multicast miss action is set to drop, then all multicast packets received on that VLAN are dropped.

Examples

- These commands create a ***vlan 333*** and then set the unicast policy to **drop** and the multicast policy to **log** for the specific ***vlan 333***.

```
switch(config)# vlan 333
switch(config-vlan-333)# mac address forwarding unicast miss action drop
switch(config-vlan-333)# mac address forwarding multicast miss action log
```

- These commands display the VLAN policy that was defined when ***vlan 333*** is created.

```
switch(config)# show vlan 333 mac address forwarding

VLAN  UcMissAction  McMissAction
----  ------------  ------------
 333  flood         flood
```

- These commands display the VLAN policy type that was defined when ***vlan 333*** is configured with the **`drop`** unicast policy and the **`log`** multicast policy.

```
switch(config)# show vlan 333 mac address forwarding

VLAN  UcMissAction  McMissAction
----  ------------  ------------
 333  drop          log

switch(config)#show vlan mac address forwarding

VLAN  UcMissAction  McMissAction
```

```
----  ------------  ------------
   1  flood         flood
 333  drop          log
```

12.3.4.6 name (VLAN configuration mode)

The **name** command configures the VLAN name. The name can have up to 32 characters. The default name for VLAN 1 is **default**. The default name for all other VLANs is VLANxxxx, where xxxx is the VLAN number. The default name for ***vlan 55*** is ***VLAN0055***. The show vlan command displays the VLAN name.

The **name** command accepts all characters except the space.

The **no name** and **default name** commands restore the default name by removing the **name** command from ***running-config***.

Command Mode

VLAN Configuration

Command Syntax

name ***label_text***

no name

default name

Parameters

label_text character string assigned to name attribute. Maximum length is ***32*** characters. The space character is not permitted in the name string.

Example

These commands assign ***corporate_100*** as the name for ***vlan 25***, then displays the VLAN name.

```
switch(config)# vlan 25
switch(config-vlan-25)# name corporate_100
switch(config-vlan-25)# show vlan 25
VLAN  Name                            Status    Ports
----- ------------------------------- --------- ---------
25    corporate_100                   active

switch(config-vlan-25)#
```

12.3.4.7 pvlan mapping

The pvlan mapping command maps a Switch Virtual Interface (SVI) available in the primary VLAN to the secondary VLAN or VLANs in the VLAN configuration mode. The show pvlan mapping interfaces command displays the list of mapped VLANs.

The **no pvlan mapping** and **default pvlan mapping** commands restore the default state of the private VLAN mapping.

Command Mode

VLAN Configuration

Command Syntax

pvlan mapping {**add** | **remove** | *vlan ID*}

no pvlan mapping {**add** | **remove** | *vlan ID*}

default pvlan mapping {**add** | **remove** | *vlan ID*}

Parameters

- **add** adding VLANs to the PVLAN mapping of the current VLAN interface.
- **remove** removing VLANs from the PVLAN mapping of the current VLAN interface.
- ***vlan ID*** The secondary VLAN IDs of the private VLAN mapping. The IDs range from ***1*** to ***4094***.

Related Commands

show pvlan mapping interfaces

Example

These commands assign a secondary VLAN ID of ***50*** to the primary VLAN.

```
switch(config)# vlan 25
switch(config-vlan-25)# pvlan mapping 50
switch(config-vlan-25)#
```

12.3.4.8 show dot1q-tunnel

The **`show dot1q-tunnel`** command displays the ports that are configured in dot1q-tunnel switching mode. The switchport mode command configures the switching mode for the configuration mode interface.

Command Mode

EXEC

Command Syntax

`show dot1q-tunnel` [**INTERFACE**]

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **loopback *l_range*** Loopback interface specified by ***l_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
- **vlan *v_range*** VLAN interface range specified by ***v_range***.
- **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.

Valid ***range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the ports that are configured in dot1q-tunnel switching mode.

```
switch> show dot1q-tunnel
dot1q-tunnel mode LAN Port (s)
------------------------------
Po4
Po21
Po22 switch>
```

12.3.4.9 show interfaces switchport backup-link

The **`show interfaces switchport backup-link`** command displays interfaces that are configured as switchport backup pairs and the operational status of each interface. For each pair, the command displays the names, roles, status, and VLAN traffic of each interface.

Command Mode

EXEC

Command Syntax

show interfaces [INTERFACE] **switchport backup-link**

show interfaces switchport backup-link [module {Fabric ***f_num*** | Linecard ***lc_num*** | Supervisor ***svr_num*** | Switchcard | ***1-2*** | ***3-6*** }]

Parameters

- **INTERFACE** Interface type and numbers. Options include:
  - ***no parameter*** Display information for all interfaces.
  - **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
  - **loopback *l_range*** Loopback interface specified by ***l_range***.
  - **management *m_range*** Management interface range specified by ***m_range***.
  - **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
  - **vlan *v_range*** VLAN interface range specified by ***v_range***.

    Valid ***e_range***, ***l_range***, ***m_range***, ***p_range***, and ***v_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **module** Displays interfaces of the specified module. Options include:
  - **Fabric *f_num*** Displays interfaces of the specified fabric module. Value ranges from ***1*** to ***6***.
  - **Linecard *lc_num*** Displays interfaces of the specified linecard module. Value ranges from ***3*** to ***6***.
  - **Supervisor *svr_num*** Displays interfaces of the specified supervisor module. Accepted values are ***1*** and ***2***.
  - **Switchcard** Displays interfaces of switchcard modules.
  - ***1-2*** Displays interfaces of the specified supervisor module.
  - ***3-6*** Displays interfaces of the specified linecard module.

Display Values

- **State** Operational status of the interface. Values include:
  - **Up** Spanning tree mode is *backup*, interface status is **up**.
  - **Down** Spanning tree mode is *backup*, interface status is **down**.
  - **Inactive Configuration** The spanning tree mode is not **backup**.
- **Forwarding vlans** VLANs forwarded by the interface. Depends on interface operation status and prefer option specified by the **switchport backup** command.

Examples

- This command displays the configured switchport primary-backup pairs.

```
switch> show interfaces switchport backup-link
Switch backup interface pair: Ethernet3/17, Ethernet3/8
Primary Interface: Ethernet3/17     State:  Inactive Configuration
Backup Interface:  Ethernet3/8      State:  Inactive Configuration
Preemption delay: 0 milliseconds
Mac move burst size: 0
Mac move burst interval: 20 milliseconds
Mac move destination: ff:ff:ff:ff:ff:ff
```

- This command displays interfaces of the module for ***linecard 4***.

```
switch(config)# show int switchport backup-link module Linecard 4
Switch backup interface pair: Ethernet4/19/1, Ethernet4/19/2
Primary Interface: Ethernet4/19/1   State:  Inactive Configuration
Backup Interface:  Ethernet4/19/2   State:  Inactive Configuration
Preemption delay: 0 milliseconds
Mac move burst size: 0
```

```
Mac move burst interval: 20 milliseconds
Mac move destination: ff:ff:ff:ff:ff:ff
```

12.3.4.10 show interfaces switchport vlan mapping

The **show interfaces switchport vlan mapping** command displays mapping information of the configured VLANs in an interface mode.

Command Mode

EXEC

Command Syntax

show interfaces switchport vlan mapping

Examples

- This command displays mapping information of the configured VLAN IDs.

```
switch# show interfaces switchport vlan mapping
--------------
Ethernet3
                                    Direction   Direction
Original Vlan  New Vlan  Status    Configured  Active
-------------- --------- --------- ----------- -----------
10             100       Active    In/Out      In/Out
11             200       Active    In          In
300            12        Active    Out         Out
```

- This command displays dual tag mapping information of the configured VLAN IDs.

```
switch(config)# show interfaces switchport vlan mapping
--------------
Ethernet3/1
Direction     Direction
Outer Tag     Inner Tag     VLAN ID     Status      Configured    Active Dot1  qTunnel
-----------   -----------   ---------   ---------   -----------   -----------  -----------
1000          100           200         active        In/Out        In/Out      -

1001          101           201         active        In            In          -

1002          102           202         active        Out           Out          -
```

- This command displays dual tag mapping information of the configured VLAN IDs.

```
switch(config)# show interfaces switchport vlan mapping
--------------
Ethernet1/1
                                                 Direction     Direction
Outer Tag    Inner Tag     VLAN ID     Status     Configured    Active
-----------  -----------   ---------   ---------  -----------   -----------
70           -             300         Active      In/Out        In/Out
10           50            100         Active      In/Out        In/Out
20           60            100         Active      In            In
30           40            200         Active      Out           Out
```

12.3.4.11 show interfaces switchport

The **show interfaces switchport** command displays the switching configuration and operational status of the specified ports.

Command Mode

EXEC

Command Syntax

show interfaces [**INTERFACE**] **switchport**

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display the switching status for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **loopback *l_range*** Loopback interface specified by ***l_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
- **vlan *v_range*** VLAN interface range specified by ***v_range***.

  Valid ***e_range***, ***l_range***, ***m_range***, ***p_range***, and ***v_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the switching status for all interfaces.

```
switch(config)# show interface switchport
Default switchport mode: access

Name: Et5/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
MAC Address Learning: enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Static Trunk Groups:
Dynamic Trunk Groups:

Name: Et5/2
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
MAC Address Learning: enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Static Trunk Groups:
Dynamic Trunk Groups:

[...]

switch(config)#
```

- This command displays the switching status of port channel interfaces ***21*** and ***22***.

```
switch> show interface port-channel 21-22 switchport
Name: Po21
Switchport: Enabled
Administrative Mode: tunnel
Operational Mode: tunnel
Access Mode VLAN: 1 (inactive)
Trunking Native Mode VLAN: 100 (VLAN0100)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Trunk Groups: foo

Name: Po22
Switchport: Enabled
Administrative Mode: tunnel
Operational Mode: tunnel
Access Mode VLAN: 1 (inactive)
```

```
Trunking Native Mode VLAN: 1 (inactive)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Trunk Groups:

switch>
```

- This command displays the configured status of VLAN scaling for the ***interface ethernet 2/1*** port.

```
switch# show interface Ethernet 2/1 switchport
Name: Ethernet 2/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
MAC Address Learning: enabled
Dot1q ethertype/TPID: 0x8100 (active)
Dot1q VLAN Tag: Allowed
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Static Trunk Groups:
Dynamic Trunk Groups:
Source interface filtering: enabled
VLAN forwarding mode: allConfiguredVlans switch>
```

12.3.4.12 show interfaces trunk

The **show interfaces trunk** command displays configuration and status information for interfaces configured in switchport trunk mode.

Command Mode

EXEC

Command Syntax

show interfaces [**INTERFACE**] **trunk**

Parameters

INTERFACE Interface type and numbers. Options include:

- ***no parameter*** Display information for all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** Management interface range specified by ***m_range***.
- **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.

  Valid ***e_range***, ***m_range***, and ***p_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the trunk status for all interfaces configured in switchport trunk mode.

```
switch> show interfaces trunk
Port            Mode            Status          Native vlan
Po1             trunk           trunking        1
Po2             trunk           trunking        1

Port            Vlans allowed
Po1             1-15
Po2             16-30
```

```
Port            Vlans allowed and active in management domain
Po1             1-10
Po2             21-30


Port            Vlans in spanning tree forwarding state
Po1             1-10
Po2             21-30 switch>
```

12.3.4.13 show interfaces vlans

The **show interfaces vlans** command displays a table that lists the VLANs that are carried by the specified interfaces. Interfaces that do not carry VLANs are not listed in the table. The table lists the untagged (native or access) and tagged VLANs for each interface.

Command Mode

EXEC

Command Syntax

show interfaces [**INT_NAME**] **vlans**

Parameters

INT_NAME Interface type and number. Values include:

- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.

Example

This command displays the VLANs carried by all L2 ports.

```
switch> show interfaces vlans
Port       Untagged Tagged
Et9        3910     -
Et11       3912     -
Et16       500      -
Et17       3908     -
Et18       3908     -
Po1        1        101-102,500,721,3000,
Po2        101      -
Po4        3902     -
Po5        3903     -
Po6        3992     -
Po7        661      -
Po8        3911     -
```

12.3.4.14 show pvlan mapping interfaces

The **show pvlan mapping interfaces** command displays information about the private VLAN mapping interfaces.

Command Mode

EXEC

Command Syntax

show pvlan mapping interfaces

Example

This command displays information about the private VLAN mapping interfaces.

```
switch(config)# int vlan 50
switch(config-if-Vl50)# pvlan mapping 70
switch(config-if-Vl50)# show pvlan mapping interfaces
Interface    Secondary Vlans
---------    ---------------
Vlan50       70
```

12.3.4.15 show vlan

The **show vlan** command displays the VLAN ID, name, status, and member ports of all configured VLANs. The command only displays active ports by default; by specifying **configured-ports**, the command displays all ports that are members of a configured VLAN regardless of their activity status, including Ethernet ports that are members of a port channel.

Command Mode

EXEC

Command Syntax

show vlan [**VLAN_LIST**] [**PORT_ACTIVITY**]

Parameters

- **VLAN_LIST** List of VLANs displayed by command. Options include:
  - ***no parameter*** all VLANs.
  - ***v_range*** VLANs specified by ***v_range***.
  - **id *v_range*** VLANs specified by ***v_range***.
  - **name *v_name*** VLANs specified by the VLAN name ***v_name***.

  v_range formats include number, number range, or comma-delimited list of numbers and ranges.
- **PORT_ACTIVITY** Ports listed in table. Options include:
  - ***no parameter*** table displays only active ports (same as **active-configuration** option).
  - **active-configuration** table displays only active ports.
  - **configured-ports** table displays all configured ports.

Display Values

- **VLAN** The VLAN ID.
- **Name** The name of the VLAN.
- **Status** The status of the VLAN.
- **Ports** The ports that are members of the VLAN.

Examples

- This command displays status and ports of VLANs ***1-1000***.

```
switch> show vlan 1-1000
VLAN  Name                      Status    Ports
----- ------------------------- --------- --------------
1     default                   active    Po1
184   fet.arka                  active    Cpu, Po1, Po2
262   mgq.net                   active    PPo2, Po1
512   sant.test                 active    Cpu, Et16, Po1
821   ipv6.net                  active    Cpu, Po1, Po7 switch>
```

- This command displays the list of all the member interfaces under each SVI.

```
switch# show vlan
```

```
VLAN  Name                      Status    Ports
----- ------------------------- --------- ----------------
1     default                   active
2148  VLAN2148                  active    Cpu, Et1, Et26
2700  VLAN2700                  active    Cpu, Et18
```

12.3.4.16 show vlan brief count

The **show vlan brief count** command displays the number of VLANs that are configured on the switch.

Command Mode

EXEC

Command Syntax

show vlan brief count

Example

This command displays the number of VLANs on the switch.

```
switch> show vlan brief count
Number of existing VLANs           : 18 switch>
```

12.3.4.17 show vlan dynamic

The **show vlan dynamic** command displays the source and quantity of dynamic VLANs on the switch. Dynamic VLANs support VM Tracer monitoring sessions.

Command Mode

EXEC

Command Syntax

show vlan dynamic

Example

This command displays the source and quantity of dynamic VLANs on the switch.

```
switch> show vlan dynamic
Dynamic VLAN source       VLANS
vmtracer-poc              88 switch>
```

12.3.4.18 show vlan internal allocation policy

The **show vlan internal allocation policy** command displays the method the switch uses to allocate VLANs to routed ports. The vlan internal order command configures the allocation method.

The allocation method consists of two configurable components:

- **range:** the list of VLANs that are allocated to routed ports.
- **direction:** the direction by which VLANs are allocated (ascending or descending).

Command Mode

EXEC

Command Syntax

show vlan internal allocation policy

Example

This command displays the internal allocation policy.

```
switch> show vlan internal allocation policy
Internal VLAN Allocation Policy: ascending
Internal VLAN Allocation Range: 1006-4094 switch>
```

12.3.4.19 show vlan internal usage

The **show vlan internal usage** command shows the VLANs that are allocated as internal VLANs for routed ports.

A routed port is an Ethernet or port channel interface that is configured as a layer 3 interface. Routed ports do not bridge frames and are not members of any VLANs. Routed ports can have IP addresses assigned to them and packets are routed directly to and from the port.

When an interface is configured as a routed port, the switch allocates an SVI with a previously unused VLAN ID. The switch prohibits the configuration of VLANs with numbers corresponding to internal VLAN interfaces allocated to a routed port. VLAN interfaces corresponding to SVIs allocated to a routed port cannot be configured by VLAN interface configuration mode commands.

Command Mode

EXEC

Command Syntax

show vlan internal usage

Example

This command displays the VLANs that are allocated to routed ports.

```
switch> show vlan internal usage
1006  Ethernet3
1007  Ethernet4 switch>
```

12.3.4.20 show vlan trunk group

The **show vlan trunk group** command displays the trunk group membership of the specified VLANs.

Command Mode

EXEC

Command Syntax

show vlan [**VLAN_LIST**] **trunk group**

Parameters

VLAN_LIST VLAN list. Options include:

- ***no parameter*** all VLANs.
- ***v_range*** VLANs specified by ***v_range***.
- **id *v_range*** VLANs specified by ***v_range***.
- **name *v_name*** VLANs specified by the VLAN name ***v_name***.

Display Values

- **VLAN** VLAN ID.
- **Trunk Groups** Trunk groups associated with the listed VLANs.

Example

This command displays the trunk group membership of all configured VLANs.

```
switch> show vlan trunk group
VLAN     Trunk Groups
----     ------------------------------------
5
10       first_group
12
40       second_group
100      third_group
101      middle_group
102
200 switch>
```

12.3.4.21 state

The **state** command configures the VLAN transmission state of the configuration mode VLAN.

- **Active state**: Ports forward VLAN traffic.
- **Suspend state**: Ports block VLAN traffic.

The default transmission status is **active**.

The **no state** command restores the default VLAN transmission state to the configuration mode VLAN by removing the corresponding **state** command from ***running-config***.

Command Mode

VLAN Configuration

Command Syntax

state OPERATION_STATE

no state

default state

Parameters

OPERATION_STATE VLAN transmission state. Options include:

- **active** VLAN traffic is forwarded.
- **suspend** VLAN traffic is blocked.

Example

These commands suspend VLAN traffic on VLANs ***100-102***.

```
switch(config)# vlan 100-102
switch(config-vlan-100-102)# state suspend
switch(config-vlan-100-102)#
```

12.3.4.22 switchport access vlan

The **switchport access vlan** command specifies the access VLAN of the configuration mode interface. Ethernet or port channel interfaces that are in access mode are members of only the access VLAN. Untagged frames that the interface receives are associated with the access VLAN. Frames tagged with the access VLAN are also associated with the access VLAN. The interface drops all other tagged frames that it receives. By default, VLAN 1 is the access VLAN of all Ethernet and port channel interfaces.

An interface's access mode is effective only when the interface is in access mode or dot1q-tunnel mode, as specified by the switchport mode command. Interfaces in dot1q-tunnel mode handle inbound traffic as

untagged traffic and associate all traffic with the access VLAN. Interfaces configured to switchport trunk mode maintain and ignore existing switchport access commands.

The **no switchport access vlan** and **default switchport access vlan** commands restore ***VLAN 1*** as the access VLAN of the configuration mode interface by removing the corresponding **switchport access vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Command Syntax

switchport access vlan ***v_num***

no switchport access vlan

default switchport access vlan

Parameters

v_num number of access VLAN. Value ranges from ***1*** to ***4094***. Default is ***1***.

Example

These commands assign ***VLAN 100*** as the access VLAN to ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport access vlan 100
switch(config-if-Et5)#
```

12.3.4.23 switchport dot1q ethertype

The **switchport dot1q ethertype** command configures the tag protocol identifier (TPID, also known as a dot1q ethertype), of the configuration mode interface. By default, all switch ports use the standard TPID of ***0x8100***.

The **no switchport dot1q ethertype** and **default switchport dot1q ethertype** commands restore the TPID to ***0x8100*** by removing the corresponding **switchport dot1q ethertype** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

switchport dot1q ethertype ***ethertype***

no switchport dot1q ethertype

default switchport dot1q ethertype

Parameters

ethertype ethertype number (TPID). Value ranges from ***0x600*** (***1536***) through ***0xFFFF*** (***65535***), and can be entered in decimal or hexadecimal notation. Value is stored and displayed in hexadecimal form; the default value is ***0x8100***.

Example

These commands configure ***0x9100*** as the TPID of ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport dot1q ethertype 0x9100
```

```
switch(config-if-Et5)#
```

12.3.4.24 switchport mode

The **switchport mode** command specifies the switching mode of the configuration mode interface. The switch supports five switching modes: access, trunk, dot1q-tunnel, tap, and tool.

- **Access switching mode**: The interface is a member of one VLAN, called the access VLAN, as specified by the switchport access vlan command. Tagged frames received on the interface are dropped unless they are tagged with the access VLAN. Frames transmitted from the interface are always untagged.
- **Trunk switching mode**: The interface may be a member of multiple VLANs, as configured by the switchport trunk allowed vlan command. Untagged traffic is associated with the interface's native VLAN, as configured with the switchport trunk native vlan command.
- **Dot1q-tunnel switching mode**: The interface treats all inbound packets as untagged traffic and handles them as traffic of its access VLAN, as specified by the switchport access vlan command.
- **Tap mode**: The interface operates as a tap port. Tap ports receive traffic for replication on one or more tool ports.The interface may be a member of multiple VLANs, as configured by the **switchport tap allowed vlan** command. Untagged traffic is associated with the interface's native VLAN, as configured with the **switchport tap native vlan** command.

Tap ports are in STP forwarding state and prohibit egress traffic. MAC learning, control plane interaction and traps for inbound traffic are disabled.

Tool mode: The interface operates as a tool port. Tool ports replicate traffic received by tap ports. The interface may be a member of multiple VLANs, as configured by the **switchport tool allowed vlan** command. MAC learning, control plane interaction and traps for inbound traffic are disabled.

Tool ports are in STP forwarding state and prohibit ingress traffic that uses port settings.

The status of switchport configured ports depends on the switch's tap aggregation mode (which can be viewed by using the **mode** command):

- tap aggregation mode enabled: tap and tool ports are enabled. Switching ports are errdisabled.
- tap aggregation mode disabled: tap and tool ports are errdisabled. Switching ports are enabled.

The **no switchport mode** and **default switchport mode** commands return the configuration mode interface to its default setting as an access port by deleting the corresponding **switchport mode** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Command Syntax

switchport mode MODE_TYPE

no switchport mode

default switchport mode

Parameters

MODE_TYPE switching mode of the configuration mode interfaces. Options include:

- **access** access switching mode.
- **dot1q-tunnel** dot1q-tunnel switching mode.
- **tap** tap switching mode.
- **tool** tool switching mode.
- **trunk** trunk switching mode.

Restrictions

Dot1q-tunnel switching mode is not available on Petra platform switches.

Tap aggregation (tap and tool modes) is available on FM6000 and Arad platform switches.

Example

These commands configure ***interface ethernet 4*** as a trunk port.

```
switch(config)# interface ethernet 4
switch(config-if-Et4)# switchport mode trunk
switch(config-if-Et4)#
```

12.3.4.25 switchport trunk allowed vlan

The **switchport trunk allowed vlan** command creates or modifies the list of VLANs for which the configuration mode interface, in trunk mode, handles tagged traffic. By default, interfaces handle tagged traffic for all VLANs. Command settings persist in ***running-config*** without taking effect when the switch is in tap aggregation mode or the interface is not in trunk mode.

The **no switchport trunk allowed vlan** and **default switchport trunk allowed vlan** commands restore the trunk mode default allowed VLAN setting of **all** by removing the corresponding **switchport trunk allowed vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Command Syntax

switchport trunk allowed vlan EDIT_ACTION

no switchport trunk allowed vlan

default switchport trunk allowed vlan

Parameters

EDIT_ACTION modifications to the VLAN list.

- ***v_range*** Creates VLAN list from *v_range*.
- **add *v_range*** Adds specified VLANs to current list.
- **all** VLAN list contains all VLANs.
- **except *v_range*** VLAN list contains all VLANs except those specified.
- **none** VLAN list is empty (no VLANs).
- **remove *v_range*** Removes specified VLANs from current list.

  Valid ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Example

These commands create the trunk mode allowed VLAN list of ***6-10*** for ***interface ethernet 14***, then verifies the VLAN list.

```
switch(config)# interface ethernet 14
switch(config-if-Et14)# switchport trunk allowed vlan 6-10
switch(config-if-Et14)# show interfaces ethernet 14 switchport
Name: Et14
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Access Mode VLAN: 1 (inactive)
Trunking Native Mode VLAN: 1 (inactive)
Administrative Native VLAN tagging: disabled
```

```
Trunking VLANs Enabled: 6-10
Trunk Groups:

switch(config-if-Et14)#
```

12.3.4.26 switchport trunk group

The **switchport trunk group** command assigns the configuration mode interface to the specified trunk group. Trunk group ports handle traffic of the VLANs assigned to the group.

The **no switchport trunk group** and **default switchport trunk group** commands remove the configuration mode interface from the specified trunk group by deleting the corresponding statement from running-config. If the command does not specify a trunk group, the interface is removed from all trunk groups to which it is assigned.

Note: On platforms which support the use of port channels as mirror destinations, a port channel which is being used as a mirror destination *must not* be assigned to an MLAG.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Command Syntax

switchport trunk group [***group_name***]

no switchport trunk group [***group_name***]

default switchport trunk group [***group_name***]

Parameters

group_name trunk group name.

Example

These commands assign ***port channel 4*** to trunk group ***fe-1***.

```
switch(config)# interface port-channel 4
switch(config-if-Po4)# switchport trunk group fe-1
switch(config-if-Po4)#
```

12.3.4.27 switchport trunk native vlan

The **switchport trunk native vlan** command specifies the trunk mode native VLAN for the configuration mode interface. Interfaces in trunk mode associate untagged frames with the native VLAN. Trunk mode interfaces can also be configured to drop untagged frames. The default native VLAN for all interfaces is VLAN 1.

The **no switchport trunk native vlan** and **default switchport trunk native vlan** commands restore ***vlan 1*** as the trunk mode native VLAN to the configuration mode interface by removing the corresponding **switchport trunk native vlan** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Command Syntax

switchport trunk native vlan VLAN_ID

no switchport trunk native vlan

default switchport trunk native vlan

Parameters

- **VLAN_ID** the ID of the native VLAN. Options include:
  - ***v_num*** VLAN number. Value ranges from ***1*** to ***4094***.
  - **tag** interface drops all untagged frames.

Example

These commands configure ***vlan 100*** as the native VLAN for ***port channel 21***.

```
switch(config)# interface port-channel 21
switch(config-if-Po21)# switchport trunk native vlan 100
switch(config-if-Po21)#
```

12.3.4.28 switchport vlan forwarding

The **switchport vlan forwarding** command forwards packets between the ports belonging to VLAN in the interface configuration mode. The scaling configuration is applicable on a per-port basis. In the 7160 platform, the hardware uses a Port-VLAN table for storing the configuration on a per port/VLAN combination and supports a maximum of ***128*** ports.

Note: The configuration is applicable to trunk ports only.

Command Mode

Interface-Ethernet Configuration

Command Syntax

switchport vlan forwarding [**accept** | **all**]

Parameters

- **accept** accepts packets for VLAN.
- **all** all VLANs.

Example

This command forwards and accepts all the packets of VLAN of ***interface ethernet 2***.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# switchport vlan forwarding accept all
switch(config-if-Et2)#
```

12.3.4.29 switchport vlan translation

The **switchport vlan translation** command allows you to map packets from one VLAN to another using VLAN translation. This is carried out on packets having a dot1q header (tagged frames) only. The translation rewrites the VLAN ID (VID) field in dot1q headers on packets passing through a switched port without changing any other fields.

By default, the translation is bidirectional. The packets ingressing an interface through ***vlan A*** are internally mapped to ***vlan B***; ***vlan B*** packets egressing the same interface are mapped to ***vlan A***.

To use VLAN translation on a switched port, the port must be configured as a trunk port using the **switchport mode** command.

VLAN translation on routed ports is accomplished through the encapsulation dot1q vlan command.

The **no switchport vlan translation** and **default switchport vlan translation** commands remove VLAN mapping by removing the switchport vlan translation command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Command Syntax

switchport vlan translation [**DIRECTION**] ***incoming_vlanid new_vlanid***

no switchport vlan translation ***incoming_vlanid new_vlanid***

no switchport vlan translation **DIRECTION** ***incoming_vlanid***

default switchport vlan translation ***incoming_vlanid new_vlanid***

default switchport vlan translation **DIRECTION** ***incoming_vlanid***

Parameters

- **DIRECTION** direction of traffic to be translated.
  - ***no parameter*** translates the specified VLAN IDs for transmitted and received traffic.
  - **in** translates the specified VLAN IDs for received traffic only.
  - **out** translates the specified VLAN IDs for transmitted traffic only.
  - ***incoming_vlanid*** Enter the VLAN ID to be translated. Value ranges from ***1*** to ***4094***.
- ***new_vlanid*** The new VLAN ID or bridging VLAN ID that will be used internally. Value ranges from ***1*** to ***4094***.

Example

- These commands translate only incoming packets, changing the VLAN ID to ***2008*** in the dot1q header of packets ingressing on ***vlan 201***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport vlan translation in 201 2008
switch(config-if-Et5)#
```

- These commands translate multiple VLAN mappings on an ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport vlan translation 50 60
switch(config-if-Et5)# switchport vlan translation 61 71
switch(config-if-Et5)# switchport vlan translation 62 72
switch(config-if-Et5)#
```

12.3.4.30 trunk group

The **trunk group** command assigns the configuration mode VLAN to a specified trunk group.

A trunk group is the set of physical interfaces that comprise the trunk and the collection of VLANs whose traffic is carried on the trunk. The traffic of a VLAN that belongs to one or more trunk groups is carried only on ports that are members of trunk groups to which the VLAN belongs. Switchport commands specify the physical interfaces that carry trunk group traffic.

The **no trunk group** and **default trunk group** commands remove the configuration mode VLAN from the specified trunk group by removing the corresponding **trunk group** statement from ***running-config***. If a trunk group is not specified, the commands remove the configuration mode VLAN from all trunk groups.

Command Mode

VLAN Configuration

Command Syntax

trunk group [***name***]

no trunk group [***name***]

default trunk group [***name***]

Parameters

name a name representing the trunk group.

Example

These commands assigns ***vlan 49*** to the trunk group ***mlagpeer***:

```
switch(config)# vlan 49
switch(config-vlan-49)# trunk group mlagpeer
switch(config-vlan-49)#
```

12.3.4.31 vlan

The **vlan** command places the switch in VLAN configuration mode to configure a set of virtual LANs. The command creates the specified VLANs if they do not exist prior to issuing the command. A VLAN that is in use as an internal VLAN may not be created or configured. The switch rejects any **vlan** command that specifies an internal VLAN ID.

The **default vlan** and **no vlan** commands removes the VLAN statements from ***running-config*** for the specified VLANs.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

vlan ***vlan_range***

no vlan ***vlan_range***

default vlan ***vlan_range***

Parameters

vlan_range VLAN list.

Formats include a name, number, number range, or comma-delimited list of numbers and ranges.

Commands Available in VLAN Configuration Mode

- name (VLAN configuration mode)
- state
- trunk group

Guidelines

In MLAG configurations, VLANs operate as follows:

- The VLAN must be configured identically on both MLAG peer switches.
- The port-specific bridging configuration originates on the switch where the port is physically located. This configuration includes the switchport access VLAN, switchport mode (trunk or access), trunk-allowed VLANs, the trunk native VLAN, and the switchport trunk groups.

Example

This command creates ***vlan 49*** and enters ***VLAN*** configuration mode for the new VLAN:

```
switch(config)# vlan 49
switch(config-vlan-49)#
```

12.3.4.32 vlan internal order

The **`vlan internal order`** command specifies the range that the switch can allocate as internal VLANs when configuring routed ports and the order of their allocation. By default, the switch allocates VLANs in ascending order from VLAN ***1006*** to VLAN ***4094***.

The **`no vlan internal order`** and **`default vlan internal order`** commands revert the policy to its default.

Command Mode

Global Configuration

Command Syntax

`vlan internal order` **DIRECTION** [**RANGE_VLAN**]

`no vlan internal order`

`default vlan internal order`

Parameters

- **DIRECTION** VLAN allocation number direction. Options include:
  - **ascending** allocates internal VLANs from lower VLAN bound to upper VLAN bound.
  - **descending** allocates internal VLAN from upper VLAN bound to lower VLAN bound.
- **RANGE_VLAN** allocation range. Options include:
  - ***no parameter*** ***1006*** (lower bound) to ***4094*** (upper bound).
  - **range**, **lower**, **upper** specifies lower bound (**lower**) and upper bound (**upper**).

Examples

- This command configures the switch to allocate internal VLANS from ***3000*** through ***3999***.

```
switch(config)# vlan internal order ascending range 3000 3999
switch(config)#
```

- This command configures the switch to allocate internal VLANS from ***4094*** through ***1006***.

```
switch(config)# vlan internal order descending
switch(config)#
```

- This command configures the switch to allocate internal VLANS from ***4094*** down through ***4000***.

```
switch(config)# vlan internal order descending range 4000 4094
switch(config)#
```

- This command reverts the allocation policy to its default (ascending, between ***1006*** and ***4094***).

```
switch(config)# no vlan internal order
switch(config)#
```

12.4 DCBX and Flow Control

This section describes Data Center Bridging Capability Exchange (DCBX) configuration tasks. Topics in this section include:

- Introduction
- Overview
- DCBX Configuration and Verification
- Configuring Priority-Flow-Control (PFC)

- Configuring PFC Watchdog
- DCBX and Flow Control Commands

12.4.1 Introduction

EOS implements Link Layer Discovery Protocol (LLDP) and the Data Center Bridging Capability Exchange (DCBX) protocol to help automate the configuration of Data Center Bridging (DCB) parameters, including the Priority-Based Flow Control (PFC) standard, which allows an end-to-end flow-control feature.

This feature enables a switch to recognize when it is connected to an iSCSI device and automatically configure the switch link parameters (such as priority flow control) to provide optimal support for that device. DCBX can be used to prioritize the handling of iSCSI traffic to help ensure that packets are not dropped or delayed. DCBX is off by default.

12.4.2 Overview

12.4.2.1 Data Center Bridging Capability Exchange (DCBX)

DCBX works with LLDP to allow switches to exchange information about their Data Center Bridging (DCB) capabilities and configuration and automatically negotiate common Priority-Based Flow Control (PFC) parameters.

Data is exchanged in Type-Length-Value (TLV) format. For DCBX to function on an interface LLDP must be enabled on that interface as well.

12.4.2.2 Priority-Based Flow Control (PFC)

Priority-Based Flow Control (PFC) uses a new control packet defined in IEEE 802.1Qbb and is not compatible with 802.3x Flow Control (FC). An interface that is configured for PFC will be disabled for FC. When PFC is disabled on an interface, the FC configuration for the interface becomes active. Any FC frames received on a PFC configured interface are ignored.

Each priority is configured as either drop or no-drop. If a priority that is designated as no-drop is congested, the priority is paused. Drop priorities do not participate in pause.

When PFC is disabled, the interface defaults to the IEEE 802.3x flow control setting for the interface. PFC is disabled by default.

12.4.2.3 PFC Watchdog

The PFC watchdog identifies the egress queues that are unable to transmit packets for a long time due to receiving continuous PFC pause frames. On identifying such stuck tx-queue PFC watchdog error-disables the respective port with a error-disable reason of stuck-queue. When there is an error reported on a port the traffic is re-routed through a different port to the destination.

The PFC watchdog supports the following PFC watchdog configurations:

- PFC watchdog forced recovery of queues
- PFC watchdog polling interval configuration
- PFC Watchdog non-disruptive priorities configuration
- Displaying stuck queue and recovery counters

12.4.3 DCBX Configuration and Verification

12.4.3.1 Set the Priority Rank to the Traffic Class

The dcbx application priority command assigns a priority rank to the specified traffic class in the application priority table. This table is transmitted on each DCBX-enabled interface.

Examples

- These commands tell the DCBX peer that iSCSI frames (TCP ports ***860*** and ***3260***) should be assigned the given priority of ***5***.

```
switch(config)# dcbx application tcp-sctp 860 priority 5
switch(config)# dcbx application tcp-sctp 3260 priority 5
```

- These commands specify a different priority for the two iSCSI traffic ports.

```
switch(config)# dcbx application tcp-sctp 860 priority 3
switch(config)# dcbx application tcp-sctp 3260 priority 4
```

- This command is equivalent to the **dcbx application tcp-sctp** command. The DCBX peer that iSCSI frames are assigned are the given the priority ***5***.

```
switch(config)# dcbx application iscsi priority 5
switch(config)#
```

- These commands prevent the peers from sending anything about the iSCSI frames.

```
switch(config)# no dcbx application tcp-sctp 860 priority 5
switch(config)# no dcbx application tcp-sctp 3260 priority 5
```

12.4.3.2 Configuring CEE DCBX Priority Group

The priority-flow-control priority command configures the Enhanced Transmission Selection (ETS) to the specified QoS group, and sets the traffic class priority and the bandwidth percentage for the packets in the traffic class.

Examples

- This command configures the ETS to the QoS group map and assigns the CoS map value as ***7*** and sets traffic class priority to ***5***.

```
switch(config)# dcbx ets qos map cos 7 traffic-class 5
```

- This command configures the ETS to the traffic class and sets the traffic class priority as ***7*** and bandwidth value to ***70*** percent.

```
switch(config)# dcbx ets traffic-class 7 bandwidth 70
```

12.4.3.3 DCBX Verification

To display the DCBX status and the interfaces on which DCBX is enabled, use the show dcbx command.

Example

This command displays the DCBX status for ***Ethernet 50***.

```
switch# show dcbx Ethernet 50
Ethernet50:
 IEEE DCBX is enabled and active
 Last LLDPDU received on Thu Feb 14 12:06:01 2013
 No priority flow control configuration TLV received
 No application priority configuration TLV received switch#
```

12.4.4 Configuring Priority-Flow-Control (PFC)

12.4.4.1 Enable Priority-Flow-Control (PFC)

The priority-flow-control command enables Priority-Flow-Control (PFC) on an individual port.

Example

The **priority-flow-control** command in DCBX mode enables PFC on an interface.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# priority-flow-control on
```

12.4.4.2 Set the Priority Flow Control Priority

The priority-flow-control priority command in DCBX mode creates a priority group that pauses priority. Each priority is configured as either drop or no-drop. If a priority that is designated as no-drop is congested, the priority is paused. Drop priorities do not participate in pause.

Examples

- The **priority-flow-control priority** command in DCBX mode creates a priority group that pauses priority ***5*** on ***interface ethernet 2***.

  ```
  switch(config)# interface ethernet 2
  switch(config-if-Et2)# priority-flow-control on
  switch(config-if-Et2)# priority-flow-control priority 5 no-drop
  ```

- To enable lossy behavior, use the drop option of the **priority-flow-control priority** command.

  ```
  switch(config)# interface ethernet 2
  switch(config-if-Et2)# priority-flow-control on
  switch(config-if-Et2)# priority-flow-control priority 5 drop
  ```

12.4.4.3 Disable Priority-Flow-Control (PFC)

To disable Priority Flow Control (PFC) on the configuration mode interface and restore the default packet drop setting on the interface, use the priority-flow-control priority command.

Example

To disable PFC, use the **no priority-flow-control** command.

```
switch (config)# interface ethernet 2
switch(config-if-Et2)# no priority-flow-control
```

12.4.5 Configuring PFC Watchdog

12.4.5.1 Enabling PFC Watchdog

The **priority-flow-control pause watchdog default timeout** command starts monitoring all the egress queues which have guaranteed bandwidth enabled and for the priorities on which PFC is enabled.

Note: To enable PFC watchdog, user is required to configure guaranteed bandwidth on the tx-queue to be monitored. Also, PFC must be enabled on the port for the traffic flowing into the queue that is being monitored.

Example

These commands enable the PFC watchdog monitoring on ***tx-queue 3*** of ***Ethernet 1/1***, and configures a PFC congestion timeout of ***10*** seconds which error-disables the port if the queue is stuck.

```
switch# config
switch(config)# interface Ethernet1/1
switch(config-if-Et1/1)# priority-flow-control on
switch(config-if-Et1/1)# priority-flow-control priority 3 no-drop
switch(config-if-Et1/1)# tx-queue 3
switch(config-if-Et1/1-txq-3)# bandwidth guaranteed 100
switch(config-if-Et1/1-txq-3)# exit
switch(config-if-Et1/1)# exit
switch(config)# priority-flow-control pause watchdog default timeout 10
```

12.4.5.2 Enabling PFC Watchdog Queue Recovery

The **`priority-flow-control pause watchdog default recovery-time forced`** command recovers a stuck queue after the PFC storm ceases. PFC watchdog supports the following two recovery methods.

- Auto Recovery – recover queue(s) after the PFC storm ceases.
- Forced Recovery – recover queue(s) after a fixed duration, irrespective of PFC storm being received.

Note: The default recovery mode is “auto”.

Example

This command recovers a stuck queue after a fixed duration of ***10*** seconds.

```
switch(config)# priority-flow-control pause watchdog default recovery-time 10
 forced
```

12.4.5.3 Configuring PFC Watchdog Polling Interval

The **`priority-flow-control pause watchdog default polling-interval`** command configures the frequency at which queues should be checked for stuck or recovery detection. By default, polling interval is calculated internally or it considers the value configured through the CLI.

Note: Configuring a very low polling interval may increase load on the CPU.

Example

This command configures a polling interval of ***10*** seconds on the switch.

```
switch(config)# priority-flow-control pause watchdog default polling-interval
 10
```

12.4.5.4 Displaying Stuck Queue and Recovery Counters

The **`show priority-flow-control counters watchdog`** command displays the value of number of times a queue is identified as stuck and recovered. These counters are maintained only for those queues that have PFC watchdog functionality enabled. These counters are cleared when either PFC or PFC watchdog configuration is disabled. Alternatively, show interfaces priority-flow-control counters watchdog command can be used to display the counters.

Examples

- This command displays the value of number of times the queue was stuck and recovered for all the interfaces.

```
switch# show priority-flow-control counters watchdog
Port       TxQ   Total times   Total times
                 stuck         recovered
-------   ----   -----------   -----------
Et1/1      UC2             2             2
Et1/1      UC3             3             3
Et2/1      UC2            12            12
Et2/1      UC3            31            30
```

- This command displays the value of number of times the queue was stuck and recovered for a specific interface. In this case it is ***Et1/1***.

```
switch# show priority-flow-control interfaces Ethernet 1/1 counters watchdog
Port       TxQ   Total times   Total times
                 stuck         recovered
-------   ----   -----------   -----------
Et1/1      UC2             2             2
Et1/1      UC3             3             3
```

12.4.5.5 PFC Watchdog Non-disruptive Priorities

The PFC Watchdog acts to drop the traffic entering or leaving the port at the stuck PFC priority. Later when the queue recovers, this action is reversed. While applying these actions, some traffic (for all priorities) is dropped on that port. In such case, the priority-flow-control pause watchdog hardware non-disruptive priority command can be used to avoid the traffic drop on ports at stuck queues.

This traffic drop can be avoided by configuring specific PFC priorities as non-disruptive. When queues corresponding to these priorities are stuck/recovered, the traffic for other priorities are not impacted.

Examples

- This command configures the specific PFC priorities as non-disruptive, and the priority is set to ***3***.

```
switch(config)# priority-flow-control pause watchdog hardware non-disruptive
 priority 3
```

- This command configures all the ports, having a subset of non-disruptive priorities as a part of their no-drop priorities, start in non-disruptive mode.

```
switch(config)# priority-flow-control pause watchdog hardware port non-
disruptive-only
```

12.4.5.6 Displaying PFC Watchdog Information

The `show priority-flow-control` command displays the PFC watchdog status information. Note, if the PFC watchdog default timeout value is non-zero then PFC watchdog is active on the switch.

Example

This command displays the PFC watchdog default timeout value, in this show example the timeout value is ***3.0*** which means the PFC watchdog is active.

```
switch# show priority-flow-control
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7
The PFC watchdog default timeout is 3.0

Port   Enabled Priorities Active Note
Et1/1   Yes       34      Yes    DCBX disabled
```

```
Et1/2   Yes       34      Yes    DCBX disabled
Et1/3   Yes       34      Yes    DCBX disabled
Et1/4   Yes       34      Yes    DCBX disabled
```

The **show interface status errdisabled** command displays the port which is error-disabled due to stuck- queue condition.

Example

This command displays the interface Ethernet ***Eth1/1*** status as errdisabled and the reason.

```
switch# show interface Eth1/1 status errdisabled

Port            Name          Status         Reason
---------- ---------------- ---------------- ---------
Et1/1                       errdisabled    stuck-queue
```

The **show priority-flow-control status** command displays the current PFC watchdog details.

Example

This command displays the PFC watchdog configuration details at global and interface level.

```
switch# show priority-flow-control status
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7
The PFC watchdog timeout is 1.0 second(s)
The PFC watchdog recovery-time is 2.0 second(s) (auto)
The PFC watchdog polling-interval is 0.1 second(s)
The PFC watchdog non-disruptive priorities are 3 4
The PFC watchdog port non-disruptive-only is False

E: PFC Enabled, D: PFC Disabled, A: PFC Active, W: PFC Watchdog Enabled
Port     Status  Priorities Note
Et1/1    E A W    1     7   DCBX disabled
Et1/2    E A -              DCBX disabled
Et1/3    D - -
Et1/4    D - -
Et2/1    D - -
..
..
```

12.4.6 DCBX and Flow Control Commands

Configuration Commands

- dcbx application priority
- dcbx ets
- dcbx mode
- no priority-flow-control
- platform fm6000 pfc-wm
- priority-flow-control
- priority-flow-control pause watchdog action
- priority-flow-control pause watchdog default
- priority-flow-control pause watchdog hardware
- priority-flow-control priority

Show Commands

- show dcbx

- show dcbx application-priority-configuration
- show dcbx priority-flow-control-configuration
- show dcbx status
- show interfaces priority-flow-control
- show platform fm6000 pfc-wm
- show priority-flow-control

12.4.6.1 dcbx application priority

The **dcbx application priority** command assigns a priority rank to the specified traffic class in the application priority table. This table is transmitted on each DCBX-enabled interface.

The **no dcbx application priority** and **default dcbx application priority** commands remove the specified DCBX traffic class priority assignment by deleting the corresponding **dcbx application priority** command from ***running-config***. When the command does not specify a traffic class, all DCBX traffic class priority assignments are removed.

Command Mode

Global Configuration

Command Syntax

dcbx application [**APPLICATION_TYPE priority**] *rank*

no dcbx application [**APPLICATION_TYPE priority**]

default dcbx application [**APPLICATION_TYPE priority**]

Parameters

- **APPLICATION_TYPE** traffic class receiving the priority assignment. Options include:
  - **ether** ***ethertype_number*** Ethernet traffic. ***ethertype_number*** varies from ***1536*** to ***65535***.
  - **iscsci** iSCSCI traffic. Maps to TCP/SCTP ports ***860*** and ***3260***.
  - **tcp-sctp** ***port_number*** TCP/SCTP traffic. Port number varies from ***1*** to ***65535***.
  - **tcp-sctp-udp** ***port_number*** TCP/SCTP/UDP traffic. Port number varies from ***1*** to ***65535***.
  - **udp** ***port_number*** UDP traffic. Port number varies from ***1*** to ***65535***.
- ***rank*** priority assigned to traffic class. Values range from ***0*** to ***7***.

Examples

- These commands tell the DCBX peer that iSCSI frames (TCP ports ***860*** and ***3260***) should be assigned the given priority of ***5***.

```
switch(config)# dcbx application tcp-sctp 860 priority 5
switch(config)# dcbx application tcp-sctp 3260 priority 5
```

- These commands specify a different priority for the two iSCSI traffic ports.

```
switch(config)# dcbx application tcp-sctp 860 priority 3
switch(config)# dcbx application tcp-sctp 3260 priority 4
```

- This command is equivalent to the **dcbx application tcp-sctp** command. The DCBX peer that iSCSI frames are assigned to is given priority ***5***.

```
switch(config)# dcbx application iscsi priority 5
switch(config)#
```

- These commands prevent the peers from sending anything about the iSCSI frames.

```
switch(config)#no dcbx application tcp-sctp 860 priority
```

```
switch(config)#no dcbx application tcp-sctp 3260 priority
```

12.4.6.2 dcbx ets

The **dcbx ets** command configures the enhanced transmission selection (ETS) to the specified QoS group, and sets the traffic class priority and the bandwidth percentage for the packets in the traffic class.

The **no dcbx ets** and **default dcbx ets** commands remove the specified DCBX traffic classs priority assignment by deleting the corresponding **dcbx ets** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

dcbx ets [qos map cos ***value*** traffic-class ***value*** | traffic-class ***value*** bandwidth ***value***]

no dcbx ets [qos map cos ***value*** traffic-class ***value*** | traffic-class ***value*** bandwidth ***value***]

default dcbx ets [qos map cos ***value*** traffic-class ***value*** | traffic-class ***value*** bandwidth ***value***]

Parameters

- **qos** QoS to configure.(The sub options include):
  - **map** QoS map to configure.
  - **cos** CoS value assigned to port. Value ranges from ***0*** to ***7***. Default value is ***0***.
  - **traffic-class** Assigns the traffic-class priority to the QoS map. The value ranges from ***0*** to ***7***.
- **traffic-class** Assigns the traffic class priority. The value ranges from ***0*** to ***7***. (The sub options include):
  - **bandwidth** The percentage of bandwidth assigned to the packets received from traffic class. The value ranges from ***0*** to ***100*** in percentage. The default value is ***0***.

Examples

- This command configures the ETS to the QoS group map and assigns the CoS map value as ***7*** and sets the traffic class priority to ***5***.

```
switch(config)# dcbx ets qos map cos 7 traffic-class 5
```

- This command configures the ETS to the traffic class and sets the traffic class priority value to ***7*** and sets the bandwidth value to ***70*** percent.

```
switch(config)# dcbx ets traffic-class 7 bandwidth 70
```

12.4.6.3 dcbx mode

The **dcbx mode** command enables DCBX mode on the configuration mode interface. The switch supports ***IEEE P802.1Qaz***. When DCBX is enabled, two TLVs are added to outgoing LLDPDUs, which instruct the peer on the interface to configure PFC (priority flow control) and the application priority table in the same way as the switch.

The **no dcbx mode**, **default dcbx mode**, and **dcbx mode none** commands disable DCBX on the configuration mode interface by removing the corresponding **dcbx mode** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

dcbx mode **MODE_NAME**

no dcbx mode

default dcbx mode

Parameters

MODE_NAME Specifies the DCBX version. Options include:

- **ieee** IEEE version.
- **cee** Converged Enhanced Ethernet version.
- **none** DCBX is disabled.

Examples

- These commands enable ***interface ethernet 2*** to use IEEE DCBX.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# dcbx mode ieee
switch(config-if-Et2)#
```

- These commands disable DCBX on ***interface ethernet 2***.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# dcbx mode none
switch(config-if-Et2)
```

12.4.6.4 no priority-flow-control

The **no priority-flow-control** and **default priority-flow-control** commands disable the priority flow control (PFC) on the configuration mode interface and restore the default packet drop setting on the interface, which takes effect when PFC is re-enabled. The commands delete all corresponding priority-flow-control commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

no priority-flow-control

default priority-flow-control

Example

These commands disable Priority Flow Control (PFC) on ***interface ethernet 3***.

```
switch(config)# interface ethernet 3
switch(config-if-Et3)# no priority-flow-control
switch(config-if-Et3)#
```

12.4.6.5 priority-flow-control

The **priority-flow-control** command enables Priority Flow Control (PFC) on the configuration mode interface to pause selected traffic classes.

The **no priority-flow-control** and **default priority-flow-control** commands disable PFC on the configuration mode interface by deleting the corresponding **priority-flow-control** command from ***running-config***. The priority-flow-control priority command also disables PFC on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

priority-flow-control on

no priority-flow-control on

default priority-flow-control on

Example

- These commands enable PFC on ***interface ethernet 3***.

```
switch(config)# interface ethernet 3
switch(config-if-Et3)# priority-flow-control on
switch(config-if-Et3)#
```

- These commands disable PFC on ***interface ethernet 3***.

```
switch(config)# interface ethernet 3
switch(config-if-Et3)# no priority-flow-control
switch(config-if-Et3)#
```

12.4.6.6 platform fm6000 pfc-wm

The **platform fm6000 pfc-wm** command configures the hardware buffer space allocated to the Priority Flow Control (PFC) RX-Private buffer. The command provides options to configure the buffer size and specify when PFC frames are sent to request that a neighbor stop sending traffic. The default values are as follows:

- **RX-Private:** ***18400*** bytes
- **on (watermark):** ***9280*** bytes
- **off (watermark):** ***1600*** bytes

Values that are entered in the command are rounded up to the closest multiple of ***160***. The **RX-Private** value must be greater than the **off** value, which must be larger than the **on** value.

The no **platform fm6000 pfc-wm** and **default platform fm6000 pfc-wm** commands restore the default settings by removing the **platform fm6000 pfc-wm** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

platform fm6000 pfc-wm [**RX-PRIVATE_SIZE**][**PFC-ON_WM**][**PFC-OFF_WM**]

no platform fm6000 pfc-wm

default platform fm6000 pfc-wm

The platform **fm6000 pfc-wm** command must explicitly configure at least one parameter.

Parameters

- **RX-PRIVATE_SIZE** Specifies size of rx-private buffer. Options include:
  - ***no parameter*** rx-private buffer retains previously configured size.
  - **rx-private** ***18268*** to ***102400*** Size of rx-private buffer (bytes).
- **PFC-ON_WM** Buffer capacity that triggers the switch to send PFC frames. Options include:
  - ***no parameter*** Parameter retains previously configured value.
  - **on** ***9134*** to ***102400*** Buffer capacity that triggers PFC frames (bytes).
- **PFC-OFF_WM** Buffer capacity that triggers the switch to stop PFC frame transmissions. Options include:
  - ***no parameter*** Parameter retains previously configured value.
  - **off** ***1536*** to ***102400*** Buffer capacity that turns off PFC frames.

Related Command

show platform fm6000 pfc-wm displays the PFC RX-Private buffer memory allocations.

Example

This command configures the rx-private hardware buffer.

```
switch(config)# platform fm6000 pfc-wm rx-private 24800 on 16000 off 3200
switch(config)#
```

12.4.6.7 priority-flow-control pause watchdog action

The **priority-flow-control pause watchdog** action command either drops the traffic on a stuck queue, or error disables the port which has a stuck queue, or notifies if there is no action on the stuck queue. The following actions are performed based on the queue status.

The **no priority-flow-control pause watchdog** action command removes the specified **priority-flow-control pause watchdog action configuration** by deleting the corresponding **priority-flow-control pause watchdog action** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

priority-flow-control pause watchdog action

no priority-flow-control pause watchdog action

Parameters

action PFC watchdog action for stuck transmit queues. Options include.

- **drop** Drop traffic on the stuck queue.
- **errdisable** Error disable port which has the stuck transmit queue.
- **notify-only** No action on the stuck queue.

Guidelines

Before enabling the PFC watchdog configuration, configure the guaranteed bandwidth on the tx-queue to be monitored. Also, enable the PFC on the port for the PFC priorities for the traffic flowing into the queue that is being monitored.

Example

These commands enables the pfc-watchdog monitoring on ***tx-queue 3*** of ***Ethernet 1/1***, and configures a PFC watchdog action drop and drops the traffic if the queue is a stuck queue.

```
switch# config
switch(config)# interface Ethernet1/1
switch(config-if-Et1/1)# priority-flow-control on
switch(config-if-Et1/1)# priority-flow-control priority 3 no-drop
switch(config-if-Et1/1)# tx-queue 3
switch(config-if-Et1/1-txq-3)# bandwidth guaranteed 100
switch(config-if-Et1/1-txq-3)# exit
switch(config-if-Et1/1)# exit
switch(config)# priority-flow-control pause watchdog action drop
```

12.4.6.8 priority-flow-control pause watchdog default

The **priority-flow-control pause watchdog** default command monitors all the egress queues which have guaranteed bandwidth enabled and for the priorities on which PFC is enabled. Guaranteed bandwidth

is needed to ensure starvation due to higher priority traffic is not wrongly flagged as a stuck-queue due to congestion. The stuck duration after which the port needs to be error disabled is also configurable.

The **no priority-flow-control pause watchdog default** command removes the specified priority-flow-control pause watchdog configuration by deleting the corresponding **priority-flow-control pause watchdog** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

priority-flow-control pause watchdog default

no priority-flow-control pause watchdog default

Parameters

default Specifies the default value. Options include.

- ***polling-interval*** Configures the interval at which the watchdog should poll the queues. The polling interval value ranges from ***0.005*** to ***30*** seconds.
- ***recovery-time*** Configures recovery-time after which stuck queue should recover and start forwarding. The recovery-time value ranges from ***0.01*** to ***60*** seconds.
  - **forced** Force recover any stuck queue(s) after the recovery-time interval, irrespective of whether PFC frames are being received or not.
- ***timeout*** Configures timeout after which port should be errdisabled or should start dropping on congested priorities. The timeout value ranges from ***0.01*** to ***60*** seconds.

Guidelines

Before enabling the PFC watchdog configuration, configure the guaranteed bandwidth on the tx-queue to be monitored. Also, enable the PFC on the port for the PFC priorities for the traffic flowing into the queue that is being monitored.

- Polling Interval Discrepancy

  For user configured polling-interval to be valid, it must satisfy the following conditions

  When the recovery-mode is auto and timeout, recovery-time, and polling-interval are non-default, polling-interval <= min (timeout, recovery-time) / 2,

  When recovery-mode is forced or recovery-time is not configured, polling-interval <= (timeout / 2)

  For better functioning of PFC Watchdog, when user configured polling interval is too large compared to either timeout or recovery time values, Watchdog will use auto calculated value instead of user configured value until the discrepancy is resolved. Also, CLI warning and syslog messages are generated to inform user of the discrepancy.
- CLI Warnings

  When there is discrepancy between timeout and polling-interval, the format of the message is as shown below:

```
! User configured polling interval <user-cfgd polling-interval> second(s)
 is
greater than half of timeout <user-cfgd timeout> second(s). Setting
polling-interval to <to-be-used polling-interval> second(s)
```

  When there is discrepancy between recovery-time and polling-interval, the format of the message is as shown below

```
! User configured polling interval <user-cfgd polling-interval> second(s)
 is
```

```
greater than half of recovery-time <user-cfgd recovery-time> second(s).
 Setting
polling-interval to <to-be-used polling-interval> second(s)
```

Examples

- These commands enable the pfc-watchdog monitoring on tx-queue ***3*** of ***interface ethernet1/1***, and configures a PFC congestion timeout of ***10*** seconds which error-disables the port if the queue is stuck.

```
switch# config
switch(config)# interface ethernet1/1
switch(config-if-Et1/1)# priority-flow-control on
switch(config-if-Et1/1)# priority-flow-control priority 3 no-drop
switch(config-if-Et1/1)# tx-queue 3
switch(config-if-Et1/1-txq-3)# bandwidth guaranteed 100
switch(config-if-Et1/1-txq-3)# exit
switch(config-if-Et1/1)# exit
switch(config)# priority-flow-control pause watchdog default timeout 10
```

- These commands enable the pfc-watchdog monitoring on tx-queue ***3*** of ***interface ethernet1/1***, and configures a PFC forced recovery-time interval of ***30*** seconds after which the stuck queue(s) are recovered, irrespective of whether PFC frames are being received or not.

```
switch# config
switch(config)# interface ethernet1/1
switch(config-if-Et1/1)#priority-flow-control on
switch(config-if-Et1/1)#priority-flow-control priority 3 no-drop
switch(config-if-Et1/1)#tx-queue 3
switch(config-if-Et1/1-txq-3)#bandwidth guaranteed 100
switch(config-if-Et1/1-txq-3)#exit
switch(config-if-Et1/1)#exit
switch(config)#priority-flow-control pause watchdog default recovery-time 30
 forced
```

- These commands enable the pfc-watchdog monitoring on tx-queue ***3*** of ***Ethernet 1/1***, and configures a PFC polling-interval of ***20*** seconds after which queue is polled.

```
switch# config
switch(config)# interface Ethernet1/1
switch(config-if-Et1/1)# priority-flow-control on
switch(config-if-Et1/1)# priority-flow-control priority 3 no-drop
switch(config-if-Et1/1)# tx-queue 3
switch(config-if-Et1/1-txq-3)# bandwidth guaranteed 100
switch(config-if-Et1/1-txq-3)# exit
switch(config-if-Et1/1)# exit
switch(config)# priority-flow-control pause watchdog default polling-inter
val 20
```

12.4.6.9 priority-flow-control pause watchdog hardware

The **`priority-flow-control pause watchdog`** hardware command configures specific PFC priorities as non-disruptive. This will avoid traffic drop on queues corresponding to these priorities are stuck/recovered, the traffic for other priorities are not impacted.

The **`no priority-flow-control pause watchdog hardware`** command removes the specified ***priority-flow-control pause watchdog non-disruptive*** configuration by deleting the corresponding **`priority-flow-control pause watchdog hardware`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

priority-flow-control pause watchdog hardware

no priority-flow-control pause watchdog hardware

Parameters

hardware Configure PFC priority through hardware. Options include:

- ***non-disruptive*** PFC watchdog non-disruptive configuration. The priority value ranges from ***0*** to ***7***.

Guidelines

Before enabling the PFC watchdog configuration, configure the guaranteed bandwidth on the tx-queue to be monitored. Also, enable the PFC on the port for the PFC priorities for the traffic flowing into the queue that is being monitored.

Example

These commands enable the pfc-watchdog monitoring on tx-queue ***3*** of ***interface ethernet 1/1***, and configures PFC priorities as non-disruptive on PFC priorities ***3*** and ***4***.

```
switch# config
switch(config)# interface ethernet1/1
switch(config-if-Et1/1)# priority-flow-control on
switch(config-if-Et1/1)# priority-flow-control priority 3 no-drop
switch(config-if-Et1/1)# tx-queue 3
switch(config-if-Et1/1-txq-3)# bandwidth guaranteed 100
switch(config-if-Et1/1-txq-3)# exit
switch(config-if-Et1/1)# exit
switch(config)# priority-flow-control pause watchdog hardware non-disruptive
 priority 3 4
```

12.4.6.10 priority-flow-control priority

The **priority-flow-control priority** command configures the packet resolution setting on the configuration mode interface. This setting determines if packets are dropped when priority flow control (PFC) is enabled on the interface. Packets are dropped by default.

The **no priority-flow-control priority** and **default priority-flow-control priority** commands restore the default packet drop setting on the configuration mode interface by deleting the corresponding **priority-flow-control priority** command from ***running-config***. The **priority-flow-control priority** command also restores the default setting on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

priority-flow-control priority pack-drop

no priority-flow-control priority

default priority-flow-control priority

Parameters

pack-drop denotes the interfaces. Options include:

- **drop** Packets are dropped. Default setting.
- **no drop** Packets are not dropped.

Examples

- These commands in DCBX mode create a priority group that pauses dot1p priority ***5*** on ***interface ethernet 2***.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# priority-flow-control on
switch(config-if-Et2)# priority-flow-control priority 5 no-drop
```

- These commands enable lossy behavior.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# priority-flow-control on
switch(config-if-Et2)# priority-flow-control priority 5 drop
```

- These commands remove the priority group that pauses dot1p priority ***5*** on ***interface ethernet 2***.

```
switch(config)# interface ethernet 2
switch(config-if-Et2)# priority-flow-control on
switch(config-if-Et2)# no priority-flow-control priority
```

12.4.6.11 show dcbx

The **`show dcbx`** command list DCBX status and the interfaces on which DCBX is enabled.

Command Mode

EXEC

Command Syntax

`show dcbx` [**INTERFACE**]

Parameters

INTERFACE Interface type and number. Options include:

- ***no parameter*** all configured DCBX interfaces.
- **ethernet *e-num*** Ethernet interface specified by ***e-num***.

Examples

- This command displays the DCBX status for ***ethernet 50***.

```
switch# show dcbx ethernet 50
Ethernet50:
  IEEE DCBX is enabled and active
  Last LLDPDU received on Thu Feb 14 12:06:01 2013
  No priority flow control configuration TLV received
  No application priority configuration TLV received switch#
```

- This command displays the DCBX status for ***ethernet 50*** when Priority Flow Control (PFC) is not enabled.

```
switch# show dcbx ethernet 50
Ethernet50:
  IEEE DCBX is enabled and active
  Last LLDPDU received on Thu Feb 14 12:08:29 2013
  - PFC configuration: willing
    not capable of bypassing MACsec
    supports PFC on up to 4 traffic classes
    PFC enabled on priorities: 5 7
    WARNING: peer PFC configuration does not match the local PFC
 configuration
  - Application priority configuration:
    2 application priorities configured:
      tcp-sctp 860 priority 5
```

```
       tcp-sctp 3260 priority 5 switch#
```

12.4.6.12 show dcbx application-priority-configuration

The **show dcbx application-priority-configuration** command displays the DCBX peer application priority configuration.

Command Mode

EXEC

Command Syntax

show dcbx [**INTERFACE**] **application-priority-configuration**

Parameters

INTERFACE Interface type and number. Options include:

- ***no parameter*** All configured DCBX interfaces.
- **ethernet *e-num*** Ethernet interface specified by ***e-num***.

Guidelines

This command and the show priority-flow-control command function identically.

Example

This command displays the DCBX peer application priority configuration for all DCBX-enabled interfaces.

```
switch# show dcbx application-priority-configuration
Ethernet1:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
  No application priority configuration TLV received
Ethernet2:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
  No application priority configuration TLV received
...
Ethernet50:
  Last LLDPDU received on Thu Feb 14 12:08:29 2013
  - Application priority configuration:
    2 application priorities configured:
      tcp-sctp 860 priority 5
      tcp-sctp 3260 priority 5 switch#
```

12.4.6.13 show dcbx priority-flow-control-configuration

The **show dcbx priority-flow-control-configuration** command displays the IEEE DCBX peer priority flow control configurations.

Command Mode

EXEC

Command Syntax

show dcbx [**INTERFACE**] **priority-flow-control-configuration**

Parameters

INTERFACE Interface type and number. Options include:

- ***no parameter*** all configured DCBX interfaces.
- **ethernet *e-num*** Ethernet interface specified by ***e-num***.

Example

This command displays the DCBX peer priority flow control configuration for the DCBX-enabled interfaces on the device.

```
switch# show dcbx priority-flow-control-configuration
Ethernet1:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
  No priority flow control configuration TLV received
Ethernet2:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
  No priority flow control configuration TLV received
...
Ethernet50:
  Last LLDPDU received on Thu Feb 14 12:11:29 2013
  - PFC configuration: willing
    not capable of bypassing MACsec
    supports PFC on up to 4 traffic classes
    PFC enabled on priorities: 5 7
    WARNING: peer PFC configuration does not match the local PFC configuration
switch#
```

12.4.6.14 show dcbx status

The **show dcbx status** command displays the DCBX status on the interfaces on which DCBX is enabled.

Command Mode

EXEC

Command Syntax

show dcbx [**INTERFACE**] **status**

Parameters

INTERFACE Interface type and number. Options include:

- ***no parameter*** all configured DCBX interfaces.
- **ethernet *e-num*** Ethernet interface specified by ***e-num***.

Example

This command displays the DCBX status for the DCBX-enabled interfaces.

```
switch# show dcbx status
Ethernet1:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
Ethernet2:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
Ethernet50:
  IEEE DCBX is enabled and active
  Last LLDPDU received on Thu Feb 14 12:11:54 2013 switch#
```

12.4.6.15 show interfaces priority-flow-control

The **show interfaces priority-flow-control** command displays the status of PFC on all interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [**INTERFACE**] **priority-flow-control** [**INFO_LEVEL**]

Parameters

- **INTERFACE** Interface type and numbers. Options include:
  - ***no parameter*** Display information for all interfaces.
  - **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
  - **loopback *l_range*** Loopback interface specified by ***l_range***.
  - **management *m_range*** Management interface range specified by ***m_range***.
  - **port-channel *p_range*** Port-Channel Interface range specified by ***p_range***.
  - **vlan *v_range*** VLAN interface range specified by ***v_range***.
  - **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.

  Valid range formats include number, number range, or comma-delimited list of numbers and ranges.
- **INFO_LEVEL** specifies the type of information displayed. Options include:
  - ***no parameter*** Displays information about all DCBX neighbor interfaces.
  - **status** Displays the DCBX status.
  - **counters** Displays the DCBX counters.

Guidelines

This command and the **show priority-flow-control** command function identically.

Example

This command displays the PFC for all interfaces.

```
switch# show interfaces priority-flow-control
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7

Port    Enabled Priorities Active Note
Et1     No                 No
Et2     No                 No
...
Et50    Yes          5     Yes
...
Port                RxPfc           TxPfc
Et1                     0               0
Et2                     0               0
...
Et50                    0               0
...
switch#
```

12.4.6.16 show platform fm6000 pfc-wm

The **`show platform fm6000 pfc-wm`** command displays the buffer space allocated to the RX-Private buffer and buffer levels that trigger PFC frame transmission activities.

Command Mode

Privileged EXEC

Command Syntax

`show platform fm6000 pfc-wm`

Related Command

priority-flow-control priority specifies the PFC RX-Private buffer memory allocation.

Example

This command displays the rx-private hardware buffer memory allocation.

```
switch# show platform fm6000 pfc-wm
```

```
Pfc_Rx_Private_WM: 24800 Bytes
Pfc_On_WM: 16000 Bytes
Pfc_Off_WM: 3200 Bytes switch#
```

12.4.6.17 show priority-flow-control

The **show priority-flow-control** command displays the status and other PFC and PFC watchdog information on all interfaces if no specific interface is specified.

Command Mode

EXEC

Command Syntax

show priority-flow-control [status | counters | ***interfaces***]

Parameters

- ***interfaces*** specifies the interface for which the information is displayed. Options include:
  - **Ethernet** Hardware Ethernet interface.
  - **Loopback** Loopback interface.
  - **Management** Management interface.
  - **Port-Channel** Lag interface.
  - **Recirc-Channel** Recirculation interface.
  - **Tunnel** Tunnel interface.
  - **Vlan** VLAN interface.
  - **Vxlan** VXLAN Tunnel Interface.
- ***status*** displays the interface PFC status.
- ***counters*** displays the interface PFC counters. Options include:
  - ***detail*** displays the DCBX counters for each priority class. This option is available only on Trident switches.
  - ***watchdog*** displays the PFC watchdog counters.

Examples

- This command displays the PFC status on all interfaces.

```
switch# show priority-flow-control
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7
Port    Enabled Priorities Active Note

Et1     No                 No

Et2     No                 No

...
Et50    Yes          5     Yes

...
Port                 RxPfc           TxPfc
Et1                      0               0
Et2                      0               0
...
Et50                     0               0
...
```

- This command displays the PFC watchdog status. If PFC watchdog default timeout is non-zero (in this case it is 3.0) then PFC watchdog is actively running on the switch.

```
switch# show priority-flow-control
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7
The PFC watchdog default timeout is 3.0

Port   Enabled Priorities Active Note
Et1/1   Yes       34      Yes    DCBX disabled
Et1/2   Yes       34      Yes    DCBX disabled
Et1/3   Yes       34      Yes    DCBX disabled
Et1/4   Yes       34      Yes    DCBX disabled
...
```

- This command displays the current value of these counters for all the interfaces being monitored by PFC watchdog. Alternatively, show interfaces priority-flow-control counters watchdog command can be used for the same.

```
switch# show priority-flow-control counters watchdog
Port       TxQ   Total times   Total times
                 stuck         recovered
-------   ----   -----------   -----------
Et1/1      UC2            2             2
Et1/1      UC3            3             3
Et2/1      UC2           12            12
Et2/1      UC3           31            30
```

- This command displays the current value of these counters for a specific subset of interfaces. Alternatively, **show interfaces priority-flow-control counters watchdog** command can be used for the same.

```
switch# show priority-flow-control interfaces ethernet 1/1 counters watchdog
Port       TxQ   Total times   Total times
                 stuck         recovered
-------   ----   -----------   -----------
Et1/1      UC2            2             2
Et1/1      UC3            3             3
```

- This command displays the configuration details of PFC watchdog at global and interface level.

```
switch# show priority-flow-control status
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7
The PFC watchdog timeout is 1.0 second(s)
The PFC watchdog recovery-time is 2.0 second(s) (auto)
The PFC watchdog polling-interval is 0.1 second(s)
The PFC watchdog non-disruptive priorities are 3 4
The PFC watchdog port non-disruptive-only is False

E: PFC Enabled, D: PFC Disabled, A: PFC Active, W: PFC Watchdog Enabled
Port     Status  Priorities Note
Et1/1    E A W    1     7   DCBX disabled
Et1/2    E A -              DCBX disabled
Et1/3    D - -
Et1/4    D - -
Et2/1    D - -
..
```

12.5 IP Locking

This section describes IP Locking configuration tasks. Topics in this section include:

- Release Updates
- IPv4 Locking Configuration Commands
- IPv6 Locking
- Show Commands
- Limitations
- IP Locking Commands

12.5.1 IP Locking

IP Locking is an EOS feature configured on an Ethernet Layer 2 port.

When enabled, IP Locking ensures that a port will only permit IP and ARP packets with IP source addresses that have been authorized. As of the ***EOS Release 4.25.0F*** release update, IP locking can run in two modes - IPv4 Locking (which will be referred to as IP Locking) and IPv6 Locking, which is configured using the commands mentioned in the following sections. IP Locking prevents another host on a different interface from claiming ownership of an IP address through ARP spoofing. IPv6 Locking extends this behavior to IPv6 packets, including ICMPv6 Neighbor Discovery (Router Advertisement, Redirect) and DHCP (server-to-client) packets. Mode specific commands are also discussed.

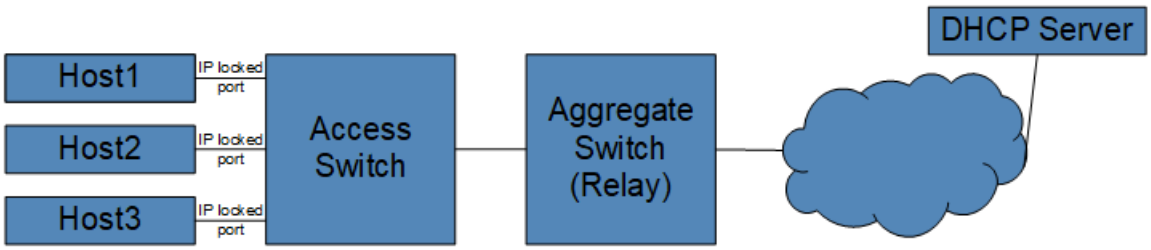
On an IP Locked Port,

- ARP probes with ***0.0.0.0*** as Sender Protocol Address (SPA) and is allowed for Duplicate Address Detection (DAD).
- Incoming DHCP server response packets are dropped to avoid rogue device(s) acting as DHCP server(s).
- Incoming DHCP client request packets are allowed for devices to complete DHCP handshake and obtain DHCP leases.

On an IPv6 Locked Port,

- Incoming DHCPv6 server response packets are dropped and incoming DHCPv6 client request packets are allowed.
- Incoming ICMPv6 ND : Router Advertisement packets are dropped as only routers should send these packets.
- Incoming ICMPv6 ND : Router Solicitation packets are allowed.
- Incoming ICMPv6 ND : Redirect packets are dropped as only routers should send these packets.

Figure 12-3: IP Locking



IP Locking relies on DHCP LeaseQuery (***RFC4388***) and MAC address learning to determine that an IP address has been authorized on a particular port. It is necessary to ensure that DHCP servers used in the network allow LeaseQuery messages.

12.5.1.1 Release Updates

Refer to the release updates for IP Locking.

***EOS Release 4.25.1F*:**

Added support for disabling address filtering for IPv6 packets while still keeping all packet type specific drop rules such as ICMPv6 ND:RA, and ICMPv6 ND:Redirect and DHCPv6 (server-to-client) packets, using the **`locked-address ipv6 enforcement disabled`** command.

***EOS Release 4.24.0F*:**

- Added support for expiration modes of locked addresses, using the **`locked-address expiration mac disabled`** command.
- Added support for counters, using the **`show address locking counters`** and **`clear address locking counters`** commands.

***EOS Release 4.23.2F*:**

Added support for **`static lease`** command, **`lease <V4ADDR> mac <MACADDR>`**.

***EOS Release 4.23.1F*:**

Added support for **`clear address locking lease`** command.

***EOS Release 4.23.0F*:**

- Initial release.
- Supports IPv4 address locking.

12.5.1.2 IPv4 Locking Configuration Commands

Configure IP Locking commands in the ***address locking*** configuration mode.

Example

```
switch# configure
switch(config)# address locking
switch(config-address-locking)#
```

To enable IP Locking, you must configure

- the DHCP servers from which hosts are expected to acquire leases. IP Locking communicates with these DHCP servers to learn the IP addresses that must be authorized on the switch.

```
switch(config-address-locking)# dhcp server ipv4 10.1.1.1
switch(config-address-locking)# dhcp server ipv4 10.30.1.3
```

- a local L3 interface to communicate with the DHCP server(s). This could be the management interface, a routed interface, or a Switch Virtual Interface (SVI). This interface requires a valid IP address assigned, routable to the configured DHCP server(s), and can reside in non-default VRF. The interface IP is used as the source IP in the switch's packets to the DHCP server.

```
switch# configure
switch(config)# interface Vlan2160
switch(config-if-Vl2160)# ip address 10.10.1.2/16

switch# configure
switch(config)# address locking
switch(config-address-locking)# local-interface Vlan2160
```

To activate IP Locking on ports connected to clients, IP Locking must be enabled in the interface configuration mode. Running this command only activates IPv4 Locking and overrides the previous configuration for the interface.

Example

```
switch(config)# interface Ethernet27/1
switch(config-if-Et27/1)# address locking ipv4
```

It is possible to disable IP Locking using the **disabled** command in address-locking mode. This turns off the feature and allows a host to use any IP address, authorized or unauthorized, on any port.

Example

```
switch# configure
switch(config)# address locking
switch(config-address-locking)# disabled
```

12.5.1.2.1 Clear Commands

The **clear address locking lease...** command removes the lease from hardware. This enable mode command removes lease bindings at different granularities.

- The **clear address locking lease ipv4 V4ADDR** command removes a single lease associated with an IPv4 address.
- The **clear address locking lease ipv6 V6ADDR** removes a single lease associated with an IPv6 address.
- The **clear address locking lease intf ethernet slot** removes all leases associated with the specified interface.
- The **clear address locking lease all** removes all leases on the switch.

12.5.1.2.2 Static Lease Commands

The **lease mac** command within address locking configuration mode installs a lease into hardware for the configured IP address on the interface the configured MAC address is associated with. If the MAC address is not in the MAC table or the MAC address is on an interface that is not configured with IP Locking feature, the lease is not installed until the MAC address is added to an interface that is configured with IP Locking.



Note: Any lease from the DHCP server that matches either the same IP or MAC as a statically configured lease is removed from the switch.

```
switch# configure
switch(config)# address locking
switch(config-address-locking)# lease 1.1.1.1 mac a.b.c
```

12.5.1.2.3 Locked Address Expiration Commands

The IP addresses remain authorized and installed even after their corresponding MAC addresses have aged out. IP Locking, by default, deauthorizes leases after their corresponding MAC addresses age out. The **locked-address expiration mac disabled** command configures IP Locking to keep leases installed, even after their corresponding MAC addresses have aged out.

Example

```
switch# configure
switch(config)# address locking
```

```
switch(config-address-locking)# locked-address expiration mac disabled
```

12.5.1.2.4 IP Locking Counters

The **show address locking counters** command displays DHCP lease query messages sent, received, and dropped. There are two sets of counters: first, the number of packets sent to and received from each DHCP server; and second, the number of packets sent and received for each locked interface. There are separate counters for the different kinds of messages communicated between the switch and the DHCP server.

Example

```
switch# show address locking counters
                  Lease Active Lease Unknown Lease Unassigned
DHCP Server Query  Rcvd   Drop   Rcvd   Drop Rcvd     Drop    Unknown
----------- ----- ----- ------ ------ ------ -------- ------- -------
80.80.80.80 32860  8002 34     8001   32     13423     134    3234


Interface Query Lease Active Lease Unknown Lease Unassigned
--------- ----- ------------ -------------- -----------------
Ethernet2  1747 1234         189            324
```

The **clear address locking counters** command resets all the counters associated with IP Locking to zero.

12.5.1.3 IPv6 Locking

12.5.1.3.1 IPv6 Locking Configuration Commands

To enable IPv6 locking, you must:

Disable enforcement of IPv6 address locking.

```
switch# configure
switch(config)# address locking
switch(config-address-locking)# locked-address ipv6 enforcement disabled
```

To activate IPv6 Locking on ports connected to clients, IPv6 Locking must be enabled in the interface configuration mode. Running this command only activates IPv6 Locking and overrides the previous configuration for the interface.

Example

```
switch(config)# interface ethernet 27/1
switch(config-if-Et27/1)# address locking ipv6
```

Both IPv4 and IPv6 locking can be activated on a port by running commands similar to the following:

```
switch(config)# interface ethernet 27/1
switch(config-if-Et27/1)# address locking ipv4 ipv6
```

It is possible to disable IPv6 Locking using the **disabled** command in address-locking mode. This turns off the feature and allows a host to use any IP address, authorized or unauthorized, on any port. Note that this will also disable IP Locking as well.

Example

```
switch# configure
switch(config)# address locking
switch(config-address-locking)# disabled
```

12.5.1.3.2 Locked Address IP Enforcement Commands

The **locked-address ipv4 enforcement disabled** command disables address filtering for all ports that have IPv4 Locking enabled. This permits IPv4 packets while still keeping all other drop rules.

Example

```
switch# configure
switch(config)# address locking
switch(config-address-locking)# locked-address ipv4 enforcement
 disabled
```

The **locked-address ipv6 enforcement disabled** command disables address filtering for all ports that have IPv6 Locking enabled. This permits IPv6 packets while still keeping all other drop rules.

Example

```
switch# configure
switch(config)# address locking
switch(config-address-locking)# locked-address ipv6 enforcement
 disabled
```

12.5.1.4 Show Commands

Use the **show address locking** command to display the status of IP and IPv6 locking.

Example

```
switch# show address locking
IP Locking is active
Interface        IPv4             IPv6
---------------- -------------------   --------------------------
Ethernet27/1     yes              no (not configured)
Ethernet31/1     no (not configured)   no (not a layer 2 interface)
```

The **show address locking** command also displays the reason as to why IP Locking is not enabled for an interface. For an interface, the following priority (highest at top) is imposed on the output when IP Locking is not enabled for an interface:

- not configured
- not a Layer 2 interface
- no local interface
- no dhcp server

The **show address locking table ipv4** command displays all the DHCP leases that IP Locking knows about, whether or not those leases are installed and the interfaces on which these IP addresses are authorized.

Example

```
switch# show address locking table ipv4
IP Address     MAC Address      Interface  Installed    Expiration Time
-------------- ---------------- ---------- ------------ ---------------
10.30.4.4      ba76.a467.7ff8   Et27/1     installed     in 0:01:57
```

12.5.1.5 Limitations

The IP Locking feature contains the following limitations:

- IP Locking is supported for IPv4 but has limited functionality for IPv6.
- IP Locking works only with DHCP servers that support ***RFC 4388*** (LeaseQuery) and are configured to allow lease queries. ISC DHCPD and BlueCat are currently known servers that support LeaseQuery.
- IP Locking can only be configured on Ethernet interfaces, excluding sub-interfaces.
- IP Locking and DHCP relay cannot be configured on the same switch. When both are configured, IP Locking is disabled.
- IP Locking and DHCP snooping cannot be configured on the same switch. When both are configured, IP Locking is disabled.
- IP Locking and DHCP server cannot be configured on the same switch. When both are configured, IP Locking is disabled.
- Do not configure IP Locking and the ARP inspection feature on the same switch.
- Do not configure IP Locking and the IP source guard feature on the same switch.
- IP Locking may not immediately invalidate a lease on an access port if the host moves to another port on a different access switch.
- IP Locking supports up to 3400 hosts on the DCS-7050X3 platform, and up to 3800 hosts on the CCS-720XP platform. This scale may reduce further with other features using TCAM resources.
- IPv6 Locking currently only allows disabling address filtering for IPv6 packets while keeping all packet type specific drop rules such as ND:RA, ND:RD, and DHCP Server-to-Client.
- Some DHCP server implementations (such as ISC DHCPD) do not respond to lease query if the **fixed-address** configuration is used. Use reserved leases instead.
- CVP Endpoint Identification is not able to identify hosts connected to an IP Locking enabled switch.

12.5.1.6 IP Locking Commands

IPv4 Clear Commands

- clear address locking lease

IPv4 Static Lease Commands

- lease mac

IPv4 Lock Address Expiration Commands

- locked-address expiration mac disabled
- locked-address ipv4 enforcement disabled
- locked-address ipv6 enforcement disabled

IP Locked Show Commands

- show address locking
- show address locking counters
- show address locking table ipv4

12.5.1.6.1 clear address locking lease

Use the **clear address locking lease** command to remove lease bindings at different granularities.

Support beginning with ***EOS Release 4.23.1F***:

- The **clear address locking lease...** command removes the lease from the hardware.
  - The **clear address locking lease ipv4 V4ADDR** command removes a single lease associated with an IPv4 address.
  - The **clear address locking lease ipv6 V6ADDR** command removes a single lease associated with an IPv6 address.
  - The **clear address locking lease intf ethernet slot** command removes all leases associated with the specified interface.
  - The **clear address locking lease all** command remove all leases on the switch.

Command Mode

Address locking mode

Command Syntax

clear address locking lease [all | interface [ethernet ***slot***] | ipv4 ***V4ADDR*** | ipv6 ***V6ADDR***]

Parameters

- **all** The entire lease table.
- **interface** The interface of the lease to be cleared.
  - **ethernet *slot*** Ethernet interface slot number.
- **ipv4 *V4ADDR*** IPv4 address of the lease to be cleared.
- **ipv6 *V6ADDR*** IPv6 address of the lease to be cleared.

Example

```
switch(config-address-locking)# clear address locking lease all
```

12.5.1.6.2 lease mac

The **lease mac** command within the address locking configuration mode installs a lease into hardware for the configured IP address on the interface the configured MAC address is associated with. If the MAC address is not in the MAC table or the MAC address is on an interface that is not configured with IP Locking feature, the lease is not installed until the MAC address is added to an interface that is configured with IP Locking. The **no** and **default** forms of the command removes the lease into hardware for the configured IP address on the interface the configured MAC address is associated with.

Command Mode

Address locking configuration mode

Command Syntax

lease *V4ADDR* mac *MACADDR*

no lease *V4ADDR* mac *MACADDR*

default lease *V4ADDR* mac *MACADDR*

Parameters

- **lease** ***V4ADDR*** The lease IP address.
- **mac** ***MACADDR*** The configured mac address for static lease.

Example

```
Arista# config t
Arista(config)# address locking
Arista(config-address-locking)# lease 1.1.1.1 mac a.b.c
```

12.5.1.6.3 locked-address expiration mac disabled

IP Locking, by default, deauthorizes leases after their corresponding MAC addresses age out. Use the **locked-address expiration mac disabled** command to configure IP Locking to keep leases installed, even after their corresponding MAC addresses have aged out.

Command Mode

Address locking configuration mode

Command Syntax

locked-address expiration mac disabled

no locked-address expiration mac disabled

default locked-address expiration mac disabled

Parameters

- **expiration** Configures expiration mode for locked addresses.
- **mac** Configures deauthorizing locked addresses upon MAC aging out.
- **disabled** Disables deauthorizing locked address upon MAC aging out.

Example

```
switch# configure t
switch(config)# address locking
switch(config-address-locking)# locked-address expiration mac disabled
```

12.5.1.6.4 locked-address ipv4 enforcement disabled

The **locked-address ipv4 enforcement disabled** command disables address filtering for all ports that have IPv4 Locking enabled. This permits IPv4 packets while still keeping all other drop rules.

Command Mode

Address locking Configuration Mode

Command Syntax

locked-address ipv4 enforcement disabled

no locked-address ipv4 enforcement disabled

default locked-address ipv4 enforcement disabled

Parameters

- ipv4 IPv4 address configuration.
- **enforcement** Configure enforcement for locked addresses.
- **disabled** Disable enforcement for locked addresses.

Example

```
switch# configure
switch(config)# address locking
switch(config-address-locking)# locked-address ipv4 enforcement disabled
```

12.5.1.6.5 locked-address ipv6 enforcement disabled

The **locked-address ipv6 enforcement disabled** command disables address filtering for all ports that have IPv6 Locking enabled. This permits IPv6 packets while still keeping all other drop rules.

Command Mode

Address locking Configuration Mode

Command Syntax

locked-address ipv6 enforcement disabled

no locked-address ipv6 enforcement disabled

default locked-address ipv6 enforcement disabled

Parameters

- ipv6 IPv6 address configuration.
- **enforcement** Configure enforcement for locked addresses.
- **disabled** Disable enforcement for locked addresses.

Example

```
switch# configure
switch(config)# address locking
switch(config-address-locking)# locked-address ipv6 enforcement disabled
```

12.5.1.6.6 show address locking

Use the **show address locking** command to display the status of IP and IPv6 locking.

The **show address locking** command also displays the reason as to why IP Locking is not enabled for an interface. For an interface, the following priority (highest at top) is imposed on the output when IP Locking is not enabled for an interface:

- not configured
- not a Layer 2 interface
- no local interface
- no dhcp server

Command Mode

EXEC

Command Syntax

show address locking

Example

```
switch# show address locking
IP Locking is active
   Interface    IPv4              IPv6
-------------- ------------------   --------------------------
  Ethernet27/1  yes               no (not configured)
```

```
  Ethernet31/1  no (not configured)    no (not a layer 2 interface)
```

12.5.1.6.7 show address locking counters

The **show address locking counters** command displays DHCP lease query messages sent, received, and dropped. There are two sets of counters: first, the number of packets sent to and received from each DHCP server; and second, the number of packets sent and received for each locked interface. There are separate counters for the different kinds of messages communicated between the switch and the DHCP server.

Command Mode

EXEC

Command Syntax

show address locking counters

Related Commands

The **clear address locking counters** command resets all the counters associated with IP Locking to zero.

Example

```
switch# show address locking counters
                  Lease Active Lease Unknown Lease Unassigned
DHCP Server Query  Rcvd   Drop   Rcvd   Drop Rcvd     Drop    Unknown
----------- ----- ----- ------ ------ ------ -------- ------- -------
80.80.80.80 32860  8002 34     8001   32     13423     134    3234


Interface Query Lease Active Lease Unknown Lease Unassigned
--------- ----- ------------ ------------- ----------------
Ethernet2  1747 1234         189           324
```

12.5.1.6.8 show address locking table ipv4

Use the **show address locking table ipv4** command to display all the DHCP leases that IP Locking knows about, whether or not those leases are installed and the interfaces on which these IP addresses are authorized.

Command Mode

EXEC

Command Syntax

show address locking table ipv4 [dynamic [installed | [interface Ethernet ***slot***] | installed | interface [Ethernet [***slot***] | static [installed | interface [Ethernet ***slot***]]]

Parameters

- **dynamic** Only display the dynamic leases.
  - **installed** Only display the leases that are installes in the hardware.
  - **interface** Only display the leases belonging to a specified interface.
- **installed** Only display the leases that are installes in the hardware.
- **interface** Only display the leases belonging to a specified interface.
  - **Ethernet *slot*** Specified Ethernet sub-interface.
- **static** Only display the static leases.
  - **installed** Only display the leases that are installes in the hardware.

- **interface** Only display the leases belonging to a specified interface.
  - **Ethernet*slot*** Specified Ethernet sub-interface.

Example

```
switch# show address locking table ipv4
IP Address     MAC Address      Interface  Installed    Expiration Time
-------------- ---------------- ---------- ------------ ---------------
10.30.4.4      ba76.a467.7ff8   Et27/1     installed     in 0:01:57
```

12.6 Layer 2 Protocol Forwarding

Layer2 (L2) Protocol Forwarding on Ethernet interfaces is supported, in addition to Type-5 PW. Also, selective forwarding of certain L2 Protocol packets (tagged/untagged/all) as opposed to forwarding all LACP frames (both tagged and untagged) is allowed. The protocol list on which L2 Protocol Forwarding is supported has been extended to PAUSE, LACP, LLDP, MACSEC, STP.

12.6.1 Configuring L2 Protocol Forwarding

The following commands allow creation of a profile that allows forwarding tagged/untagged/all types of PAUSE/LACP/LLDP/MACSEC/STP/E-LMI/ ISIS/micro-BFD frames.

```
switch(config-l2-protocol)# forwarding profile abc
switch(config-l2-protocol-abc)# isis forward
switch(config-l2-protocol-abc)# macsec tagged forward
switch(config-l2-protocol-abc)# pause untagged forward
switch(config-l2-protocol-abc)# exit
```

The following commands allow application of a profile on an interface or subinterface:

```
switch(config)#int ethernet1/1
switch(config-if-Et1/1)# l2-protocol forwarding profile abc
switch(config-if-Et1/1)# exit
switch(config)# int ethernet2/1.1
switch(config-if-Et2/1.1)# l2-protocol forwarding profile def
switch(config-if-Et2/1.1)# exit
```

12.6.1.1 TCAM Profile

A specific TCAM profile is required to configure L2 Protocol Forwarding profiles except if the L2 Protocol Forwarding profile has only ‘lacp forward’ and is configured on Type-5 PW ports on Type-1 platforms (mentioned above). One such TCAM profile is as follows:

```
switch(config)# hardware tcam
switch(config-tcam)# profile l2protocolfwd
switch(config-tcam-profile-l2protocolfwd)# feature acl port ip
switch(config-tcam-feature-acl-port-ip)# sequence 45
switch(config-tcam-feature-acl-port-ip)# key size limit 160
switch(config-tcam-feature-acl-port-ip)# key field dscp dst-ip ip-frag ip-
protocol l4-dst-port l4-ops l4-src-port src-ip tcp-control ttl
switch(config-tcam-feature-acl-port-ip)# action count drop
switch(config-tcam-feature-acl-port-ip)# packet ipv4 forwarding bridged
switch(config-tcam-feature-acl-port-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-acl-port-ip)# packet ipv4 forwarding routed
 multicast
```

```
switch(config-tcam-feature-acl-port-ip)# packet ipv4 mpls ipv4 forwarding mpls
 decap
switch(config-tcam-feature-acl-port-ip)# packet ipv4 mpls ipv6 forwarding mpls
 decap
switch(config-tcam-feature-acl-port-ip)# packet ipv4 non-vxlan forwarding
 routed decap
switch(config-tcam-feature-acl-port-ip)# packet ipv4 vxlan eth ipv4 forwarding
 routed decap
switch(config-tcam-feature-acl-port-ip)# packet ipv4 vxlan forwarding bridged
 decap
switch(config-tcam-feature-acl-port-ip)# feature acl port ipv6
switch(config-tcam-feature-acl-port-ipv6)# sequence 25
switch(config-tcam-feature-acl-port-ipv6)# key field dst-ipv6 ipv6-next-header
 ipv6-traffic-class l4-dst-port l4-ops-3b l4-src-port src-ipv6-high src-ipv6-
low tcp-control
switch(config-tcam-feature-acl-port-ipv6)# action count drop
switch(config-tcam-feature-acl-port-ipv6)# packet ipv6 forwarding bridged
switch(config-tcam-feature-acl-port-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-port-ipv6)# packet ipv6 forwarding routed
 multicast
switch(config-tcam-feature-acl-port-ipv6)# packet ipv6 ipv6 forwarding routed
 decap
switch(config-tcam-feature-acl-port-ipv6)# feature acl port mac
switch(config-tcam-feature-acl-port-mac)# sequence 55
switch(config-tcam-feature-acl-port-mac)# key size limit 160
switch(config-tcam-feature-acl-port-mac)# key field dst-mac ether-type src-mac
switch(config-tcam-feature-acl-port-mac)# action count drop
switch(config-tcam-feature-acl-port-mac)# packet ipv4 forwarding bridged
switch(config-tcam-feature-acl-port-mac)# packet ipv4 forwarding routed
switch(config-tcam-feature-acl-port-mac)# packet ipv4 forwarding routed
 multicast
switch(config-tcam-feature-acl-port-mac)# packet ipv4 mpls ipv4 forwarding mpls
 decap
switch(config-tcam-feature-acl-port-mac)# packet ipv4 mpls ipv6 forwarding mpls
 decap
switch(config-tcam-feature-acl-port-mac)# packet ipv4 non-vxlan forwarding
 routed decap
switch(config-tcam-feature-acl-port-mac)# packet ipv4 vxlan forwarding bridged
 decap
switch(config-tcam-feature-acl-port-mac)# packet ipv6 forwarding bridged
switch(config-tcam-feature-acl-port-mac)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-port-mac)# packet ipv6 forwarding routed decap
switch(config-tcam-feature-acl-port-mac)# packet ipv6 forwarding routed
 multicast
switch(config-tcam-feature-acl-port-mac)# packet ipv6 ipv6 forwarding routed
 decap
switch(config-tcam-feature-acl-port-mac)# packet mpls forwarding bridged decap
switch(config-tcam-feature-acl-port-mac)# packet mpls ipv4 forwarding mpls
switch(config-tcam-feature-acl-port-mac)# packet mpls ipv6 forwarding mpls
switch(config-tcam-feature-acl-port-mac)# packet mpls non-ip forwarding mpls
switch(config-tcam-feature-acl-port-mac)# packet non-ip forwarding bridged
switch(config-tcam-feature-acl-port-mac)# feature acl subintf ip
switch(config-tcam-feature-acl-subintf-ip)# sequence 40
switch(config-tcam-feature-acl-subintf-ip)# key size limit 160
switch(config-tcam-feature-acl-subintf-ip)# key field dscp dst-ip ip-frag ip-
protocol l4-dst-port l4-ops-18b l4-src-port src-ip tcp-control ttl
switch(config-tcam-feature-acl-subintf-ip)# action count drop
switch(config-tcam-feature-acl-subintf-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-acl-subintf-ip)# feature acl subintf ipv6
switch(config-tcam-feature-acl-subintf-ipv6)# sequence 15
```

```
switch(config-tcam-feature-acl-subintf-ipv6)# key field dst-ipv6 ipv6-next-hea
der l4-dst-port l4-src-port src-ipv6-high src-ipv6-low tcp-control
switch(config-tcam-feature-acl-subintf-ipv6)# action count drop
switch(config-tcam-feature-acl-subintf-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-subintf-ipv6)# feature acl vlan ip
switch(config-tcam-feature-acl-vlan-ip)# sequence 35
switch(config-tcam-feature-acl-vlan-ip)# key size limit 160
switch(config-tcam-feature-acl-vlan-ip)# key field dscp dst-ip ip-frag ip-
protocol l4-dst-port l4-ops-18b l4-src-port src-ip tcp-control ttl
switch(config-tcam-feature-acl-vlan-ip)# action count drop
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 mpls ipv4 forwarding mpls
 decap
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 mpls ipv6 forwarding mpls
 decap
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 non-vxlan forwarding
 routed decap
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 vxlan eth ipv4 forwarding
 routed decap
switch(config-tcam-feature-acl-vlan-ip)# feature acl vlan ipv6
switch(config-tcam-feature-acl-vlan-ipv6)# sequence 10
switch(config-tcam-feature-acl-vlan-ipv6)# key field dst-ipv6 ipv6-next-header
 l4-dst-port l4-src-port src-ipv6-high src-ipv6-low tcp-control
switch(config-tcam-feature-acl-vlan-ipv6)# action count drop
switch(config-tcam-feature-acl-vlan-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-vlan-ipv6)# packet ipv6 ipv6 forwarding routed
 decap
switch(config-tcam-feature-acl-vlan-ipv6)# feature acl vlan ipv6 egress
switch(config-tcam-feature-acl-vlan-ipv6-egress)# sequence 20
switch(config-tcam-feature-acl-vlan-ipv6-egress)# key field dst-ipv6 ipv6-next-
header ipv6-traffic-class l4-dst-port l4-src-port src-ipv6-high src-ipv6-low
 tcp-control
switch(config-tcam-feature-acl-vlan-ipv6-egress)# action count drop
switch(config-tcam-feature-acl-vlan-ipv6-egress)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-vlan-ipv6-egress)# feature counter lfib
switch(config-tcam-feature-counter-lfib)# sequence 85
switch(config-tcam-feature-counter-lfib)# feature l2-protocol forwarding
switch(config-tcam-feature-l2-protocol-fowarding)# sequence 95
switch(config-tcam-feature-l2-protocol-fowarding)# key size limit 160
switch(config-tcam-feature-l2-protocol-fowarding)# key field dst-mac vlan-tag-
format
switch(config-tcam-feature-l2-protocol-fowarding)# action mirror redirect-to-c
pu set-tc
switch(config-tcam-feature-l2-protocol-fowarding)# packet non-ip forwarding
 bridge
switch(config-tcam-feature-l2-protocol-fowarding)# packet non-ip forwarding
 bridged sub-interface
switch(config-tcam-feature-l2-protocol-fowarding)# feature mirror ip
switch(config-tcam-feature-mirror-ip)# sequence 80
switch(config-tcam-feature-mirror-ip)# key size limit 160
switch(config-tcam-feature-mirror-ip)# key field dscp dst-ip ip-frag ip-
protocol l4-dst-port l4-ops l4-src-port src-ip tcp-control
switch(config-tcam-feature-mirror-ip)# action count mirror set-policer
switch(config-tcam-feature-mirror-ip)# packet ipv4 forwarding bridged
switch(config-tcam-feature-mirror-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-mirror-ip)# packet ipv4 forwarding routed multicast
switch(config-tcam-feature-mirror-ip)# packet ipv4 non-vxlan forwarding routed
 decap
switch(config-tcam-feature-mirror-ip)# feature mpls
switch(config-tcam-feature-mpls)# sequence 5
switch(config-tcam-feature-mpls)# key size limit 160
```

```
switch(config-tcam-feature-mpls)# action drop redirect set-ecn
switch(config-tcam-feature-mpls)# packet ipv4 mpls ipv4 forwarding mpls decap
switch(config-tcam-feature-mpls)# packet ipv4 mpls ipv6 forwarding mpls decap
switch(config-tcam-feature-mpls)# packet mpls ipv4 forwarding mpls
switch(config-tcam-feature-mpls)# packet mpls ipv6 forwarding mpls
switch(config-tcam-feature-mpls)# packet mpls non-ip forwarding mpls
switch(config-tcam-feature-mpls)# feature mpls pop ingress
switch(config-tcam-feature-mpls-pop-ingress)# sequence 90
switch(config-tcam-feature-mpls-pop-ingress)# feature pbr ip
switch(config-tcam-feature-pbr-ip)# sequence 60
switch(config-tcam-feature-pbr-ip)# key size limit 160
switch(config-tcam-feature-pbr-ip)# key field dscp dst-ip ip-frag ip-protocol
 l4-dst-port l4-ops-18b l4-src-port src-ip tcp-control
switch(config-tcam-feature-pbr-ip)# action count redirect
switch(config-tcam-feature-pbr-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-pbr-ip)# packet ipv4 mpls ipv4 forwarding mpls decap
switch(config-tcam-feature-pbr-ip)# packet ipv4 mpls ipv6 forwarding mpls decap
switch(config-tcam-feature-pbr-ip)# packet ipv4 non-vxlan forwarding routed
 decap
switch(config-tcam-feature-pbr-ip)# packet ipv4 vxlan forwarding bridged decap
switch(config-tcam-feature-pbr-ip)# feature pbr ipv6
switch(config-tcam-feature-pbr-ipv6)# sequence 30
switch(config-tcam-feature-pbr-ipv6)# key field dst-ipv6 ipv6-next-header l4-
dst-port l4-src-port src-ipv6-high src-ipv6-low tcp-control
switch(config-tcam-feature-pbr-ipv6)# action count redirect
switch(config-tcam-feature-pbr-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-pbr-ipv6)# feature pbr mpls
switch(config-tcam-feature-pbr-mpls)# sequence 65
switch(config-tcam-feature-pbr-mpls)# key size limit 160
switch(config-tcam-feature-pbr-mpls)# key field mpls-inner-ip-tos
switch(config-tcam-feature-pbr-mpls)# action count drop redirect
switch(config-tcam-feature-pbr-mpls)# packet mpls ipv4 forwarding mpls
switch(config-tcam-feature-pbr-mpls)# packet mpls ipv6 forwarding mpls
switch(config-tcam-feature-pbr-mpls)# packet mpls non-ip forwarding mpls
switch(config-tcam-feature-pbr-mpls)# feature qos ip
switch(config-tcam-feature-qos-ip)# sequence 75
switch(config-tcam-feature-qos-ip)# key size limit 160
switch(config-tcam-feature-qos-ip)# key field dscp dst-ip ip-frag ip-protocol
 l4-dst-port l4-ops l4-src-port src-ip tcp-control
switch(config-tcam-feature-qos-ip)# action set-dscp set-policer set-tc
switch(config-tcam-feature-qos-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-qos-ip)# packet ipv4 forwarding routed multicast
switch(config-tcam-feature-qos-ip)# packet ipv4 mpls ipv4 forwarding mpls decap
switch(config-tcam-feature-qos-ip)# packet ipv4 mpls ipv6 forwarding mpls decap
switch(config-tcam-feature-qos-ip)# packet ipv4 non-vxlan forwarding routed
 decap
switch(config-tcam-feature-qos-ip)# feature qos ipv6
switch(config-tcam-feature-qos-ipv6)# sequence 70
switch(config-tcam-feature-qos-ipv6)# key field dst-ipv6 ipv6-next-header ipv6-
traffic-class l4-dst-port l4-src-port src-ipv6-high src-ipv6-low
switch(config-tcam-feature-qos-ipv6)# action set-dscp set-policer set-tc
switch(config-tcam-feature-qos-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-qos-ipv6)# feature tunnel vxlan
switch(config-tcam-feature-tunnel-vxlan)# sequence 50
switch(config-tcam-feature-tunnel-vxlan)# key size limit 160
switch(config-tcam-feature-tunnel-vxlan)# packet ipv4 vxlan eth ipv4 forwarding
 routed decap
switch(config-tcam-feature-tunnel-vxlan)# packet ipv4 vxlan forwarding bridged
 decap
```

For ISIS protocol forwarding, “snoop” action has to be present in the TCAM profile for l2-protocol forwarding feature as follows:

```
switch(config-tcam)# feature l2-protocol forwarding
switch(config-tcam-l2-protocol forwarding)# sequence 95
switch(config-tcam-l2-protocol forwarding)# key size limit 160
switch(config-tcam-l2-protocol forwarding)# key field dst-mac vlan-tag-format
switch(config-tcam-l2-protocol forwarding)# action mirror redirect-to-cpu set-
tc snoop
switch(config-tcam-l2-protocol forwarding)# packet non-ip forwarding bridged
switch(config-tcam-l2-protocol forwarding)# packet non-ip forwarding bridged
 sub-interface
```

For the special case of micro-BFD protocol forwarding the following TCAM profile needs to be applied:

```
switch(config)# hardware tcam
switch(config-tcam)# profile l2protocolfwd-bfd-rfc-7130
switch(config-tcam-profile l2protocolfwd-bfd-rfc-7130)# feature acl port ip
 egress mpls-tunnelled-match
switch(config-tcam-feature-acl-port-ip-egress-mpls-tunnelled-match)# sequence
 95
switch(config-tcam-feature-acl-port-ip-egress-mpls-tunnelled-match)# feature
 acl port ipv6 egress
switch(config-tcam-feature-acl-port-ipv6-egress)# sequence 105
switch(config-tcam-feature-acl-port-ipv6-egress)# key field dst-ipv6 ipv6-next-
header ipv6-traffic-class l4-dst-port l4-src-port src-ipv6-high src-ipv6-low
 tcp-control
switch(config-tcam-feature-acl-port-ipv6-egress)# action count drop mirror
switch(config-tcam-feature-acl-port-ipv6-egress)# packet ipv6 forwarding
 bridged
switch(config-tcam-feature-acl-port-ipv6-egress)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-port-ipv6-egress)# feature acl port mac
switch(config-tcam-feature-acl-port-mac)# sequence 55
switch(config-tcam-feature-acl-port-mac)# key size limit 160
switch(config-tcam-feature-acl-port-mac)# key field dst-mac ether-type src-mac
switch(config-tcam-feature-acl-port-mac)# action count drop mirror
switch(config-tcam-feature-acl-port-mac)# packet ipv4 forwarding bridged
switch(config-tcam-feature-acl-port-mac)# packet ipv4 forwarding routed
switch(config-tcam-feature-acl-port-mac)# packet ipv4 forwarding routed
 multicast
switch(config-tcam-feature-acl-port-mac)# packet ipv4 mpls ipv4 forwarding mpls
 decap
switch(config-tcam-feature-acl-port-mac)# packet ipv4 mpls ipv6 forwarding mpls
 decap
switch(config-tcam-feature-acl-port-mac)# packet ipv4 non-vxlan forwarding
 routed decap
switch(config-tcam-feature-acl-port-mac)# packet ipv4 vxlan forwarding bridged
 decap
switch(config-tcam-feature-acl-port-mac)# packet ipv6 forwarding bridged
switch(config-tcam-feature-acl-port-mac)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-port-mac)# packet ipv6 forwarding routed decap
switch(config-tcam-feature-acl-port-mac)# packet ipv6 forwarding routed
 multicast
switch(config-tcam-feature-acl-port-mac)# packet ipv6 ipv6 forwarding routed
 decap
switch(config-tcam-feature-acl-port-mac)# packet mpls forwarding bridged decap
switch(config-tcam-feature-acl-port-mac)# packet mpls ipv4 forwarding mpls
switch(config-tcam-feature-acl-port-mac)# packet mpls ipv6 forwarding mpls
switch(config-tcam-feature-acl-port-mac)# packet mpls non-ip forwarding mpls
switch(config-tcam-feature-acl-port-mac)# packet non-ip forwarding bridged
switch(config-tcam-feature-acl-port-mac)# feature acl subintf ip
```

```
switch(config-tcam-feature-acl-subintf-ip)# sequence 40
switch(config-tcam-feature-acl-subintf-ip)# key size limit 160
switch(config-tcam-feature-acl-subintf-ip)# key field dscp dst-ip ip-frag ip-
protocol l4-dst-port l4-ops-18b l4-src-port src-ip tcp-control ttl
switch(config-tcam-feature-acl-subintf-ip)# action count drop
switch(config-tcam-feature-acl-subintf-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-acl-subintf-ip)# feature acl subintf ipv6
switch(config-tcam-feature-acl-subintf-ipv6)# sequence 15
switch(config-tcam-feature-acl-subintf-ipv6)# key field dst-ipv6 ipv6-next-hea
der l4-dst-port l4-src-port src-ipv6-high src-ipv6-low tcp-control
switch(config-tcam-feature-acl-subintf-ipv6)# action count drop
switch(config-tcam-feature-acl-subintf-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-subintf-ipv6)# feature acl vlan ip
switch(config-tcam-feature-acl-vlan-ip)# sequence 35
switch(config-tcam-feature-acl-vlan-ip)# key size limit 160
switch(config-tcam-feature-acl-vlan-ip)# key field dscp dst-ip ip-frag ip-
protocol l4-dst-port l4-ops-18b l4-src-port src-ip tcp-control ttl
switch(config-tcam-feature-acl-vlan-ip)# action count drop
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 mpls ipv4 forwarding mpls
 decap
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 mpls ipv6 forwarding mpls
 decap
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 non-vxlan forwarding
 routed decap
switch(config-tcam-feature-acl-vlan-ip)# packet ipv4 vxlan eth ipv4 forwarding
 routed decap
switch(config-tcam-feature-acl-vlan-ip)# feature acl vlan ipv6
switch(config-tcam-feature-acl-vlan-ipv6)# sequence 10
switch(config-tcam-feature-acl-vlan-ipv6)# key field dst-ipv6 ipv6-next-header
 l4-dst-port l4-src-port src-ipv6-high src-ipv6-low tcp-control
switch(config-tcam-feature-acl-vlan-ipv6)# action count drop
switch(config-tcam-feature-acl-vlan-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-vlan-ipv6)# packet ipv6 ipv6 forwarding routed
 decap
switch(config-tcam-feature-acl-vlan-ipv6)# feature acl vlan ipv6 egress
switch(config-tcam-feature-acl-vlan-ipv6-egress)# sequence 20
switch(config-tcam-feature-acl-vlan-ipv6-egress)# key field dst-ipv6 ipv6-next-
header ipv6-traffic-class l4-dst-port l4-src-port src-ipv6-high src-ipv6-low
 tcp-control
switch(config-tcam-feature-acl-vlan-ipv6-egress)# action count drop mirror
switch(config-tcam-feature-acl-vlan-ipv6-egress)# packet ipv6 forwarding
 bridged
switch(config-tcam-feature-acl-vlan-ipv6-egress)# packet ipv6 forwarding routed
switch(config-tcam-feature-acl-vlan-ipv6-egress)# feature counter lfib
switch(config-tcam-feature-counter-lfib)# sequence 85
switch(config-tcam-feature-counter-lfib)# feature forwarding-destination mpls
switch(config-tcam-feature-forwarding-destination-mpls)# sequence 100
switch(config-tcam-feature-forwarding-destination-mpls)# feature l2-protocol
 forwarding
switch(config-tcam-feature-forwarding-l2-protocol-forwarding)# sequence 95
switch(config-tcam-feature-forwarding-l2-protocol-forwarding)# key size limit
 160
switch(config-tcam-feature-forwarding-l2-protocol-forwarding)# key field dst-
mac vlan-tag-format
switch(config-tcam-feature-forwarding-l2-protocol-forwarding)# action mirror
 redirect-to-cpu set-tc
switch(config-tcam-feature-forwarding-l2-protocol-forwarding)# packet ipv4
 forwarding bridged
switch(config-tcam-feature-forwarding-l2-protocol-forwarding)# packet ipv6
 forwarding bridged
```

```
switch(config-tcam-feature-forwarding-l2-protocol-forwarding)# packet non-ip
 forwarding bridged
switch(config-tcam-feature-forwarding-l2-protocol-forwarding)# feature mirror
 ip
switch(config-tcam-feature-mirror-ip)# sequence 80
switch(config-tcam-feature-mirror-ip)# key size limit 160
switch(config-tcam-feature-mirror-ip)# key field dscp dst-ip ip-frag ip-
protocol l4-dst-port l4-ops l4-src-port src-ip tcp-control
switch(config-tcam-feature-mirror-ip)# action count mirror set-policer
switch(config-tcam-feature-mirror-ip)# packet ipv4 forwarding bridged
switch(config-tcam-feature-mirror-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-mirror-ip)# packet ipv4 forwarding routed multicast
switch(config-tcam-feature-mirror-ip)# packet ipv4 non-vxlan forwarding routed
 decap
switch(config-tcam-feature-mirror-ip)# feature mpls
switch(config-tcam-feature-mpls)# sequence 5
switch(config-tcam-feature-mpls)# key size limit 160
switch(config-tcam-feature-mpls)# action drop redirect set-ecn
switch(config-tcam-feature-mpls)# packet ipv4 mpls ipv4 forwarding mpls decap
switch(config-tcam-feature-mpls)# packet ipv4 mpls ipv6 forwarding mpls decap
switch(config-tcam-feature-mpls)# packet mpls ipv4 forwarding mpls
switch(config-tcam-feature-mpls)# packet mpls ipv6 forwarding mpls
switch(config-tcam-feature-mpls)# packet mpls non-ip forwarding mpls
switch(config-tcam-feature-mpls)# feature mpls pop ingress
switch(config-tcam-feature-mpls-pop-ingress)# sequence 90
switch(config-tcam-feature-mpls-pop-ingress)# feature pbr ip
switch(config-tcam-feature-pbr-ip)# sequence 60
switch(config-tcam-feature-pbr-ip)# key size limit 160
switch(config-tcam-feature-pbr-ip)# key field dscp dst-ip ip-frag ip-protocol
 l4-dst-port l4-ops-18b l4-src-port src-ip tcp-control
switch(config-tcam-feature-pbr-ip)# action count redirect
switch(config-tcam-feature-pbr-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-pbr-ip)# packet ipv4 mpls ipv4 forwarding mpls decap
switch(config-tcam-feature-pbr-ip)# packet ipv4 mpls ipv6 forwarding mpls decap
switch(config-tcam-feature-pbr-ip)# packet ipv4 non-vxlan forwarding routed
 decap
switch(config-tcam-feature-pbr-ip)# packet ipv4 vxlan forwarding bridged decap
switch(config-tcam-feature-pbr-ip)# feature pbr ipv6
switch(config-tcam-feature-pbr-ipv6)# sequence 30
switch(config-tcam-feature-pbr-ipv6)# key field dst-ipv6 ipv6-next-header l4-
dst-port l4-src-port src-ipv6-high src-ipv6-low tcp-control
switch(config-tcam-feature-pbr-ipv6)# action count redirect
switch(config-tcam-feature-pbr-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-pbr-ipv6)# feature pbr mpls
switch(config-tcam-feature-pbr-mpls)# sequence 65
switch(config-tcam-feature-pbr-mpls)# key size limit 160
switch(config-tcam-feature-pbr-mpls)# key field mpls-inner-ip-tos
switch(config-tcam-feature-pbr-mpls)# action count drop redirect
switch(config-tcam-feature-pbr-mpls)# packet mpls ipv4 forwarding mpls
switch(config-tcam-feature-pbr-mpls)# packet mpls ipv6 forwarding mpls
switch(config-tcam-feature-pbr-mpls)# packet mpls non-ip forwarding mpls
switch(config-tcam-feature-pbr-mpls)# feature qos ip
switch(config-tcam-feature-qos-ip)# sequence 75
switch(config-tcam-feature-qos-ip)# key size limit 160
switch(config-tcam-feature-qos-ip)# key field dscp dst-ip ip-frag ip-protocol
 l4-dst-port l4-ops l4-src-port src-ip tcp-control
switch(config-tcam-feature-qos-ip)# action set-dscp set-policer set-tc
switch(config-tcam-feature-qos-ip)# packet ipv4 forwarding routed
switch(config-tcam-feature-qos-ip)# packet ipv4 forwarding routed multicast
switch(config-tcam-feature-qos-ip)# packet ipv4 mpls ipv4 forwarding mpls decap
switch(config-tcam-feature-qos-ip)# packet ipv4 mpls ipv6 forwarding mpls decap
```

```
switch(config-tcam-feature-qos-ip)# packet ipv4 non-vxlan forwarding routed
 decap
switch(config-tcam-feature-qos-ip)# feature qos ipv6
switch(config-tcam-feature-qos-ipv6)# sequence 70
switch(config-tcam-feature-qos-ipv6)# key field dst-ipv6 ipv6-next-header ipv6-
traffic-class l4-dst-port l4-src-port src-ipv6-high src-ipv6-low
switch(config-tcam-feature-qos-ipv6)# action set-dscp set-policer set-tc
switch(config-tcam-feature-qos-ipv6)# packet ipv6 forwarding routed
switch(config-tcam-feature-qos-ipv6)# feature tunnel vxlan
switch(config-tcam-feature-tunnel-vxlan)# sequence 50
switch(config-tcam-feature-tunnel-vxlan)# key size limit 160
switch(config-tcam-feature-tunnel-vxlan)# packet ipv4 vxlan eth ipv4 forwarding
 routed decap
switch(config-tcam-feature-tunnel-vxlan)# packet ipv4 vxlan forwarding bridged
 decap
```

Once **l2protocolfwd** or **l2protocolfwd-bfd-rfc-7130** tcam profile is created, the profile needs to be applied to the switch using the following command under config mode:

```
switch(config)# hardware tcam
switch(config-tcam)# system profile l2protocolfwd
```

Consumption of TCAM

The following table describes how TCAM is consumed when the **same** L2 Protocol Forwarding profile is applied:

Level of application	Type-1 platforms	Type-2 platforms
Front panel ports	Separate TCAM entries for each port	Same set of TCAM entries for ports on the same fap **and** core
Subinterfaces	Same set of TCAM entries for subinterfaces on the same fap	Same set of TCAM entries for subinterfaces on the same fap **and** core

12.6.2 L2 Protocol Forwarding Limitations

1. A maximum of ***31*** distinct L2 Protocol Forwarding profiles are applied across multiple subinterfaces on the same FAP.
2. A maximum of ***31*** distinct L2 Protocol Forwarding profiles are applied across multiple subinterfaces, and front panel ports on the same FAP on Type-2 platforms.
3. The feature is not supported for subinterfaces on Type-1 platforms if any of the below features are present in the TCAM Profile:
   - feature pbr subintf ip
   - feature pbr subintf ipv6
   - feature pbr subintf mpls

12.6.3 L2 Protocol Forwarding Show commands

The show l2-protocol forwarding interface displays the L2 protocol forwarding profile configuration corresponding to the interfaces or subinterfaces.

```
switch(config)# show l2-protocol forwarding interface

Interface     Profile
------------- -------
```

```
Ethernet1/1   abc
Ethernet2/1.1 def


switch> show l2-protocol forwarding interface Ethernet1/1

Interface   Profile
----------- --------
Ethernet1/1 abc
```

The following show commands displays all packets forwarding behaviour on an interface or a subinterface:

```
switch(config)# show l2-protocol forwarding interface detail



Tagging Types: T: tagged U: untagged
Actions: F: forward
                     BFD RFC-7130 E-LMI ISIS LACP LLDP MACSEC PAUSE STP
Interface     Profile   T     U    T  U  T U  T U  T U  T   U  T  U  T U
------------- ------- ----- ------ -- -- - -- - -- - -- -- --- -- -- - -


Ethernet1/1   abc       -     -    -  -  F  F - -  - -  F   -  -  F  - -
Ethernet2/2.1 def       -     -    -  -  - -  - -  - -  -   -  -  -  - -

switch> show l2-protocol forwarding interface Ethernet1/1 detail
Tagging Types: T: tagged U: untagged


Actions: F: forward


                     BFD RFC-7130 E-LMI ISIS LACP LLDP MACSEC PAUSE STP
Interface     Profile   T     U    T  U  T U  T U  T U  T   U  T  U  T U
------------- ------- ----- ------ -- -- - -- - -- - -- -- --- -- -- - -


Ethernet1/1   abc       -     -    -  -  F F  - -  - -  F   -  -  F  - -

switch> show l2-protocol forwarding interface Ethernet2/2.1 detail
Tagging Types: T: tagged U: untagged


Actions: F: forward

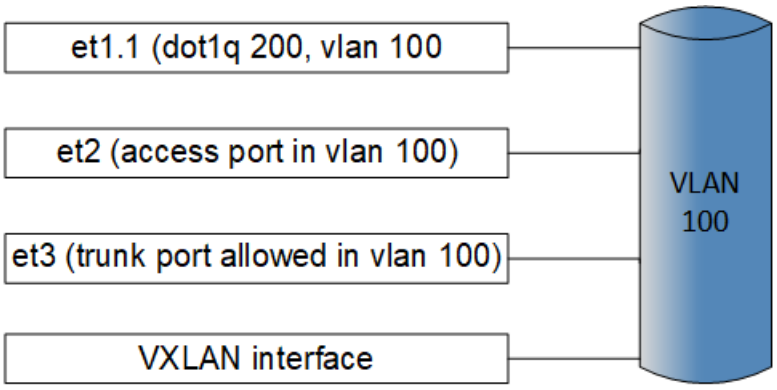

                     BFD RFC-7130 E-LMI ISIS LACP LLDP MACSEC PAUSE STP
Interface     Profile   T     U    T  U  T U  T U  T U  T   U  T  U  T U
------------- ------- ----- ------ -- -- - -- - -- - -- -- --- -- -- - -


Ethernet2/2.1 def       -     -    -  -  - -  - -  - -  -   -  -  -  - -
```

12.7 Layer 2 Subinterfaces

A Layer 2 subinterface is a logical bridging endpoint associated with traffic on an interface distinguished by 802.1Q tags, where each ***interface, 802.1q tag*** tuple is treated as a first-class bridging interface.

Like other types of interfaces, an L2 subinterface is a normal bridging endpoint in the bridging domain.



12.7.1 Configurations

12.7.1.1 Creating a Layer 2 Subinterface

Complete the following steps to configure a Layer 2 (L2) subinterface on an Arista switch:

1. Configure the parent interface to be a routed port.

```
switch(config)# interface et1
switch(config-if-Et1)# no switchport
```

2. Create a subinterface on the parent interface (***et1.1***), assign 802.1q encapsulation (***vlan 100***), and assign the forwarding VLAN ID (***vlan 200***).

```
switch(config-if-Et1)# interface et1.1
switch(config-if-Et1.1)# encapsulation dot1q vlan 100
switch(config-if-Et1.1)# vlan id 200
! VLAN does not exist. Creating vlan 200
```

3. An alternative to configuring a forwarding VLAN id is to use VLAN name (***office***).

```
switch(config)# vlan 200
switch(config-vlan-200)# name office
switch(config-vlan-200)# int et1.2
switch(config-if-Et1.2)# encapsulation dot1q vlan 101
switch(config-if-Et1.2)# vlan name office
```

4. Now subinterfaces ***et1.1*** and ***et1.2*** have been created and added to ***vlanVLAN 200***.

```
switch# show interface et1.1-2 status
Port      Name   Status       Vlan     Duplex Speed  Type                 Flags Encapsulation
Et1.1            connected    200      full   10G    dot1q-encapsulation  100

Et1.2            connected    200      full   10G    dot1q-encapsulation  101
```

12.7.1.2 MAC Address on Layer 2 Subinterface

MAC addresses can either be statically configured or dynamically assigned behind Layer 2 (L2) subinterfaces.

```
switch(config)# mac address-table static 0000.000a.000a vlan 200
 interface et1.1
```

```
switch# show mac address-table interface et1.1-2
          Mac Address Table
------------------------------------------------------------

Vlan  Mac Address     Type     Ports   Moves  Last Move
----  -----------     ----     -----   -----  ---------
 200  0000.000a.000a  STATIC   Et1.1
 200  0000.000b.000b  DYNAMIC  Et1.2   1      0:00:06 ago
Total Mac Addresses for this criterion: 2
```

MAC address learning can be enabled or disabled on an L2 subinterface using the following commands:

In the following example, the **`show interface ethernet1.1 switchport`** command has this ***running-config***:

```
switch(config-if-Et1.1)# show interface ethernet1.1 switchport
Name: Et1.1
Switchport: Enabled
Administrative Mode: tunnel
Operational Mode: tunnel
MAC Address Learning: disabled
Dot1q ethertype/TPID: 0x8100 (active)
Dot1q VLAN Tag: Allowed
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Static Trunk Groups:
Dynamic Trunk Groups:
Source interface filtering: enabled
VLAN forwarding mode: allowedVlansOnly
```

To enable MAC address learning, use the **`no mac address learning disabled`** command:

```
switch(config-if-Et1.1)# no mac address learning disabled
```

12.7.1.3 QoS Feature

Supported QoS features are:

Shaping

After creating an L2 subinterface, you can configure a shape rate (in Kbps) on the sub-interface. For example, the shape rate is configured to ***50000000*** Kbps.

```
switch(config-if-Et1.1)# shape rate 50000000
```

The configuration of non-default shape rate results in the allocation of dedicated virtual output queues (VOQ) for the subinterface. Each subinterface allocates four (4) VOQs. Different TC traffic goes to the VOQ according to the following mapping:

TC6-7 :	VOQ3
TC4-5 :	VOQ2
TC2-3 :	VOQ1
TC0-1 :	VOQ0

VOQ3 is in strict-priority mode to the other VOQs.

VOQ2, VOQ1, and VOQ0 are in WRR with a static credit ratio 2:3:6 (higher ratio implies more credits).

The subinterface inherits the trust mode of the parent interface.

Before ***EOS Release 4.24.2F***, shaping is supported only on L2 subinterfaces of which the parents are Ethernet interfaces, for example, ***Et1.1***. Beginning with ***EOS Release 4.24.2F***, shaping on L2 subinterface over a parent interface which is port-channel (for example, ***Po1.1***), is supported.

Guaranteed Bandwidth

After configuring shaping on an L2 sub-interface, user can configure a guaranteed bandwidth (in Kbps or percent) on the subinterface using the **`bandwidth guaranteed`** command.

```
switch(config-if-Et1.1)# bandwidth guaranteed 10000000
switch(config-if-Et1.1)# bandwidth guaranteed percent 10
```

Policing

For policing to work on the L2 subinterface, you must switch to the QoS profile.

Example

```
switch(config)# hardware tcam
switch(config-hw-tcam)# system profile qos

Sample Policy-map Configuration:

switch(config)# ip access-list a1
switch(config-acl-a1)# statistics per-entry
switch(config-acl-a1)# 10 permit ip any any

switch(config)# class-map type qos match-any c1
switch(config-cmap-qos-c1)# match ip access-group a1

switch(config)# class-map type qos match-any c2
switch(config-cmap-qos-c2)# match vlan 100 0xfff

switch(config)# ipv6 access-list a1
switch(config-ipv6-acl-a1)# statistics per-entry
switch(config-ipv6-acl-a1)# 10 permit ipv6 any any
```

```
switch(config)# class-map type qos match-any c3
switch(config-cmap-qos-c3)# match ipv6 access-group a1

switch(config)# policy-map type quality-of-service p1
switch(config-pmap-quality-of-service-p1)# class c1
switch(config-pmap-quality-of-service-p1-c1)# police cir 10 Mbps bc 100000
 bytes
    exit
exit
```

After you create an L2 subinterface, you can configure a policy-map on the sub-interface, similar to the following example.

```
switch(config-if-Et1.1)# service-policy type qos input p1
```

Interface Counters

To enable the hardware features for counting packets on L2 subinterfaces ingress and/or egress, use the **`hardware counter feature`** command, similar to the following example. In the example, subinterface ***layer2*** is enabled for ingress then enabled for egress.

Example

```
switch(config)# hardware counter feature subinterface in layer2
switch(config)# hardware counter feature subinterface out layer2
```

To display the L2 subinterface counters, use the **`show interface counters`** command similar to the following example. In the example, subinterface ***et1.1*** is configured to be displayed.

```
switch# show interfaces et1.1 counters

Port         InOctets      InPkts
Et1.1        0             0

Port          OutOctets    OutPkts
Et1.1         0             0
```

To clear all of the interface counters, use the **`clear counters`** command similar to the following example:

```
switch# clear counters
```

To learn counters for a specific L2 interface, use the **`clear counters`** command, and specify the L2 subinterface to clear, similar to the following example. In the example, L2 subinterface ***et1.1*** is to be cleared.

```
switch# clear counters et1.1
```

12.7.1.4 Limitations

The following limitations apply to the Layer 2 subinterface feature:

- A total of ***256*** Layer 2 subinterfaces with shaping are supported across the entire switch and they can be distributed across any number of Ethernet ports.
- When a shape rate is configured on an L2 subinterface over a parent interface which is port-channel (example: ***Po1.1***), traffic load-balancing is disabled and is directed to a selected port-channel member. Also, the bandwidth of the port-channel subinterface will be equal to the selected member. However, the **`show interface`** command continues to show the bandwidth of the port-channel which is incorrect.
- After configuring a shape rate on an L2 subinterface, the L2 subinterface must be flapped by using the **`shut`**” and **`no shut`** commands.

- Shaping of BUM traffic on L2 subinterfaces is supported only with "***ingress replication***".
- Layer 3 forwarding through SVIs is not supported.
- Routing through the parent interface of a subinterface is not supported.
- Control plane processing, such as IGMP snooping and STP BPDU is not supported.
- When IGMP protocol packets are expected to be forwarded on L2 subinterfaces, then IGMP snooping must be disabled globally on the entire switch using the **`no ip igmp snooping`** command. When IGMP snooping is configured on any VLAN, then IGMP protocol packets are discarded by L2 subinterfaces.
- Double tagged packets arriving on L2 subinterfaces with a single **`encapsulation dot1q vlan <outer_vid>`** command configured will match on the outer VLAN tag, and have only the outer VLAN tags terminated.
- Configuration of double tagged L2 subinterfaces through the **`encapsulation dot1q vlan <outer_vid> inner <inner_vid>`** command is not supported.
- Mixing of shaped and non-shaped subinterfaces under the same parent interface is not supported.
- Traffic classification on ingress traffic to l2 subinterface is disabled by default. To enable this feature, configure using the **`qos trust cos`** command on the parent interface.
- L2 subinterfaces are not supported in an MLAG environment.

12.7.2 QoS Show Commands

Use the **`show interfaces status`** command to display the subinterface status.

Example

```
switch# show interfaces status sub-interfaces
Port     Name   Status       Vlan     Duplex Speed  Type                Flags
 Encapsulation
Et1.1           connected    200      full   10G    dot1q-encapsulation       100
Et1.2           connected    200      full   10G    dot1q-encapsulation       101
```

Use the **`show vlan`** command to display the VLAN membership. In the following example, ***vlan 200*** is configured to be displayed.

Example

```
switch# show vlan 200
VLAN  Name                  Status    Ports
----- --------------------- --------- -------------------
200   office                active    Et1.1, Et1.2, Et5
```

Use the **`show qos interface`** command to display the QoS configuration on an L2 subinterface. In the following example, QoS subinterface ***Ethernet 1.1*** is configured to be displayed.

Example

```
switch# show qos interface Ethernet 1.1
Ethernet1.1:

   Trust Mode: DSCP
   Default COS: 0
   Default DSCP: 0
```

```
   Port shaping rate: 50625 / 50000 kbps
```

Use the **`show interface counters`** with the queue keyword to display the L2 subinterface counters. For example subinterface ***Ethernet 1.1*** is configured to display the L2 subinterface counters.

Example

```
switch# show interface Ethernet 1.1 counters queue
Aggregate VoQ Counters
Egress    Traffic    Pkts      Octets      DropPkts  DropOctets
Port      Class
-----------------------------------------------------------
Et1.1     TC0-1      0         0           0         0
Et1.1     TC2-3      0         0           0         0
Et1.1     TC4-5      0         0           0         0
Et1.1     TC6-7      460266    276159600   109316    65589600
```

Use the **`show mac address-table`** command to display the MAC address on L2 subinterfaces. For example, subinterfaces ***Et1.1*** and ***Et1.2*** are configured to be displayed.

Example

```
switch# show mac address-table interface et1.1-2
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
 200    0000.000a.000a    STATIC      Et1.1
 200    0000.000b.000b    DYNAMIC     Et1.2      1       0:00:16 ago
Total Mac Addresses for this criterion: 2
```

Chapter 13

Layer 3 Configuration

This chapter covers the following Layer 3 sections:

- IPv4
- IPv6
- Ingress and Egress Per-Port for IPv4 and IPv6 Counters
- ACLs and Route Maps
- VRRP and VARP
- DirectFlow
- Decap Groups
- Nexthop Groups
- Global Knob to Set MTU for all Layer 3 Interfaces
- Support for Layer 3 MTU on 7280R3/7500R3/7800R3
- Segment Security

13.1 IPv4

Arista switches support Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) for routing packets across network boundaries. This section describes Arista's implementation of IPv4 and includes these topics:

- IPv4 Addressing
- IPv4 Routing
- IPv4 Multicast Counters
- Route Management
- IPv4 Route Scale
- IP Source Guard
- DHCP Server
- DHCP Relay Global Configuration Mode
- DHCP Relay Across VRF
- DHCP Relay in VXLAN EVPN
- DHCP Snooping with Bridging
- TCP MSS Clamping
- IPv4 GRE Tunneling
- GRE Tunneling Support
- BfRuntime to use non-default VRFs
- IPv4 Commands

13.1.1 IPv4 Addressing

Each IPv4 network device is assigned a 32-bit IP address that identifies its network location. These sections describe IPv4 address formats, data structures, configuration tasks, and display options:

- IPv4 Address Formats
- IPv4 Address Configuration
- Address Resolution Protocol (ARP)

- Displaying ARP Entries

13.1.1.1 IPv4 Address Formats

IPv4 addresses are composed of 32 bits, expressed in dotted decimal notation by four decimal numbers, each ranging from **0** to **255**. A subnet is identified by an IP address and an address space defined by a routing prefix. The switch supports the following subnet formats:

- **IP address and subnet mask:** The subnet mask is a 32-bit number (dotted decimal notation) that specifies the subnet address space. The subnet address space is calculated by performing an AND operation between the IP address and subnet mask.
- **IP address and wildcard mask:** The wildcard mask is a 32-bit number (dotted decimal notation) that specifies the subnet address space. Wildcard masks differ from subnet masks in that the bits are inverted. Some commands use wildcard masks instead of subnet masks.
- **CIDR notation:** CIDR notation specifies the scope of the subnet space by using a decimal number to identify the number of leading ones in the routing prefix. When referring to wildcard notation, CIDR notation specifies the number of leading zeros in the routing prefix.

Examples

- These subnets (subnet mask and CIDR notation) are calculated identically:

```
10.24.154.13 255.255.255.0
10.24.154.13/24
```

- The defined space includes all addresses between ***10.24.154.0*** and ***10.24.154.255***. These subnets (wildcard mask and CIDR notation) are calculated identically:

```
124.17.3.142 0.0.0.15
124.17.3.142/28
```

  The defined space includes all addresses between ***124.17.3.128*** and ***124.17.3.143***.

13.1.1.2 IPv4 Address Configuration

Assigning an IPv4 Address to an Interface

The ip address command specifies the IPv4 address of an interface and the mask for the subnet to which the interface is connected.

Example

These commands configure an IPv4 address with subnet mask for ***VLAN 200***:

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ip address 10.0.0.1/24
switch(config-if-Vl200)#
```

Assigning an IPv4 Class E Address to an Interface

The ipv4 routable 240.0.0.0/4 command assigns a class E addresses to an interface. When configured, the class E address traffic are routed through BGP, OSPF, ISIS, RIP, static routes and programmed to the FIB and kernel. By default, this command is disabled.

Example

- These commands configure an IPv4 Class E (***240/4***) address to an interface.

```
switch(config)# router general
switch(config-router-general)# ipv4 routable 240.0.0.0/4
```

Detecting duplicate IP Addresses on an Interface

The **ip address duplicate detection disabled** command detects any duplicate IP address on the interface. When the duplicate IP address is detected, a syslog message is generated. It helps the network operator to identify IP addresses misconfiguration. By default, this feature is enabled.

Examples

- This command disables the feature on the switch.

```
switch(config)# ip address duplicate detection disabled
```

- This command enables the feature.

```
switch(config)# ip address duplicate detection logging
```

Note: Commands are in global configuration mode, and are not per VRF.

This is an example of a Syslog message, when a duplicate IP address is detected.

```
Mar 24 16:41:57 cd290 Arp: %INTF-4-DUPLICATE_ADDRESS_WITH_HOST: IP address
 100.1.1.2
configured on interface Ethernet1/1 is in use by a host with
MAC address 00:00:01:01:00:00 on interface Ethernet1/1 in VRF default
```

13.1.1.3 Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) is a protocol that maps IP addresses to MAC addresses that local network devices recognize. The ARP cache is a table that stores the correlated addresses of the devices for which the router facilitates data transmissions.

After receiving a packet, routers use ARP to find the MAC address of the device assigned to the packet's destination IP address. If the ARP cache contains both addresses, the router sends the packet to the specified port. If the ARP cache does not contain the addresses, ARP broadcasts a request packet to all devices in the subnet. The device at the requested IP address responds and provides its MAC address. ARP updates the ARP cache with a dynamic entry and forwards the packet to the responding device. Static ARP entries can also be added to the cache through the CLI.

Proxy ARP is an ARP variant. A network device (proxy) responds to ARP requests for network addresses on a different network with its MAC address. Traffic to the destination is directed to the proxy device which then routes the traffic toward the ultimate destination.

Configuring ARP

The switch uses ARP cache entries to correlate 32-bit IP addresses to 48-bit hardware addresses. The arp aging timeout command specifies the duration of dynamic address entries in the Address Resolution Protocol (ARP) cache for addresses learned through the Layer 3 interface. The default duration is ***14400*** seconds (four hours).

Entries are refreshed and expired at a random time that is in the range of ***80%-100%*** of the cache expiry time. The refresh is tried three times at an interval of ***2%*** of the configured timeout.

Static ARP entries never time out and must be removed from the table manually.

Example

This command specifies an ARP cache duration of ***7200*** seconds (two hours) for dynamic addresses added to the ARP cache that were learned through ***VLAN 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# arp aging timeout 7200
switch(config-if-Vl200)# show active
interface Vlan200
   arp aging timeout 7200
switch(config-if-Vl200)#
```

The arp command adds a static entry to an Address Resolution Protocol (ARP) cache.

Example

This command adds a static entry to the ARP cache in the default VRF.

```
switch(config)# arp 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

13.1.1.3.1 Gratuitous ARP

Gratuitous ARP packets are broadcast by a device in response to an internal change rather than as a response to an ARP request. The gratuitous ARP packet is a request packet (no reply expected) that supplies an unrequested update of ARP information. In a gratuitous ARP packet, both the source and destination IP addresses are the IP of the sender, and the destination MAC address is the broadcast address (***ff:ff:ff:ff:ff:ff***).

Gratuitous ARP packets are generated to update ARP tables after an IPv4 address or a MAC address change occurs.

Configuring Gratuitous ARP

By default, Arista switch interfaces reject gratuitous ARP request packets. The arp gratuitous accept command configures an L3 interface to accept the gratuitous ARP request packets sent from a different device in the network and add their mappings to the ARP table. Gratuitous ARP can be configured on Ethernet interfaces, VLANs/SVI, or L3 port channels, but has no effect on L2 interfaces.

Example

These commands enable gratuitous ARP packet acceptance on ***interface ethernet 2/1***.

```
switch (config)# interface ethernet 2/1
switch (config-if-Et2/1)# arp gratuitous accept
```

13.1.1.4 Displaying ARP Entries

The show ip arp command displays ARP cache entries that map an IP address to a corresponding MAC address. The table displays addresses by their host names when the command includes the ***resolve*** argument.

Examples

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses.

```
switch> show ip arp

Address         Age (min)  Hardware Addr   Interface
172.25.0.2              0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
172.22.0.2              0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
172.22.0.5              0  f012.3118.c09d  Vlan1000, Port-Channel1
172.22.0.6              0  00e1.d11a.a1eb  Vlan1000, Ethernet5
172.22.0.7              0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
172.22.0.9              0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
 switch>
```

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses. Host names assigned to IP addresses are displayed in place of the address.

```
switch> show ip arp resolve

Address         Age (min)  Hardware Addr   Interface
green-vl101.new         0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
orange-vl1000.n         0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
purple.newcompa         0  f012.3118.c09d  Vlan1000, Port-Channel1
pink.newcompany         0  00e1.d11a.a1eb  Vlan1000, Ethernet5
yellow.newcompa         0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
royalblue.newco         0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
 switch>
```

13.1.1.4.1 ARP Inspection

Address Resolution Protocol (ARP) inspection command ip arp inspection vlan activates a security feature that protects the network from ARP spoofing. ARP requests and responses on untrusted interfaces are intercepted on specified VLANs, and intercepted packets are verified to have valid IP-MAC address bindings. All invalid ARP packets are dropped. On trusted interfaces, all incoming ARP packets are processed and forwarded without verification.

Enabling and Disabling ARP Inspection

By default, ARP inspection is disabled on all VLANs.

Examples

- This command enables ARP inspection on VLANs ***1*** through ***150***.

```
switch(config)# ip arp inspection vlan 1 - 150
switch(config)#
```

- This command disables ARP inspection on VLANs ***1*** through ***150***.

```
switch(config)# no ip arp inspection vlan 1 - 150
```

```
switch(config)#
```

- This command sets the ARP inspection default to VLANs ***1*** through ***150***.

```
switch(config)# default ip arp inspection vlan 1 - 150
switch(config)#
```

- These commands enable ARP inspection on multiple VLANs ***1*** through ***150*** and ***200*** through ***250***.

```
switch(config)# ip arp inspection vlan 1-150,200-250
switch(config)#
```

Syslog for Invalid ARP Packets Dropped

When an invalid ARP packet is dropped, the following syslog message appears. The log severity level can be set higher if required.

```
%SECURITY-4-ARP_PACKET_DROPPED: Dropped ARP packet on interface Ethernet28/1
 Vlan
2121 because invalid mac and ip binding. Received: 00:0a:00:bc:00:de/1.1.1.1.
```

Displaying ARP Inspection States

The command show ip arp inspection vlan displays the configuration and operation state of ARP inspection. For a VLAN range specified by **`show ip arp inspection vlan`** only VLANs with ARP inspection enabled will be displayed. If no VLAN is specified, all VLANs with ARP inspection enabled are displayed. The operation state turns to **Active** when hardware is ready to trap ARP packets for inspection.

Example

This command displays the configuration and operation state of ARP inspection for VLANs ***1*** through ***150***.

```
switch(config)# show ip arp inspection vlan 1 - 150

VLAN 1
----------
Configuration
: Enabled
Operation State : Active
VLAN 2
----------
Configuration
: Enabled
Operation State : Active
{...}
VLAN 150
----------
Configuration
: Enabled
Operation State : Active

switch(config)#
```

Displaying ARP Inspection Statistics

The command show ip arp inspection statistics displays the statistics of inspected ARP packets. For a VLAN specified by **`show ip arp inspection vlan`** only VLANs with ARP inspection enabled will be displayed. If no VLAN is specified, all VLANs with ARP inspection enabled are displayed.

The command clear arp inspection statistics clears ARP inspection.

Examples

- This command displays ARP inspection statistics for ***VLAN 1***.

```
switch(config)# show ip arp inspection statistics vlan 2

Vlan : 2
------------
ARP Req Forwarded = 20
ARP Res Forwarded = 20
ARP Req Dropped = 1
ARP Res Dropped = 1

Last invalid ARP:
Time: 10:20:30 ( 5 minutes ago )
Reason: Bad IP/Mac match
Received on: Ethernet 3/1
Packet:
  Source MAC: 00:01:00:01:00:01
  Dest MAC: 00:02:00:02:00:02
  ARP Type: Request
  ARP Sender MAC: 00:01:00:01:00:01
  ARP Sender IP: 1.1.1

switch(config)#
```

- This command displays ARP inspection statistics for ***ethernet interface 3/1***.

```
switch(config)# show ip arp inspection statistics ethernet interface
 3/1

Interface : 3/1
--------
ARP Req Forwarded = 10
ARP Res Forwarded = 10
ARP Req Dropped = 1
ARP Res Dropped = 1

Last invalid ARP:
Time: 10:20:30 ( 5 minutes ago )
Reason: Bad IP/Mac match
Received on: VLAN 10
Packet:
  Source MAC: 00:01:00:01:00:01
  Dest MAC: 00:02:00:02:00:02
  ARP Type: Request
  ARP Sender MAC: 00:01:00:01:00:01
  ARP Sender IP: 1.1.1

switch(config)#
```

- This command clears ARP inspection statistics.

```
switch(config)# clear arp inspection statistics
```

```
switch(config)#
```

Configure Trust Interface

By default, all interfaces are untrusted. The command ip arp inspection trust configures the trust state of an interface.

Examples

- This command configures the trust state of an interface.

```
switch(config)# ip arp inspection trust
switch(config)#
```

- This command configures the trust state of an interface to untrusted.

```
switch(config)# no ip arp inspection trust
switch(config)#
```

- This command configures the trust state of an interface to its default (untrusted).

```
switch(config)# default ip arp inspection trust
switch(config)#
```

Configure Rate Limit

When ARP inspection is enabled, ARP packets are trapped to the CPU. Two actions can be taken when the incoming ARP rate exceeds expectation. For notification purpose, the command ip arp inspection logging will enable logging of the incoming ARP packets. To prevent a denial-of-service attack, the command ip arp inspection limit will error-disable interfaces.

Examples

- This command enables logging of incoming ARP packets when its rate exceeds the configured value, and sets the rate to ***2048*** (which is the upper limit for the number of invalid ARP packets allowed per second), and sets the burst consecutive interval over which the interface is monitored for a high ARP rate to ***15*** seconds.

```
switch(config)# ip arp inspection logging rate 2048 burst interval
 15
switch(config)#
```

- This command configures the rate limit of incoming ARP packets to errdisable the interface when the incoming ARP rate exceeds the configured value, sets the rate to ***512*** (which is the upper limit for the number of invalid ARP packets allowed per second), and sets the burst consecutive interval over which the interface is monitored for a high ARP rate to ***11*** seconds.

```
switch(config)# ip arp inspection limit rate 512 burst interval 11
switch(config)#
```

- This command displays verification of the interface specific configuration.

```
switch(config)# interface ethernet 3/1
switch(config)# ip arp inspection limit rate 20 burst interval 5
switch(config)# interface Ethernet 3/3
switch(config)# ip arp inspection trust
```

```
switch(config)# show ip arp inspection interfaces

 Interface      Trust State  Rate (pps) Burst Interval
 -------------  -----------  ---------- --------------
 Et3/1          Untrusted    20         5
 Et3/3          Trusted      None       N/A

switch(config)#
```

Configure Errdisable Caused by ARP Inspection

If the incoming ARP packet rate on an interface exceeds the configured rate limit in burst interval, the interface will be errdisabled (by default). If errdisabled, the interface will stay in this state until you intervene with the command **`errdisable detect cause arp-inspection`** (e.g., after you perform a **`shutdown`** or **`no shutdown`** of the interface) or it automatically recovers after a certain time period. The command **`errdisable recovery cause arp-inspection`** will enable auto recovery. The command **`errdisable recovery interval`** will enable sharing the auto recovery interval among all errdisable interfaces. (See the chapter **Data Transfer Introduction** for information on all **`errdisable`** commands.

Examples:

- This command enables errdisable caused by an ARP inspection violation.

```
switch(config)# errdisable detect cause arp-inspection
switch(config)#
```

- This command disables errdisable caused by an ARP inspection violation.

```
switch(config)# no errdisable detect cause arp-inspection
switch(config)#
```

- This command enables auto recovery.

```
switch(config)# errdisable recovery cause arp-inspection
switch(config)#
```

- This command disables auto recovery.

```
switch(config)# no errdisable recovery cause arp-inspection
switch(config)#
```

- This command enables sharing the auto recovery interval of ***10*** seconds among all errdisable interfaces.

```
switch(config)# errdisable recovery interval 10
switch(config)#
```

- This command disables sharing the auto recovery interval of ***10*** seconds among all errdisable interfaces.

```
switch(config)# no errdisable recovery interval 10
switch(config)#
```

- This command displays the reason for a port entering the errdisable state.

```
switch(config)# show interfaces status errdisabled

Port          Name          Status       Reason
------------ ------------ ------------ ---------------
```

```
Et3/2                         errdisabled  arp-inspection

switch(config)#
```

Configure Static IP MAC Binding

The ARP inspection command ip source binding allows users to add static IP-MAC binding. If enabled, ARP inspection verifies incoming ARP packets based on the configured IP-MAC bindings. The static IP-MAC binding entry can only be configured on Layer 2 ports. By default, there is no binding entry on the system.

Examples

- This command configures static IP-MAC binding for IP address ***127.0.0.1,*** MAC address ***0001.0001.0001***, ***vlan 1***, and Ethernet interface ***slot 4*** and ***port 1***.

```
switch(config)# ip source binding 127.0.0.1 0001.0001.0001 vlan 1
 interface
ethernet 4/1
switch(config)#
```

- This command configures static IP-MAC binding for IP address ***127.0.0.1***, MAC address ***0001.0001.0001***, ***vlan 1***, and ***port-channel interface 20***.

```
switch(config)# ip source binding 127.0.0.1 0001.0001.0001 vlan 1
 interface
port-channel 20
switch(config)#
```

- This command displays the configured IP-MAC binding entries. Note that the Lease column is mainly used for displaying dynamic DHCP snooping binding entries. For static binding entries, lease time is shown as infinite.

```
switch(config)# show ip source binding 127.0.0.1 0001.0001.0001 static vlan 1
interface port-channel 20

MacAddress      IpAddress   Lease(sec)  Type   VLAN  Interface
--------------- ----------- ----------- ------ ----- --------------
0001.0001.0001  127.0.0.1   infinite    static 1     Port-Channel20

switch(config)#
```

13.1.2 IPv4 Routing

Internet Protocol version 4 (IPv4) is a communications protocol used for relaying network packets across a set of connected networks using the Internet Protocol suite. Routing transmits network layer data packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range. The connected subnets have IP address ranges that do not overlap.

A router is a network device that connects multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address. IPv4 and IPv6 are internet layer protocols that define packet-switched internetworking, including source-to-destination datagram transmission across multiple networks.

These sections describe IPv4 routing and route creation options:

- Enabling IPv4 Routing
- Static and Default IPv4 Routes

- Dynamic IPv4 Routes
- Viewing IPv4 Routes and Network Components

13.1.2.1 Enabling IPv4 Routing

When IPv4 routing is enabled, the switch attempts to deliver inbound packets to destination IPv4 addresses by forwarding them to interfaces or next hop addresses specified by the forwarding table.

The ip routing command enables IPv4 routing.

Example

This command enables IP routing:

```
switch(config)# ip routing
switch(config)#
```

13.1.2.2 Static and Default IPv4 Routes

Static routes are entered through the CLI and are typically used when dynamic protocols are unable to establish routes to a specified destination prefix. Static routes are also useful when dynamic routing protocols are not available or appropriate. Creating a static route associates a destination IP address with a local interface. The routing table refers to these routes as connected routes that are available for redistribution into routing domains defined by dynamic routing protocols.

The ip route command creates a static route. The destination is a network segment; the nexthop is either an IP address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

By default, the administrative distance assigned to static routes is ***1***. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of ***200*** is overridden by OSPF intra-area routes, which have a default distance of ***110***.

A route tag is a 32-bit number that is attached to a route. Route maps use tags to filter routes. Static routes have a default tag value of ***0***.

Example

This command creates a static route:

```
switch(config)# ip route 172.17.252.0/24 vlan 500
switch(config)#
```

Creating Default IPv4 Routes

The default route denotes the packet forwarding rule that takes effect when no other route is configured for a specified IPv4 address. All packets with destinations that are not established in the routing table are sent to the destination specified by the default route.

The IPv4 destination prefix is ***0.0.0.0/0*** and the next-hop is the default gateway.

Example

This command creates a default route and establishes ***192.14.0.4*** as the default gateway address:

```
switch(config)# ip route 0.0.0.0/0 192.14.0.4
```

```
switch(config)#
```

13.1.2.3 Dynamic IPv4 Routes

Dynamic routes are established by dynamic routing protocols. These protocols also maintain the routing table and modify routes to adjust for topology or traffic changes. Routing protocols assist the switch in communicating with other devices to exchange network information, maintaining routing tables, and establishing data paths.

The switch supports these dynamic IPv4 routing protocols:

- OSPFv2 Introduction
- Border Gateway Protocol (BGP)
- Routing Information Protocol (RIP)
- IS-IS

13.1.2.4 Viewing IPv4 Routes and Network Components

Displaying the FIB and Routing Table

The show ip route command displays routing table entries that are in the forwarding information base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal-cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **`show running-config`** command displays configured commands not in the FIB. The show ip route summary command displays the number of routes, categorized by source, in the routing table.

Examples

- This command displays IP routes learned through BGP.

```
switch> show ip route bgp

Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, A - Aggregate

 B E    170.44.48.0/23 [20/0] via 170.44.254.78
 B E    170.44.50.0/23 [20/0] via 170.44.254.78
 B E    170.44.52.0/23 [20/0] via 170.44.254.78
 B E    170.44.54.0/23 [20/0] via 170.44.254.78
 B E    170.44.254.112/30 [20/0] via 170.44.254.78
 B E    170.53.0.34/32 [1/0] via 170.44.254.78
 B I    170.53.0.35/32 [1/0] via 170.44.254.2
                             via 170.44.254.13
                             via 170.44.254.20
                             via 170.44.254.67
                             via 170.44.254.35
                             via 170.44.254.98 switch>
```

- This command displays a summary of routing table contents.

```
switch> show ip route summary

Route Source         Number Of Routes
```

```
-------------------------------------
connected                   15
static                       0
ospf                        74
  Intra-area: 32 Inter-area:33 External-1:0 External-2:9
  NSSA External-1:0 NSSA External-2:0
bgp                          7
  External: 6 Internal: 1
internal                    45
attached                    18
aggregate                    0 switch>
```

Displaying the IP Route Age

The show ip route age command displays the time when the route for the specified network was present in the routing table. It does not account for the changes in parameters like metric, next-hop etc.

Example:

This command displays the amount of time since the last update to ip route ***172.17.0.0/20***.

```
switch> show ip route 172.17.0.0/20 age

Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I - ISIS, A - Aggregate

 B E    172.17.0.0/20 via 172.25.0.1, age 3d01h

switch>
```

Displaying Gateways

A gateway is a router that provides access to another network. The gateway of last resort, also known as the default route, is the route that a packet uses when the route to its destination address is unknown. The IPv4 default route in is ***0.0.0.0/0***.

The show ip route gateway command displays IP addresses of all gateways (next hops) used by active routes.

Example

This command displays next hops used by active routes.

```
switch> show ip route gateway

The following gateways are in use:
   172.25.0.1 Vlan101
   172.17.253.2 Vlan2000
   172.17.254.2 Vlan2201
   172.17.254.11 Vlan2302
   172.17.254.13 Vlan2302
   172.17.254.17 Vlan2303
   172.17.254.20 Vlan2303
   172.17.254.66 Vlan2418
```

```
   172.17.254.67 Vlan2418
   172.17.254.68 Vlan2768
   172.17.254.29 Vlan3020 switch>
```

Displaying Host Routes

The show ip route host command displays all host routes in the host forwarding table. Host routes are those whose destination prefix is the entire address (mask = ***255.255.255.255*** or prefix = ***/32***). Each displayed host route is labeled with its purpose:

- **F** static routes from the FIB.
- **R** routes defined because the IP address is an interface address.
- **B** broadcast address.
- **A** routes to any neighboring host for which the switch has an ARP entry.

Example

This command displays all host routes in the host forwarding table.

```
switch# show ip route host

R - receive B - broadcast F - FIB, A - attached

F   127.0.0.1 to cpu
B   172.17.252.0 to cpu
A   172.17.253.2 on Vlan2000
R   172.17.253.3 to cpu
A   172.17.253.10 on Vlan2000
R   172.17.254.1 to cpu
A   172.17.254.2 on Vlan2901
B   172.17.254.3 to cpu
B   172.17.254.8 to cpu
A   172.17.254.11 on Vlan2902
R   172.17.254.12 to cpu

F   172.26.0.28 via 172.17.254.20 on Vlan3003
                via 172.17.254.67 on Vlan3008
                via 172.17.254.98 on Vlan3492
via 172.17.254.86 on Vlan3884
                via 172.17.253.2 on Vlan3000
F   172.26.0.29 via 172.25.0.1 on Vlan101
F   172.26.0.30 via 172.17.254.29 on Vlan3910
F   172.26.0.31 via 172.17.254.33 on Vlan3911
F   172.26.0.32 via 172.17.254.105 on Vlan3912 switch#
```

13.1.3 IPv4 Multicast Counters

IPv4 multicast counters allow association of IPv4 multicast routes with a packet or byte counter.

This chapter contains the following sections.

- Multicast Counters Hardware Overview
- Multicast Counters iBGP and eBGP Configuration
- Configuring IPv4 Multicast Counters

13.1.3.1 Multicast Counters Hardware Overview

This section describes a hardware overview for multicast counters, and contains the following sections.

- Platform Independent Requirements for Counters
- Policer Counter Overview
- BGP Functions Supported for Arista Switches
- Additional Requirements

13.1.3.1.1 Platform Independent Requirements for Counters

The following platform independent requirements include:

- Enable/Disable counters
- Clear counters
- Show counters
- Configure counter mode for byte (default) or frame mode

13.1.3.1.2 Policer Counter Overview

The switch hardware has two policer banks, each with 4k entries and each entry has one 32 bit entry1, and one 32 bit entry2, which can be used as either packet counter or byte counter.

In the pipeline, each bank can have one policer index coming from upstream blocks, which means different features cannot update multiple policer entries in the same bank simultaneously. Therefore, different features cannot share entries in the same bank.

In switch hardware routing, each FFU/BST entry points to a corresponding RAM. A policer index is saved in the action ram, so when installing a multicast route into hardware, platform code will get a policer index and saved in the action field. If a policer index is unavailable, a counter is not added to the action field.

Switch hardware can have multiple features competing for the policer banks. It is desirable to have a platform command to reserve policer banks dedicated for a certain feature.

The following command reserves one or two policer banks to be used only by the named feature:

```
[no] platform fm6000 [nat|acl|qos|multicast] policer banks <1|2>
```

Available bank(s) are reserved for the feature. Otherwise the command takes effect at the next reboot or FocalPointV2 agent restart. This reservation guarantees the configured number of bank(s) for this feature. However, the feature can still possibly obtain the other policer bank if it needs more, and the other bank is available.

If a feature has a pending reservation request which is not fulfilled because of availability, and some other feature frees a bank, the bank will be allocated to the pending feature.

13.1.3.1.3 BGP Functions Supported for Arista Switches

Arista switches support these BGP functions:

- A single BGP instance
- Simultaneous internal (IBGP) and external (EBGP) peering
- Multiprotocol BGP
- BGP Confederations

13.1.3.1.4 Additional Requirements

On switch hardware, the following additional requirements include:

- Reservation of policer banks
- Notification of policer bank availability when a policer entry is freed by other features

13.1.3.2 Multicast Counters iBGP and eBGP Configuration

This section describes the commands required to configure an iBGP and an eBGP topology, and contains the following sections.

- **Policer Usage**

13.1.3.2.1 Policer Usage

There are two types of counters – those created by wildcard creation and by specific creation. When a specific counter is required and the hardware runs out of policer entries, a wildcard counter is forced to give up its policer entry.

If the user configures a specific counter and the Starter Group (SG) already has a wildcard-created counter for it, then this counter is upgraded to a specific one, with no change in hardware policer index. If the user configures both a wildcard counter and specific counter for this SG, and subsequently deletes the specific counter, the counter for this SG is downgraded to a wildcard, with no change in hardware policer index. However, if another specific counter is pending for a hardware policer index, then this policer entry will be assigned to that counter due to its higher precedence.

Even if a counter is configured by the user, in order to conserve the use of hardware resources, do not allocate a policer entry until a real route (G, S) is programmed into the Frame Filtering and Forwarding Unit (FFU).

13.1.3.3 Configuring IPv4 Multicast Counters

Perform the following CLI steps to configure IPv4 multicast counters on the FM6000 platform:

1. Execute the global configuration command:

   - **no|default `ip multicast count` bytes| packets**

   Enables wildcard counters. Also used to change bytes / packets mode. When hardware runs of resources, specific creation has priority to preempt counters from wildcard creation. The **bytes | packets** optional keyword enables the counter to be in either bytes mode or packets mode. This mode applies to all counters. When the counter mode changes, all counter values will be reset to zero.

   - **no|default `ip multicast count` <*G*> <*S*>**

   This is only takes affect when **`ip multicast count`** is enabled. Either **<*G*> <*S*>** or **bytes|packets** optional keyword is used. They can not be used concurrently.

   No | default Commands: (default is same as no)

   - **`no ip multicast count`** Deletes all multicast counters, including explicit **<*G*> <*S*>** routes
     - **`no ip multicast count` <*G*> <*S*>** Removes the config. Does not delete the counter because the wildcard is still active.
     - If no **<*G, S*>** is specified, all multicast routes will have counters unless the hardware runs out of resources. The creation of counters is referred to as “wildcard creation.”
     - If **<*G, S*>** is specified, only **<*G, S*>** will get a counter (and no other route). The creation of counters is referred to as “specific creation.” By default, all mcast routes will have counters allocated. This **<*G, S*>** configuration is applicable when the hardware runs out of resources. Specific **<*G, S*>** creation has priority to preempt counters from wildcard creation.

   The ***byte | frame*** optional keyword enables the counter to be in either byte mode or frame mode. This mode applies to all counters. When the counter mode changes, all counter values will be reset to zero.

   Either **<*G, S*>**, or ***byte | frame*** optional keywords are used but cannot be used together. All counters are ***byte|frame***. The ***byte|frame*** mode is global, and not applicable on a **<*G, S*>** basis.

2. Execute clear command:

```
clear ip multicast count <G> <S>
```

3. Execute show command:

```
show multicast fib ipv4 <G> count
```

This command currently exists but does not show anything.

This show command is intended to display the following (example):

```
switch> show multicast fib ipv4 count
Activity poll time: 60 seconds
225.1.1.1 100.0.0.2
Byte: 123
Vlan100 (iif)
Vlan200
Activity 0:00:47 ago
```

Total counts is the sum of counts from all sources in that group.

The count value can be **N/A** if a mroute does not have an associated counter.

If the count value for any source in a ***G*** is **N/A**, then the total counts for ***G*** will be shown as **N/A**. However, the count values for other sources are still shown.

13.1.4 Route Management

When routing is enabled, the switch discovers the best route to a packet's destination address by exchanging routing information with other devices. IP routing is disabled by default.

The following sections describes routing features that the switch supports:

- Route Redistribution
- Equal Cost Multipath Routing (ECMP) and Load Sharing
- Unicast Reverse Path Forwarding (uRPF)
- Routing Tables / Virtual Routing and Forwarding (VRF)
- RIB Route Control

13.1.4.1 Route Redistribution

Route redistribution is the advertisement, into a dynamic routing protocol's routing domain, of connected (static) routes or routes established by other routing protocols. By default, the switch advertises only routes in a routing domain that are established by the protocol that defined the domain.

Route redistribution commands specify the scope of the redistribution action. By default, all routes from a specified protocol (or all static routes) are advertised into the routing domain. Commands can also filter routes by applying a route map, which defines the subset of routes to be advertised.

13.1.4.2 Equal Cost Multipath Routing (ECMP) and Load Sharing

Equal Cost Multi-Path (ECMP) is a routing strategy where traffic is forwarded over multiple paths that have equal routing metric values.

Configuring ECMP (IPv4)

All ECMP paths are assigned the same tag value; commands that change the tag value of a path also change the tag value of all paths in the ECMP route.

In a network topology using ECMP routing, hash polarization may result when all switches perform identical hash calculations. Hash polarization leads to uneven load distribution among the data paths. Hash polarization is avoided when switches use different hash seeds to perform hash calculations.

The ip load-sharing command provides the hash seed to an algorithm that the switch uses to distribute data streams among multiple equal-cost routes to a specified subnet.

Example

This command sets the IPv4 load sharing hash seed to ***20***:

```
switch(config)# ip load-sharing fm6000 20
switch(config)#
```

Multicast Traffic Over ECMP

The switch attempts to spread outbound unicast and multicast traffic to all ECMP route paths equally. To disable the sending of multicast traffic over ECMP, use the multipath none command or the no version of the multipath deterministic command.

Resilient ECMP

Resilient ECMP is used for those prefixes where it is not desirable for routes to be rehashed due to link flap, typically where ECMP is being used for load balancing. Resilient ECMP configures a fixed number of next-hop entries in the hardware ECMP table for all the routes within a specified IP address prefix. Implementing fixed table entries for a specified next-hop address allows data flows that are hashed to a valid next-hop number to remain intact even when some of the next hops go down or come back online.

Resilient ECMP is enabled for all routes within a specified prefix using the ip hardware fib ecmp resilience command. The command specifies the maximum number of next-hop addresses that the hardware ECMP table can contain for the specified IP prefix, and configures a redundancy factor that facilitates the duplication of next-hop addresses in the table. The fixed table space for the address is the maximum number of next hops multiplied by the redundancy factor. When the table contains the maximum number of next-hop addresses, the redundancy factor specifies the number of times each address is listed in the table. When the table contains fewer than the maximum number of next-hop addresses, the table space entries are filled by additional duplication of the nexthop addresses.

Resilient ECMP is also available for IPv6 IP addresses.

Example

This command configures a hardware ECMP table space of 24 entries for the IP address ***10.14.2.2/24***. A maximum of six next-hop addresses can be specified for the IP address. When the table contains six next-hop addresses, each appears in the table four times. When the table contains fewer than six next-hop addresses, each is duplicated until the 24 table entries are filled.

```
switch(config)# ip hardware fib ecmp resilience 10.14.2.2/24 capacity
 6 redundancy 4
switch(config)#
```

13.1.4.2.1 Resilient Equal-Cost Multi-Path (RECMP) Deduping

Routes covered by a Resilient Equal-Cost Multi-Path (RECMP) prefix are types of routes that make use of hardware tables dedicated for Equal-Cost Multi-Path (ECMP) routing. Resilient ECMP (RECMP) deduping reduces the number of ECMP hardware table entries allocated by the switch by forcing the routes that share

the same set of next hops but point to different hardware table entries to point to the same hardware table entry when hardware resource utilization is high. Forcing RECMP routes to change the hardware table entry that they point to may potentially cause a traffic flow disruption for any existing flows going over that route. The deduping process will attempt to minimize the amount of potential traffic loss caused.

Each route needs to allocate hardware table entries in the ASIC, which contains forwarding information for the route, such as what its next-hops are, what egress links each next-hop uses, etc. The network device uses these hardware table entries when making forwarding decisions for a packet that is meant for a certain route. These ECMP hardware tables are limited in size and can fill up quickly if there are a large number of these hardware table entries allocated. One option to ease the usage of these hardware tables is to force RECMP routes to share hardware table entries.

There is already an existing feature for RECMP routes to get them to point to the same hardware table entry if they share the same set of next hops and the ordering of the next-hops is the same. However, RECMP routes may end up sharing the same set of next-hops, but the next-hop ordering may be different between them, and therefore the routes end up occupying different hardware table entries in the ASIC. RECMP routing has a property wherein the current ordering of next-hops for a given route is influenced by its previous orderings. The ordering between the routes can differ because these routes may have had a different set of next hops at some previous time before they finally converged onto the same set of next-hops.

When the ECMP hardware resource usage crosses the high threshold, the deduping process begins, and it lasts until the ECMP hardware resource usage falls below the low threshold. Use the **`IP hardware fib next-hop resource optimization thresholds`** command to modify the thresholds.

13.1.4.2.1.1Configuring Resilient ECMP Deduping

The Resilient ECMP Deduping is enabled by default.

- The following command is used to disable all the hardware resource optimization features:

```
switch(config)# ip hardware fib next-hop resource optimization disabled
```

- The following command is used to re-enable the all hardware resource optimization features after disabling them:

```
switch(config)# no ip hardware fib next-hop resource optimization disabled
```

- The following command is used to configure the thresholds for starting and stopping the optimization:

```
switch(config)# ip hardware fib next-hop resource optimization thresholds
 low <20> high <80>
```

Note:

- The value specified for the threshold represents the percentage of resource utilization, and is an integer between ***0*** and ***100***.
- Setting the high threshold to ***80*** indicates that optimization starts when the resource utilization is above ***80%***. The default value of this threshold is ***90***.
- Setting the low threshold to ***20*** indicates that optimization stops when the resource utilization is below ***20%***. The default value of this threshold is ***85***.

13.1.4.2.1.2Show Commands

- The **`show ip hardware fib summary`** command is used to display the statistics of this RECMP deduping:

  Example

```
switch# show ip hardware fib summary
Fib summary
```

```
-----------
Adjacency sharing: disabled
BFD peer event: enabled
Deletion Delay: 0
Protect default route: disabled
PBR: supported
URPF: supported
ICMP unreachable: enabled
Max Ale ECMP: 600
UCMP weight deviation: 0.0
Maximum number of routes: 0
Fib compression: disabled
Resource optimization for adjacency programming: enabled
Adjacency resource optimization thresholds: low 20, high 80
```

The last two lines of the output shows whether RECMP deduping is enabled, and what are the corresponding threshold values for starting and stopping the optimization process.

- The **`show hardware capacity`** command is used to display the utilization of the hardware resources. The example below shows the multi-level hierarchy ECMP resources:

```
switch# show hardware capacity
Forwarding Resources Usage

Table    Feature         Chip    Used     Used    Free Committed    Best Case    High
                                 Entries  (%)     Entries    Entries      Max          Watermark
                                                                                     Entries
------- --------------- ------- -------- ------- --------   ------------ ------------ ---------
ECMP                             0        0%     4095        0        4095         0
ECMP     Mpls                    0        0%     4095        0        4095         0
ECMP     Routing                 0        0%     4095        0        4095         0
ECMP     VxlanOverlay            0        0%     4095        0        4095         0
ECMP     VxlanTunnel             0        0%     3891        0        3891         0
```

13.1.4.2.1.3Limitations

- With RECMP deduping, optimization of a sub-optimal ECMP route requires releasing and reallocating hardware resources for the route. Therefore the process may increase overall convergence time for route programming. It may not be desirable to always start the optimization when the hardware resource is sufficient. The threshold value for starting the optimization should be adjusted based on the route scale of the network.
- The deduping of ECMP hardware resources may cause potential traffic flow disruption for traffic flows going over RECMP routes with changing hardware table entries. While the deduping process tries to minimize the amount of traffic flow disruption, it is still sometimes inevitable.
- RECMP hardware table entries can only be deduped to other RECMP hardware table entries that share the same set of nexthops. This puts a limit to the amount of RECMP hardware table entries that can be reduced to the number of RECMP hardware table entries with unique nexthop sets.

13.1.4.3 Unicast Reverse Path Forwarding (uRPF)

Unicast Reverse Path Forwarding (uRPF) verifies the accessibility of source IP addresses in packets that the switch forwards. The switch drops a packet when uRPF determines that the routing table does not contain an entry with a valid path to that packet's source IP address.

IPv4 and IPv6 uRPF operate independently. uRPF is VRF aware. Commands that do not specify a VRF utilize the default instance. Multicast routing is not affected by uRPF.

uRPF defines two operational modes: strict mode and loose mode.

- **Strict mode:** uRPF also verifies that a packet is received on the interface that its routing table entry will use for its return packet.
- **Loose mode:** uRPF validation does not consider the inbound packet's ingress interface.

13.1.4.3.1 uRPF Operation

uRPF is configurable on interfaces. For packets arriving on a uRPF-enabled interfaces, the source IP address is verified by examining the source and destination addresses of unicast routing table entries.

uRPF requires a reconfigured routing table to support IP address verification. When uRPF is enabled for the first time, unicast routing is briefly disabled to facilitate the routing table reconfiguration. Multicast routing is not affected by the initial uRPF enabling.

A packet fails uRPF verification if the table does not contain an entry whose source or destination address matches the packet's source IP address. In strict mode, the uRPF also fails when the matching entry's outbound interface does not match the packet's ingress interface.

uRPF verification is not available for the following packets:

- DHCP (Source is ***0.0.0.0*** – Destination is ***255.255.255.255***).
- IPv6 link local (***FE80::/10***).
- Multicast packets.

ECMP uRPF

When verifying ECMP routes, strict mode checks all possible paths to determine that a packet is received on the correct interface. Strict mode is supported for ECMP groups with a maximum of eight routing table entries. The switch reverts to loose mode for ECMP groups that exceed eight entries.

Default Routes

uRPF strict mode provides an **`allow-default`** option that accepts default routes. On interfaces that enable allow-default and a default route is defined, uRPF strict mode validates a packet even when the routing table does not contain an entry that matches the packet's source IP address. When allow-default is not enabled, uRPF does not consider the default route when verifying an inbound packet.

Null Routes

`NULL0` routes drop traffic destined to a specified prefix. When uRPF is enabled, traffic originating from null route prefixes is dropped in strict and loose modes.

13.1.4.3.2 uRPF Configuration

Unicast Reverse Path Forwarding (uRPF) is enabled for IPv4 packets ingressing the configuration mode interface through the **ip verify** command.

Note: uRPF cannot be enabled on interfaces with ECMP member FECs.

Examples

- This command enables uRPF loose mode on ***interface vlan 17***.

```
switch(config)# interface vlan 17
switch(config-if-Vl17)# ip verify unicast source reachable-via any
switch(config-if-Vl17)# show active
 interface Vlan17
   ip verify unicast source reachable-via any
switch(config-if-Vl17)#
```

- This command enables uRPF strict mode on ***interface vlan 18***.

```
switch(config)# interface vlan 18
switch(config-if-Vl18)# ip verify unicast source reachable-via rx
switch(config-if-Vl18)# show active
```

```
 interface Vlan18
   ip verify unicast source reachable-via rx
switch(config-if-Vl18)#
```

13.1.4.4 Routing Tables / Virtual Routing and Forwarding (VRF)

An IP routing table is a data table that lists the routes to network destinations and metrics (distances) associated with those routes. A routing table is also known as a Routing Information Base (RIB).

Virtual Routing and Forwarding (VRF) allows traffic separation by maintaining multiple routing tables. Arista switches support multiple VRF instances: one global or default VRF called “default” and multiple user-defined VRFs; the number of user-defined VRFs supported varies by platform. VRFs can be used as management or data plane VRFs.

- Management VRFs have routing disabled. They are typically used for management-related traffic.
- Dataplane VRFs have routing enabled. They support routing protocols and packet forwarding (hardware and software).

Dataplane VRFs are supported by Trident, FM6000, and Arad platform switches.

VRFs support unicast IPv4 and IPv6 traffic and multicast traffic. Loopback, SVI, and routed ports may be added to VRFs. Management ports may be added without any hardware forwarding.

To allow overlap in the sets of IP addresses used by different VRF instances, a Route Distinguisher (RD) may be prepended to each address. RDs are defined in ***RFC 4364***.

13.1.4.4.1 Default VRF

The default VRF on Arista switches is called “default.” It is created automatically and cannot be renamed or configured. Some configuration options accept “default” as a VRF input.

13.1.4.4.2 User-Defined VRFs

A user-defined VRF is created with the vrf instance command. After its creation, a VRF may be assigned a Route Distinguisher (RD) with the rd (VRF configuration mode) command in the VRF submode of Router-BGP Configuration Mode.

Examples

- These commands create a VRF named ***purple***, place the switch in BGP VRF configuration mode for that VRF, and specify a route distinguisher for the VRF identifying the administrator as ***AS 530*** and assigning ***12*** as its local number.

```
switch(config)# vrf instance purple
switch(config-vrf-purple)# router bgp 50
switch(config-router-bgp)# vrf purple
switch(config-router-bgp-vrf-purple)# rd 530:12
switch(config-router-bgp-vrf-purple)#
```

- To add interfaces to a user-defined VRF, enter configuration mode for the interface and use the vrf (Interface mode) command. Loopback, SVI, and routed ports can be added to a VRF.

  These commands add ***vlan 20*** to the VRF named ***purple***.

```
switch(config)# interface vlan 20
switch(config-if-Vl20)# vrf purple
switch(config-if-Vl20)#
```

- The show vrf command shows information about user-defined VRFs on the switch.

This command displays information for the VRF named ***purple***.

```
switch> show vrf purple
Vrf     RD          Protocols  State       Interfaces
------- ----------- ---------- ----------- ------------
purple  64496:237   ipv4       no routing  Vlan42, Vlan43 switch>
```

13.1.4.4.2.1rd (VRF configuration mode)

The **rd** command issued in VRF Configuration Mode is a legacy command supported for backward compatibility. To configure a Route Distinguisher (RD) for a VRF, use the **rd (VRF configuration mode)** command.

Note: Legacy RDs that were assigned to a VRF in VRF Configuration Mode will still appear in **show vrf** outputs if an RD has not been configured in Router-BGP VRF Configuration Mode, but they no longer have an effect on the system.

13.1.4.4.3 Context-Active VRF

The context-active VRF specifies the default VRF that VRF-context aware commands use when displaying or refreshing routing table data.

VRF-context aware commands include:

- clear arp-cache
- show ip
- show ip arp
- show ip route
- show ip route gateway
- show ip route host

The cli vrf command specifies the context-active VRF.

Example

This command specifies ***magenta*** as the context-active VRF.

```
switch# cli vrf magenta
switch# show routing-context vrf
Current VRF routing-context is magenta
```

The show routing-context vrf command displays the context-active VRF.

Example

This command displays the context-active VRF.

```
switch> show routing-context vrf
Current VRF routing-context is magenta switch>
```

13.1.4.5 RIB Route Control

The Routing Information Base (RIB) is composed of the routing information learned by the routing protocols, including static routes. The Forwarding Information Base (FIB) is composed of the routes actually used to forward traffic through a router.

Forwarding Information Base (FIB) makes IP destination prefix-based switching decisions. The FIB is similar to a routing table or information base. It maintains the forwarding information for the winning routes from the RIB. When routing or topology changes occur in the network, the IP routing table information is updated, and those changes are reflected in the FIB.

13.1.4.5.1 Configuring FIB policy

The RIB calculates the best/winning routes to each destination and place these routes in the forwarding table. Based on the FIB policy configured the best routes are advertised.

For example, a FIB policy can be configured to deny the routes for FIB programming, however, it does not prevent these routes from being advertised by a routing protocol, or to be redistributed into another routing domain, or to be used for recursive resolution in the IP RIB. FIB policies control the size and content of the routing tables, and the best route to take to reach a destination.

The **`rib ipv4 | ipv6 fib policy`** command is used to enable FIB policy for a particular VRF under router general configuration mode.

The following match statements are supported:

- **`match interface`**
- **`match`** [ip | ipv6] address ***prefix-list***
- **`match`** [ip | ipv6] resolved-next-hop ***prefix-list***
- **`match isis level`**
- **`match metric`**
- match source-protocol

Example

The following example enables FIB policy for IPv4 in the default VRF, using the route map, ***map1***.

```
switch(config)# router general
switch(config-router-general)# vrf default
switch(config-router-general-vrf-default)# rib ipv4 fib policy map1
```

13.1.4.5.2 Displaying FIB Information

Use the **`show rib route <ipv4|ipv6> fib policy exclude`** command to display the RIB information. The **fib policy excluded** option displays the RIB routes that have been excluded from being programmed into FIB, by FIB policy.

Example

The following example displays the routes filtered by FIB policy using the **fib policy excluded** option of the **`show rib route ip|ipv6`** command.

```
switch# show rib route ipv6 fib policy excluded
switch# show rib route ip bgp fib policy excluded

VRF name: default, VRF ID: 0xfe, Protocol: bgp
Codes: C - Connected, S - Static, P - Route Input
```

```
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis
       > - Best Route, * - Unresolved Nexthop
       L - Part of a recursive route resolution loop
>B    10.1.0.0/24 [200/0]
         via 10.2.2.1 [115/20] type tunnel
            via 10.3.5.1, Ethernet1
         via 10.2.0.1 [115/20] type tunnel
            via 10.3.4.1, Ethernet2
            via 10.3.6.1, Ethernet3
>B    10.1.0.0/24 [200/0]
         via 10.2.2.1 [115/20] type tunnel
            via 10.3.5.1, Ethernet1
         via 10.2.0.1 [115/20] type tunnel
            via 10.3.4.1, Ethernet2
            via 10.3.6.1, Ethernet3
```

13.1.4.5.3 Displaying RIB Route Information

Use the show rib route ip command to view the IPv4 RIB information.

Example:

This command displays IPv4 RIB static routes.

```
switch# show rib route ip static

VRF name: default, VRF ID: 0xfe, Protocol: static
Codes: C - Connected, S - Static, P - Route Input
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis
       > - Best Route, * - Unresolved Nexthop
       L - Part of a recursive route resolution loop
>S    10.80.0.0/12 [1/0]
         via 172.30.149.129 [0/1]
            via Management1, directly connected
>S    172.16.0.0/12 [1/0]
         via 172.30.149.129 [0/1]
            via Management1, directly connected switch#
```

13.1.5 IPv4 Route Scale

IPv4 routes are optimized to achieve route scale when route distribution has a large number of routes of one or two parameters, with each parameter consisting of prefix lengths ***12***, ***16***, ***20***, ***24***, ***28***, and ***32***. If two separate prefix lengths are configured (in any order), one of them must be the prefix length of ***32***.

Note: IPv4 Route Scale cannot be used with AlgoMatch.

The following sections describes IPv4 route scale configuration, show commands, and system log messages:

- Configuring IPv4 Route Scale
- IPv4 Routescale with 2-to-1 Compression
- Show Commands

13.1.5.1 Configuring IPv4 Route Scale

IPv4 route scale is enabled by the ip hardware fib optimize command for the configuration mode interface. The platform Layer 3 agent is restarted to ensure IPv4 routes are optimized with the agent SandL3Unicast terminate command for the configuration mode interface.

Example

This configuration command allows configuring prefix lengths ***12*** and ***32***.

```
switch(config)# ip hardware fib optimize exact-match prefix-length 12
 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

One of the two prefixes in this command is a prefix-length of ***32***, which is required in the instance where there are two prefixes. For this command to take effect, you must restart the platform Layer 3 agent.

Example

This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)# agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

Example:

This configuration command allows configuring prefix lengths ***32*** and ***16***.

```
switch(config)# ip hardware fib optimize exact-match prefix-length 32
 16
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

One of the two prefixes in this command is a prefix-length of ***32***, which is required in the instance where there are two prefixes. For this command to take effect, you must restart the platform Layer 3 agent.

Examples

- This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

  ```
  switch(config)#agent SandL3Unicast terminate
  SandL3Unicast was terminated
  ```

  Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

- This configuration command allows configuring prefix length ***24***.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 24
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

In this instance, there is only one prefix-length, so a prefix-length of ***32*** is not required. For this command to take effect, you must restart the platform Layer 3 agent.

Examples

- This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

  Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.
- This configuration command allows configuring prefix length ***32***.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

  For this command to take effect, you must restart the platform Layer 3 agent.
- This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)# agent SandL3Unicast terminate
SandL3Unicast was terminated
```

  Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.
- This configuration command disables configuring prefix lengths ***12*** and ***32***.

```
switch(config)#no ip hardware fib optimize exact-match prefix-length
 12 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 not optimized
```

One of the two prefixes in this command is a prefix-length of ***32***, which is required in the instance where there are two prefixes. For this command to take effect, you must restart the platform Layer 3 agent.

Examples

- This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are not optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

  Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

- This configuration command attempts to configure prefix length ***20*** and ***28*** which triggers an error exception. One of the two prefixes in this command must be a prefix-length of ***32***, which is required in the instance where there are two prefixes.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 20
 28
% One of the prefix lengths must be 32
```

IPv4 routes of certain prefix lengths can be optimized for enhanced route scale. The following command disable prefix optimization on the specified VRF(s) to provide more flexibility.

Examples

- This configuration command disables prefix optimization on the default VRF.

```
switch(config)# ip hardware fib optimize disable-vrf default
! Please restart layer 3 forwarding agent to ensure that the
 disable-vrf option change takes effect
```

- This configuration command disables prefix optimization on VRFs named *vrf1* and*vrf2*.

```
switch(config)# ip hardware fib optimize disable-vrf vrf1 vrf2
! Please restart layer 3 forwarding agent to ensure that the
 disable-vrf option change takes effect
```

- This configuration command restarts the platform Layer 3 agent to ensure disable-vrf configuration to take effect.

```
switch(config)# agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Starting from the *EOS Release 4.26.0F*, /32 prefix length optimization command is supported in the R3 series.

Examples

- This configuration command enables prefix optimization on the default VRF.

```
switch(config)# ip hardware fib optimize vrf default prefix-length
 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

- This configuration command enables prefix optimization on VRFs named ***vrf1*** and ***vrf2***.

```
switch(config)# ip hardware fib optimize vrf vrf1 vrf2 prefix-length
 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

- This configuration command disables optimization on ***vrf1*** and ***vrf2*** optimization configured in above example.

```
switch(config)# no ip hardware fib optimize vrf vrf1
```

```
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

The **platform trident forwarding-table partition flexible** command enables ALPM Mode in Flexible UFT mode using a subset of resources, so ALPM and Exact Match can coexist. Prior to this release, ALPM could only be programmed in mode 4 where all the UFT resources were used and in flexible partition mode, configuring ALPM was not supported. This limits the number of IP routes that can be supported.

Examples

- This configuration command sets up the flexible partition.

```
switch(config)# platform trident forwarding-table partition
 flexible ?
  alpm          Shared UFT bank entries for the ALPM table
  exact-match   Shared UFT bank entries for the exact-match table
  l2-shared     Shared UFT bank entries for the MAC table
  l3-shared     Shared UFT bank entries for the host table
```

- ALPM gives the route prefix in DEFIM (TCAM table for longest prefix matched (LPM) lookup) and ALPM tables.

```
switch(config)# platform trident forwarding-table partition flexible
 alpm ?
  184320  Upto 180K LPM routes
  368640  Upto 360K LPM routes
```

Note: The size parameter has following values:

- DCS-7300X3: 180k and 360k are accepted.
- CCS-720XP: 144k and 96k are accepted.
- Other sizes are invalid.

13.1.5.2 IPv4 Routescale with 2-to-1 Compression

The IPv4 routescale with 2-to-1 compression optimizes certain prefix lengths and enhances the route scale capabilities on 7500R, 7280R, 7500R2, and 7280R2 platforms. The compression is best suited to achieve route scale when route distribution has a large number of routes of one or two prefix lengths.

13.1.5.2.1 Configuring IPv4 Routescale 2-to-1 Compression

Use the **compress** command to increase the hardware resources available for the specified prefix length. This command allows configuring up to one compressed prefix length, and this command is supported only on 7500R, 7280R, 7500R2, and 7280R2 platforms.

Note: The **compress** command takes effect only when you restart the platform Layer3 agent on 7500R, 7280R, 7500R2, and 7280R2 platforms. Use command **agent SandL3Unicast terminate** to restart the platform Layer3 agent.

Examples

- In the following example we are configuring prefix length ***20*** and ***24***, expanding prefix length ***19*** and ***23***, and compressing prefix length ***25***.

```
switch(config)# ip hardware fib optimize prefix-length 20 24 expand 19 23
 compress 25
```

```
 ! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

- In the following example we are configuring prefix length ***20*** and ***23***, expanding prefix length ***19***, compressing prefix length ***24***.

```
switch(config)# ip hardware fib optimize prefix-length 20 23 expand 19
 compress 24
 ! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

- Optionally, you can also use the **internet** profile to configure the IPv4 route scale compression.

```
switch(config)# ip hardware fib optimize prefixes profile internet
 ! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

Configure a new TCAM profile for the **`compress`** configuration to work, and disable a few features in the new TCAM profile to make space for the flex-route feature in the hardware. Features like **acl vlan ip** and the **mirror ip** have to be disabled, if you need any of these features or any other features to be enabled with flex-route feature, contact the Arista team.

The **internet** profile works differently based on whether the flex-route feature is enabled in the TCAM profile or not. If the flex-route feature is enabled, the **internet** profile behaves like **`ip hardware fib optimize prefix-length 20 23 expand 19 22 compress 24`**. If the flex-route feature is disabled, the **internet** profile behaves as **`ip hardware fib optimize prefix-length 20 24 expand 19 23`**.

Example

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile flex-route copy default
switch(config-hw-tcam-profile-flex-route)# feature flex-route copy system-
feature-source-profile
switch(config-hw-tcam-profile-flex-route-feature-flex-route)# exit
switch(config-hw-tcam-profile-flex-route)# no feature acl vlan ip
switch(config-hw-tcam-profile-flex-route)# no feature mirror ip
switch(config-hw-tcam-profile-flex-route)# exit
Saving new profile 'flex-route'
switch(config-hw-tcam)# system profile flex-route
```

13.1.5.2.2 Limitations

- A maximum of two prefix lengths can be optimized directly at any point of time, of which only one can be a non-nibble aligned prefix length. Additional prefix lengths can be optimized using the **expand** or the **compress** options.
- A maximum of 1-to-4 way expansion and 2-to-1 way compression into any optimized prefix length is supported. Multiple expansion prefix lengths can be programmed at any time, however, there can be just one compression prefix length programmed at any given point in time.
- A maximum of ***4096*** next-hops can be reliably pointed to by the compressed prefixes using 2-to-1 way compression.
- The 2-to-1 compression cannot be enabled along with unicast RPF. When both features are enabled together, unicast RPF functionality may not be correct.
- The flex-route feature in TCAM profiles based only on the default profile, while disabling the **acl vlan ip** and the **mirror ip** features. Contact the Arista team if any other feature, that is not available in the default TCAM profile, is required to be supported along with the flex-route feature, including support for Mirror to GRE tunnel or ACLs on SVI.
- VXLAN is not supported with the compress option of this feature. There is no Syslog or a warning message when VXLAN is configured along with the 2-to-1 way compression feature.

13.1.5.3 Show Commands

The IPv4 route scale summary is displayed by the show platform arad ip route summary command for the configuration mode interface. Resources for all IPv4 route scale routes are displayed by the show platform arad ip route command for the configuration mode interface.

Examples

- This command shows hardware resource usage of IPv4 routes.

```
switch(config)# show platform arad ip route summary

Total number of VRFs: 1
Total number of routes: 25
Total number of route-paths: 21
Total number of lem-routes: 4
```

- This command shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources appear with an asterisk (*).

```
switch(config)# show platform arad ip route

Tunnel Type: M(mpls), G(gre)
* - Routes in LEM
-------------------------------------------------------------------------------------------------
|                              Routing Table                                      |             |
|-----------------------------------------------------------------------------------------------
|VRF|  Destination   |     |                  |    |Acl  |                 |ECMP | FEC | Tunnel
|ID |    Subnet      | Cmd |    Destination   |VID |Label| MAC / CPU Code  |Index|Index|T Value
-------------------------------------------------------------------------------------------------
|0  |0.0.0.0/8       |TRAP |CoppSystemL3DstMiss|0   | -   |ArpTrap          |  -  |1030 |   -
|0  |100.1.0.0/32    |TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive     |  -  |1032 |   -
|0  |100.1.0.0/32    |TRAP |CoppSystemIpUcast  |0   | -   |Receive          |  -  |32766|   -
|0  |100.1.255.255/32|TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive     |  -  |1032 |   -
|0  |200.1.255.255/32|TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive     |  -  |1032 |   -
|0  |200.1.0.0/16    |TRAP |CoppSystemL3DstMiss|1007| -   |ArpTrap          |  -  |1029 |   -
|0  |0.0.0.0/0       |TRAP |CoppSystemL3LpmOver|0   | -   |SlowReceive      |  -  |1024 |   -
|0  |4.4.4.0/24*     |ROUTE|Et10               |1007| -   |00:01:00:02:00:03|  -  |1033 |   -
|0  |10.20.30.0/24*  |ROUTE|Et9                |1006| -   |00:01:00:02:00:03|  -  |1027 |   -
```

13.1.6 IP Source Guard

IP Source Guard (IPSG) prevents IP spoofing attacks.

IP Source Guard (IPSG) filters inbound IP packets based on their source MAC and IP addresses. IPSG is supported in hardware. IPSG enabled on a Layer 2 port verifies IP packets received on this port. Packets are permitted if each packet source MAC and IP addresses match any of the user-configured IP-MAC binding entries on the receiving VLAN and port. Packets with no match are dropped immediately.

13.1.6.1 Configuring IPSG

IPSG is applicable only to Layer 2 ports, and is enabled by the ip verify source command for the configuration mode interface. When configured on Layer 3 ports, IPSG does not take effect until this interface is converted to Layer 2.

IPSG is supported on Layer 2 Port-Channels, not member ports. The IPSG configuration on port channels supersedes the configuration on the physical member ports. Therfore, source IP MAC binding entries should be configured on port channels using the ip source binding command. When configured on a port channel member port, IPSG does not take effect until this port is deleted from the port channel configuration.

Examples

- These configuration commands exclude VLAN IDs ***1*** through ***3*** from IPSG filtering. When enabled on a trunk port, IPSG filters the inbound IP packets on all allowed VLANs. IP packets

received on VLANs ***4*** through ***10*** on ***ethernet 36*** will be filtered by IPSG, while those received on VLANs ***1*** through ***3*** are permitted.

```
switch(config)# no ip verify source vlan 1-3
switch(config)# interface ethernet 36
switch(config-if-Et36)# switchport mode trunk
switch(config-if-Et36)# switchport trunk allowed vlan 1-10
switch(config-if-Et36)# ip verify source
switch(config-if-Et36)#
```

- This configuration command configures source IP-MAC binding entries to IP address ***10.1.1.1***, MAC address ***0000.aaaa.1111***, ***VLAN ID 4094***, and ***interface ethernet 36***.

```
switch(config)# ip source binding 10.1.1.1 0000.aaaa.1111 vlan 4094
 interface ethernet 36
switch(config)#
```

13.1.6.2 DHCP Server Show Commands

Use the **`show dhcp server`** command to display DHCP server information.

- DHCPv4 display example:

```
switch# show dhcp server ipv4
IPv4 DHCP Server is active
Debug log is enabled
DNS server(s): 10.2.2.2
DNS domain name: domainFoo
Lease duration: 1 days 0 hours 0 minutes
TFTP server:
serverFoo (Option 66)
10.0.0.3 (Option 150)
TFTP file: fileFoo
Active Leases: 1
IPv4 DHCP interface status:
   Interface   Status
------------------------------------------------
   Ethernet1   Inactive (Could not determine VRF)
   Ethernet2   Inactive (Not in default VRF)
   Ethernet3   Inactive (Kernel interface not created yet)
   Ethernet4   Inactive (Not up)
   Ethernet5   Inactive (No IP address)
   Ethernet6   Active

Vendor information:
Vendor ID: default
  Sub-options         Data
---------------- ----------------
      1          192.0.2.0, 192.0.2.1

Vendor ID: vendorFoo
  Sub-options       Data
---------------- -----------
      2           192.0.2.2
      3           "Foo"

Subnet: 10.0.0.0/8
Subnet name: subnetFoo
Range: 10.0.0.1 to 10.0.0.10
DNS server(s): 10.1.1.1 10.2.2.2
```

```
Lease duration: 3 days 3 hours 3 minutes
Default gateway address: 10.0.0.3
TFTP server:
subnetServerFoo (Option 66)
10.0.0.4 (Option 150)
TFTP boot file: subnetFileFoo
Active leases: 1
Reservations:
MAC address: 1a1b.1c1d.1e1f
IPv4 address: 10.0.0.1

MAC address: 2a2b.2c2d.2e2f
IPv4 address: 10.0.0.2
```

- For DHCPv6, there are two additional fields in subnet information output, **`Direct`** field and the **`Relay`** field. These two fields specify if the DHCP Server is accepting broadcast or relayed messages.

  The **`Direct`** field displays **`Active`** when the subnet matches the interface with DHCPv6 configured. This indicates the server is accepting broadcast messages.

  The **`Direct`** field displays **Inactive** when there is another existing subnet already matching the interface, or when the subnet matches more than one DHCP configured interface.

  Examples of outputs for the DHCPv6 **`show dhcp server`** command:

  In this example, DHCPv6 is configured with subnet ***fe80::/10*** while being enabled on ***Ethernet1*** with address ***fe80::1/64*** and on ***Ethernet3*** with address ***fe80::2/64***.

```
switch# show dhcp server ipv6
IPv6 DHCP server is active
Debug log is enabled


DNS server(s): fe80::6


DNS domain name: testaristanetworks.com


Lease duration: 1 days 3 hours 30 minutes


Active leases: 0


IPv6 DHCP interface status:


   Interface    Status


--------------- ------


   Ethernet1    Active
```

```
   Ethernet3    Active









Subnet: fe80::/10


Subnet name: foo




Range: fe80::1 to fe80::3
DNS server(s): fe80::4 fe80::5




Direct: Inactive (Multiple interfaces match this subnet: Ethernet1
 Ethernet3)
Relay: Active


Active leases: 0
```

- This example illustrates when multiple subnets match an interface. In this example, DHCPv6 is configured with subnets ***fc00::/7*** and **fe80::/10** while being enabled on ***Ethernet1*** with address ***fe80::1/10*** and ***fc00::1/7***.

```
switch#show dhcp server ipv6
IPv6 DHCP server is active


DNS server(s):  fc00::2


DNS domain name: testaristanetworks.com


Lease duration: 1 days 3 hours 30 minutes


Active leases: 0


IPv6 DHCP interface status:
```

```
   Interface    Status

--------------- ------

   Ethernet1    Active









Subnet: fc00::/7

Subnet name: foo

Range: fc00::1 to fc00::5

DNS server(s): fc00::6 fc00::8




Direct: Inactive (This and other subnets match interface Ethernet1)
Relay: Active

Active leases: 0

Subnet: fe80::/10

Subnet name: bar




Direct: Inactive (This and other subnets match interface Ethernet1)
Relay: Active

Active leases: 0
```

- When a subnet is disabled, the **show dhcp server** command displays the disable message with a reason. The number of active leases of the disabled subnets will be ***0***. In this example, there are overlapping subnets.

```
switch# show dhcp server
IPv4 DHCP Server is active
DNS server(s): 10.2.2.2
Lease duration: 1 days 0 hours 0 minutes
Active Leases: 0
IPv4 DHCP interface status:
   Interface   Status
------------------------------------------------
   Ethernet1   Active

Subnet: 10.0.0.0/24 (Subnet is disabled - overlapping subnet 10.0.0.0/8)
Range: 10.0.0.1 to 10.0.0.10
DNS server(s): 10.3.3.3 10.4.4.4
Default gateway address: 10.0.0.4
Active leases: 0

Subnet: 10.0.0.0/8 (Subnet is disabled - overlapping subnet 10.0.0.0/24)
DNS server(s):
Default gateway address: 10.0.0.3
Active leases: 0
```

- In this example, the display output shows overlapping ranges.

```
switch# show dhcp server
IPv4 DHCP Server is active
DNS server(s): 10.2.2.2
Lease duration: 1 days 0 hours 0 minutes
Active Leases: 0
IPv4 DHCP interface status:
   Interface   Status
------------------------------------------------
   Ethernet1   Active

Subnet: 10.0.0.0/8 (Subnet is disabled - range 10.0.0.9-10.0.0.12 overlaps
 with an existing pool)
Range: 10.0.0.1 to 10.0.0.10
Range: 10.0.0.9 to 10.0.0.12
DNS server(s): 10.3.3.3 10.4.4.4
Default gateway address: 10.0.0.4
Active leases: 0
```

- This example shows duplicate static IP address reservation.

```
Subnet: 10.0.0.0/8 (Subnet is disabled - ipv4-address 10.0.0.11 is reserved
 more than once)
Subnet name:
DNS server(s):
Default gateway address: 10.0.0.3
Active leases: 0
Reservations:
MAC address: 1a1b.1c1d.1e1f
IPv4 address: 10.0.0.11

MAC address: 2a2b.2c2d.2e2f
IPv4 address: 10.0.0.11
```

- Use the **show dhcp server leases** command to display detailed information about the IP addresses allocated by the DHCP Server (including the IP address, the expected end time for that address, the time when the address is handed out, and the equivalent MAC address).

```
switch# show dhcp server leases
10.0.0.10
End: 2019/06/20 17:44:34 UTC
Last transaction: 2019/06/19 17:44:34 UTC
MAC address: 5692.4c67.460a

2000:0:0:40::b



End: 2019/06/20 18:06:33 UTC



Last transaction: 2019/06/20 14:36:33 UTC



MAC address: 165a.a86d.ffac
```

13.1.7 DHCP Server

The router with DHCP Server enabled acts as a server that allocates and delivers network addresses with desired configuration parameters to its hosts.

The DHCP server is based on ISC Kea.

The router with an DHCP Server enabled acts as a server that allocates and delivers network addresses with desired configuration parameters to its hosts.

DHCP Server support includes:

DHCPv4 support includes:

- Configurable on different interfaces: Routed, VLAN, LAG, Sub-interface, and LAG Sub-interface.
- Configurable lease time for allocated network addresses.
- Configurable DNS domain.
- Configurable DNS servers.
- Configurable subnets with parameters:
  - Default gateway
  - DNS servers
  - Ranges
  - Lease time

Additional features for DHCPv4 include:

- Configurable TFTP server
- Configurable TFTP bootfile

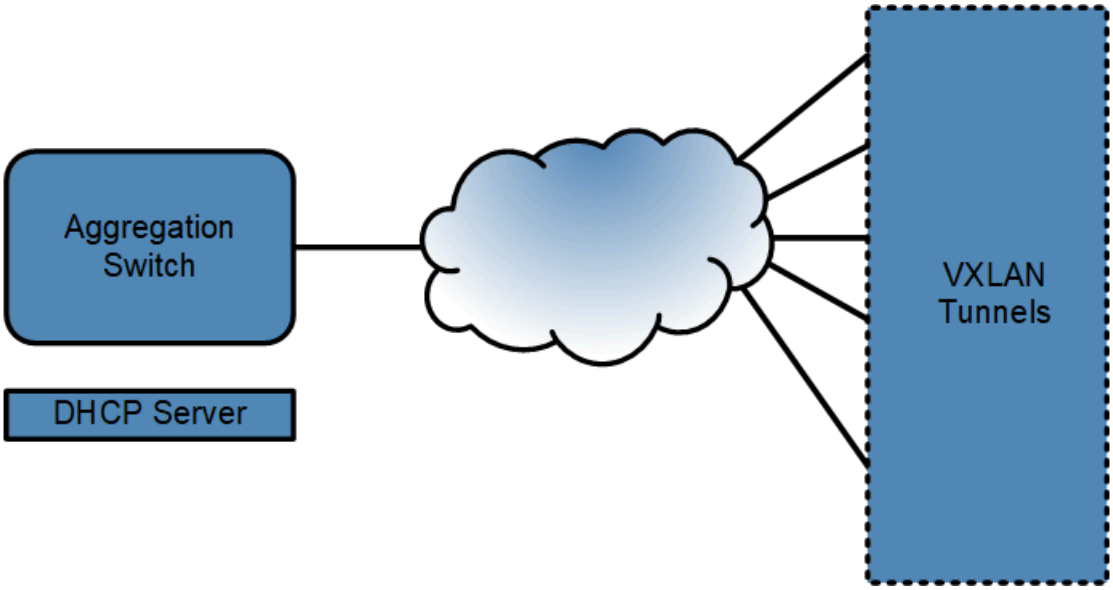
Additional features for DHCPv4 includes:

- Configurable Vendor options with sub options
- Configurable sub option types include: IPv4 address, array of IPv4 addresses, and string
- TFTP bootfile now supports an URI

Additional features for DHCPv4 include a configurable static IP address for exclusive use by a given client, based on the client's MAC address.

Example deployment:

DHCP Server on an aggregation switch, via VXLAN tunnels.



13.1.7.1 Configuring the DHCP Server

You can enable DHCP Server per interface with the IPv4 or IPv6 address family option.

Example

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# dhcp server ipv4

switch(config)# interface Ethernet2
switch(config-if-Et2)# dhcp server ipv6
```

For DHCPv4 to receive and reply to requests, minimum configurations of a valid subnet corresponding to the enabled interface with at least one valid range are required.

Example

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# dhcp server ipv4
switch(config-if-Et1))# ip address 10.0.0.1/8
switch(config-if-Et1)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.0/8
switch(config-if-Et1)# range 10.0.0.4 10.0.0.6
```

At least one non link-local address is required for DHCPv6.

Example

```
switch(config)# interface Vlan1409
```

```
switch(config-if-vlan1409)# dhcp server ipv6
switch(config-dhcp-server)# ipv6 address fe80::1/10

switch(config-if-vlan1409)# dhcp server
   subnet fe80::/10
      range fe80::4 fe80::6
```

To enter DHCP Server global configuration mode ***config-dhcp-server***:

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)#
```

To disable DHCP Server:

```
switch(config-dhcp-server)# disabled
switch(config)#
```

To set the lease time for allocated IP addresses. Lease time is configured globally and applied to all subnets. In the example, the lease time is set to 1 day.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# lease time (ipv4|ipv6) 1 days 0 hours 0 minutes
```

To set the domain name for allocated IP addresses. In this example, the domain name is configured to ***aristanetworks.com***.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# dns domain name (ipv4|ipv6) aristanetworks.com
```

To set the DNS servers. You can only configure up to two servers. In this example, ***10.2.2.2*** for ***DHCPv4*** and ***fe80::2*** for ***DHCPv6***.

```
switch>config
switch(config)# dhcp server
switch(config-dhcp-server)# dns server ipv4 10.2.2.2
switch(config-dhcp-server)# dns server ipv6 fe80::2
```

To set TFTP server. The server can be in the form of either an IP address or a fully qualified domain name. This is only available in DHCPv4. In this example, ***TFTP*** server ***10.0.0.2***.

```
switch>config
switch(config)#dhcp server
switch(config-dhcp-server)#tftp server option 66 ipv4 10.0.0.2
```

To set a list of TFTP servers. The server can only be in the form of an IP address. This is only available in DHCPv4. In this example, ***3.0.0.3*** and ***4.0.0.4*** TFTP servers are configured.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# tftp server option 150 ipv4 10.0.0.3 10.0.0.4
```

To set TFTP server boot file (such as DHCP option 67). This is only available in DHCPv4. In this example, TFTP boot **file bootfile.conf** is configurd.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# tftp server file ipv4 bootfile.conf
```

Or, a TFTP boot file URL **//john.doe@www.example.com:123/example/one**

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# tftp server file ipv4 https://john.doe@www.examp
le.com:123/example/one
```

To set a Vendor option, DHCP option 43, with sub options. Most of the time, different clients need different Vendor options. For this reason option 60, vendor class identifier, is used by the client to identify itself in order to request a specific Vendor option. To enter the Vendor option submode ***(config-dhcp-vendor)*** from ***(config-dhcp-server)*** configuration mode, specify a vendor class identifier. This is only available in DHCPv4.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# vendor-option ipv4 vendorClassIDA
switch(config-dhcp-vendor)#
```

If **default** is specified, the configured Vendor option is sent to those clients requesting for a Vendor option whose vendor class identifier does not match any configured Vendor option. In this example, the default Vendor option for clients without a configured Vendor option.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# vendor-option ipv4 default
```

The Vendor option may contain ***0*** or more sub options. The resulting Vendor option is sent in hexadecimal format to the desired client. In theis example, the Vendor option holding a sub option with IPv4 address ***10.0.0.1***, for clients with vendor class identifier **vendorClassIDA**, resulting in Vendor option ***1:4:a:0:0:1***. The sub option number is ***1***. Length of the Data is ***4***. Data is ***a:0:0:1***.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# vendor-option ipv4 vendorClassIDA
switch(config-dhcp-vendor)# sub-option 1 type ipv4-address data 10.0.0.1
```

In this example, Vendor option holding a sub option with IPv4 addresses ***10.0.0.5*** and ***10.0.0.6***, for clients with vendor class identifier **vendorClassIDA**. Resulting in Vendor option ***fe:4:a:0:0:5:a:0:0:6***.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# vendor-option ipv4 vendorClassIDA
switch(config-dhcp-vendor)# sub-option 254 type ipv4-address data 10.0.0.5
 10.0.0.6
```

In this example, Vendor option holding a sub option with a string ***FOO***, for all clients whose vendor class identifier does not match any configured Vendor option. Resulting in Vendor option ***1e:3:46:4f:4f***.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# vendor-option ipv4 default
switch(config-dhcp-vendor-option-default)# sub-option 30 type string data "FOO"
```

In this example, Vendor option holding two sub options, where sub option 1 holds the IPv4 address ***10.0.0.1***, and sub option 2 holds a string ***FOO***, for all clients whose vendor class identifier does not match any configured Vendor option. Resulting in Vendor option ***1:4:a:0:0:1:2:3:46:4f:4f***.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# vendor-option ipv4 default
switch(config-dhcp-vendor-option-default)# sub-option 1 type ipv4-address data
 10.0.0.1
switch(config-dhcp-vendor-option-default)# sub-option 2 type string data "FOO"
```

Configuration options for subnets are configured under IPv4 or IPv6 address family specific subnet submode ***(config-dhcp-subnet)***. To enter the submode from ***(config-dhcp-server)*** config mode, specify a subnet mask. In this example, ***10.0.0.0/8*** for DHCPv4 and ***fe80::/10*** for DHCPv6.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.0/8
switch(config-dhcp-server)# subnet fe80::/10
```

There can be multiple subnets configured, but they must not overlap. If this happens, all overlapping subnets are disabled.

For DHCPv6, a subnet may match only one interface and vice versa; otherwise no lease is assigned for that subnet.

In this example. the user enables DHCPv6 on ***ethernet1*** (with address ***fe80::1/64***) and ***ethernet3*** (with address ***fe80::2/64***), then configures a subnet ***fe80::/10*** for the DHCPv6:

```
switch(config)# interface ethernet1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# ipv6 address fe80::1/64
switch(config-if-Et1)# dhcp server ipv6

switch(config)# interface Ethernet3
switch(config-if-Et3)# no switchport
switch(config-if-Et3)# ipv6 address fe80::1/64
switch(config-if-Et3)# dhcp server ipv6

switch(config)# dhcp server
switch(config-dhcp-server)# subnet fe80::/1
```

To set default-gateway for a subnet. Only one default-gateway is allowed. This is only available in DHCPv4. In this example, ***10.0.0.3*** ia configured as the default gateway for subnet ***10.0.0.0/8***:

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.0/8
switch(config-dhcp-server-subnet-ipv4)# default-gateway 10.0.0.3
```

To set the DNS servers for a subnet. In this example, ***10.1.1.1*** and ***10.2.2.2*** for DHCPv4 and ***fe80::1*** and ***fe80::2*** for DHCPv6. The number of DNS servers cannot exceed two (2).

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.0/8
switch(config-dhcp-server-subnet-ipv4)# dns server 10.1.1.1 10.2.2.2
switch(config-dhcp-server-subnet-ipv4)# subnet fe80::/10
switch(config-dhcp-server-subnet-ipv6)# dns server fe80::1 fe80::2
```

To set the name of the subnet. In this example, the name ***foo*** for DHCPv4 and name ***bar*** for DHCPv6 are the selected names.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.0/8
switch(config-dhcp-server-subnet-ipv4)# name foo
switch(config-dhcp-server-subnet-ipv4)# subnet fe80::/10
switch(config-dhcp-server-subnet-ipv6)# name bar
```

To set the range of IP addresses of the subnet. In this example, the IP address range for DHCPv4 is ***10.0.0.1*** to ***10.0.0.10*** and for DHCPv6 it is ***fe80::1 fe80::5***. The range must be within the subnet mask, otherwise the subnet is disabled.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.0/8
switch(config-dhcp-server)#range 10.0.0.1 10.0.0.10
switch(config-dhcp-server)# subnet fe80::/10
switch(config-dhcp-server)# range fe80::1 fe80::5
```

To set TFTP server for a subnet. The server can be in the form of either an IP address or a fully qualified domain name. This is only available in DHCPv4. In this example, ***tftpserver.com*** is configured.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.8/24
switch(config-dhcp-server)# tftp server option 66 tftpserver.com
```

To set a list of TFTP servers. The server can only be in the form of an IP address. This is only available in DHCPv4. In this example, TFTP servers ***3.0.0.3 4.0.0.4*** are configured.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# 10.0.0.0/8
switch(config-dhcp-server)# tftp server option 150 3.0.0.3 4.0.0.4
```

To set a TFTP boot file for a subnet. This is only available in DHCPv4. In this example, ***subnet-bootfile.conf*** is configurd.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.0/8
switch(config-dhcp-server)# tftp server file subnet-bootfile.conf
```

To set a static IP address for exclusive use by a client, first enter the client's MAC address reservation submode.

To enter the MAC address reservation submode: ***(config-dhcp-mac-address-ipv4)*** from the ***(config-dhcp-reservations-ipv4)*** configuration mode, then specify the client's MAC address. This is only available in DHCPv4. In this example, we use the static IP address of ***10.0.0.1*** for a client with MAC address of ***1a1b.1c1d.1e1f***.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# subnet 10.0.0.0/8
switch(config-dhcp-server)# reservations
switch(config-dhcp-server)# mac-address 1a1b.1c1d.1e1f
switch(config-dhcp-server)# ipv4-address 10.0.0.1
```

To enable a debug log and save log to a file (this disables kea syslog messages). In this example, you enable **/tmp/debug-dhcp.log** as the debug log.

```
switch> config
switch(config)# dhcp server
switch(config-dhcp-server)# debug log file:/tmp/debug-dhcp.log
```

13.1.7.2 DHCP Server Limitations

The DHCP Server and DHCP Relay cannot be configured on the same router. The DHCP Relay has a higher precedence so the DHCP Server will be disabled when the DHCP Relay is configured.

If DHCPD is enabled on the router, configuring the EOS DHCP Server may cause DHCPD to crash or have unexpected behaviors.

During reload, the DHCP Server is not able to save its leases information. In a modular system, the second supervisor does not have an up-to-date lease database. The clients need to request for new leases. During the synchronizing phase for clients and server, duplicate IP addresses in the network may occur.

The DHCP Server is only supported on default VRF and it currently does not support legacy bootp requests.

All DHCP Server options must be less than 256 bytes in size, otherwise, the DHCP Server will not be able to send the affected option.

On certain platforms VXLAN Recirculation has to be enabled: VXLAN Routing Configuration.

13.1.7.3 DHCP Server Show Commands

Use the **show dhcp server** command to display DHCP server information.

- DHCPv4 display example:

```
switch# show dhcp server ipv4
IPv4 DHCP Server is active
Debug log is enabled
DNS server(s): 10.2.2.2
DNS domain name: domainFoo
Lease duration: 1 days 0 hours 0 minutes
TFTP server:
serverFoo (Option 66)
10.0.0.3 (Option 150)
TFTP file: fileFoo
Active Leases: 1
IPv4 DHCP interface status:
   Interface   Status
------------------------------------------------
   Ethernet1   Inactive (Could not determine VRF)
   Ethernet2   Inactive (Not in default VRF)
   Ethernet3   Inactive (Kernel interface not created yet)
   Ethernet4   Inactive (Not up)
   Ethernet5   Inactive (No IP address)
   Ethernet6   Active

Vendor information:
Vendor ID: default
  Sub-options         Data
---------------- ----------------
      1          192.0.2.0, 192.0.2.1

Vendor ID: vendorFoo
  Sub-options       Data
---------------- -----------
```

```
      2            192.0.2.2
      3            "Foo"

Subnet: 10.0.0.0/8
Subnet name: subnetFoo
Range: 10.0.0.1 to 10.0.0.10
DNS server(s): 10.1.1.1 10.2.2.2
Lease duration: 3 days 3 hours 3 minutes
Default gateway address: 10.0.0.3
TFTP server:
subnetServerFoo (Option 66)
10.0.0.4 (Option 150)
TFTP boot file: subnetFileFoo
Active leases: 1
Reservations:
MAC address: 1a1b.1c1d.1e1f
IPv4 address: 10.0.0.1

MAC address: 2a2b.2c2d.2e2f
IPv4 address: 10.0.0.2
```

- For DHCPv6, there are two additional fields in subnet information output, **`Direct`** field and the **`Relay`** field. These two fields specify if the DHCP Server is accepting broadcast or relayed messages.

  The **`Direct`** field displays **`Active`** when the subnet matches the interface with DHCPv6 configured. This indicates the server is accepting broadcast messages.

  The **`Direct`** field displays **Inactive** when there is another existing subnet already matching the interface, or when the subnet matches more than one DHCP configured interface.

  Examples of outputs for the DHCPv6 **`show dhcp server`** command:

  In this example, DHCPv6 is configured with subnet ***fe80::/10*** while being enabled on ***Ethernet1*** with address ***fe80::1/64*** and on ***Ethernet3*** with address ***fe80::2/64***.

```
switch# show dhcp server ipv6
IPv6 DHCP server is active
Debug log is enabled



DNS server(s): fe80::6



DNS domain name: testaristanetworks.com



Lease duration: 1 days 3 hours 30 minutes



Active leases: 0



IPv6 DHCP interface status:



   Interface    Status
```

```
--------------- ------


   Ethernet1    Active
   Ethernet3    Active









Subnet: fe80::/10


Subnet name: foo




Range: fe80::1 to fe80::3
DNS server(s): fe80::4 fe80::5




Direct: Inactive (Multiple interfaces match this subnet: Ethernet1
 Ethernet3)
Relay: Active


Active leases: 0
```

- This example illustrates when multiple subnets match an interface. In this example, DHCPv6 is configured with subnets ***fc00::/7*** and **fe80::/10** while being enabled on ***Ethernet1*** with address ***fe80::1/10*** and ***fc00::1/7***.

```
switch#show dhcp server ipv6
IPv6 DHCP server is active


DNS server(s):  fc00::2


DNS domain name: testaristanetworks.com


Lease duration: 1 days 3 hours 30 minutes
```

```
Active leases: 0


IPv6 DHCP interface status:


   Interface    Status


--------------- ------


   Ethernet1    Active









Subnet: fc00::/7


Subnet name: foo


Range: fc00::1 to fc00::5


DNS server(s): fc00::6 fc00::8




Direct: Inactive (This and other subnets match interface Ethernet1)
Relay: Active


Active leases: 0

Subnet: fe80::/10


Subnet name: bar




Direct: Inactive (This and other subnets match interface Ethernet1)
```

```
Relay: Active



Active leases: 0
```

- When a subnet is disabled, the **show dhcp server** command displays the disable message with a reason. The number of active leases of the disabled subnets will be ***0***. In this example, there are overlapping subnets.

```
switch# show dhcp server
IPv4 DHCP Server is active
DNS server(s): 10.2.2.2
Lease duration: 1 days 0 hours 0 minutes
Active Leases: 0
IPv4 DHCP interface status:
   Interface   Status
------------------------------------------------
   Ethernet1   Active

Subnet: 10.0.0.0/24 (Subnet is disabled - overlapping subnet 10.0.0.0/8)
Range: 10.0.0.1 to 10.0.0.10
DNS server(s): 10.3.3.3 10.4.4.4
Default gateway address: 10.0.0.4
Active leases: 0

Subnet: 10.0.0.0/8 (Subnet is disabled - overlapping subnet 10.0.0.0/24)
DNS server(s):
Default gateway address: 10.0.0.3
Active leases: 0
```

- In this example, the display output shows overlapping ranges.

```
switch# show dhcp server
IPv4 DHCP Server is active
DNS server(s): 10.2.2.2
Lease duration: 1 days 0 hours 0 minutes
Active Leases: 0
IPv4 DHCP interface status:
   Interface   Status
------------------------------------------------
   Ethernet1   Active

Subnet: 10.0.0.0/8 (Subnet is disabled - range 10.0.0.9-10.0.0.12 overlaps
 with an existing pool)
Range: 10.0.0.1 to 10.0.0.10
Range: 10.0.0.9 to 10.0.0.12
DNS server(s): 10.3.3.3 10.4.4.4
Default gateway address: 10.0.0.4
Active leases: 0
```

- This example shows duplicate static IP address reservation.

```
Subnet: 10.0.0.0/8 (Subnet is disabled - ipv4-address 10.0.0.11 is reserved
 more than once)
Subnet name:
DNS server(s):
Default gateway address: 10.0.0.3
Active leases: 0
Reservations:
MAC address: 1a1b.1c1d.1e1f
IPv4 address: 10.0.0.11
```

```
MAC address: 2a2b.2c2d.2e2f
IPv4 address: 10.0.0.11
```

- Use the **show dhcp server leases** command to display detailed information about the IP addresses allocated by the DHCP Server (including the IP address, the expected end time for that address, the time when the address is handed out, and the equivalent MAC address).

```
switch# show dhcp server leases
10.0.0.10
End: 2019/06/20 17:44:34 UTC
Last transaction: 2019/06/19 17:44:34 UTC
MAC address: 5692.4c67.460a

2000:0:0:40::b



End: 2019/06/20 18:06:33 UTC



Last transaction: 2019/06/20 14:36:33 UTC



MAC address: 165a.a86d.ffac
```

13.1.8 DHCP Relay Global Configuration Mode

The dhcp relay command is configured under the global configuration mode. When configured, it places the switch on DHCP relay mode and allows the user to configure DHCP relay specific configuration on several interfaces with a single command. Furthermore, configuration entered in the DHCP Relay global configuration mode can be overridden by equivalent interface specific commands.

Example

- The **dhcp relay** command places the switch in the DHCP relay configuration mode.

```
switch(config)# dhcp relay
switch(config-dhcp-relay)#
```

- It is possible to specify the IP address of the default DHCP or DHCPv6 Server. Multiple IP addresses can be specified and DHCP requests are forwarded to all specified helper addresses. This is equivalent to configuring **ip helper-address <IP_Address>** under each desired routing interface.
- Example to forward DHCP broadcast packets received on interface ***Ethernet1*** and ***Vlan2*** to DHCP servers at ***10.0.0.1***, ***10.0.0.2***, and to hostname ***DefaultDHCPHostname***:

```
switch(config)# interface ethernet1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# ip address 192.168.1.1/16

switch(config)# interface vlan2
switch(config-if-Et1)# ip address 172.16.1.1/16

switch(config)# dhcp relay
switch(config-dhcp-relay)# server 10.0.0.1
switch(config-dhcp-relay)# server 10.0.0.2
switch(config-dhcp-relay)# server DefaultDHCPHostname
```

- Example to forward DHCPv6 broadcast packet received on interface ***ethernet1*** to DHCPv6 Server at ***fc00::3***.

```
switch(config)# interface ethernet1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# ipv6 address fc00::1/10

switch(config)# dhcp relay
switch(config-dhcp-relay)# server fc00::3
```

- This will point a routed interface to the specified DHCP and DHCPv6 server(s), if all the following criteria are met:
  - The routed interface is in the default VRF.
  - The interface has an IP address configured.
  - The interface is not a Management or Loopback.
- Example to remove the default DHCP or DHCPv6 Server.

```
switch(config)# dhcp relay
switch(config-dhcp-relay)# no server 10.0.0.1
switch(config-dhcp-relay)# no server 10.0.0.2
switch(config-dhcp-relay)# no server DefaultDHCPHostname
switch(config-dhcp-relay)# no server fc00::3
```

To override the default DHCP Server on an interface, **`ip helper-address <IP_Address>`** must be used.

•

Example to forward DHCP broadcast packet received on interface Ethernet1 to DHCP Servers at ***10.0.0.1***, ***10.0.0.2*** and hostname ***DefaultDHCPHostname***, but VLAN2s broadcast packets to DHCP Server at ***10.0.0.3*** only.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# ip address 192.168.1.1/16

switch(config)# interface vlan2
switch(config-if-Et1)# ip address 172.16.1.1/16
switch(config-if-Et1)# ip helper-address 11.0.0.3

switch(config)# dhcp relay
switch(config-dhcp-relay)# server 10.0.0.1
switch(config-dhcp-relay)# server 10.0.0.2
switch(config-dhcp-relay)# server DefaultDHCPHostname
```

To override the default DHCPv6 Server on an interface, **`ipv6 helper-address <IPv6_Address>`** must be used.

Example to forward DHCPv6 broadcast packet received on interface Ethernet1 to DHCPv6 Server at ***fc00::3***, but VLAN2s broadcast packets to DHCPv6 Server at ***fc00::4*** only.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# ipv6 address fc00::1/10

switch(config)# interface vlan2
switch(config-if-Et1)# ipv6 address fc00::2/10
switch(config-if-Et1)# ipv6 helper-address fc00::4

switch(config)# dhcp relay
```

```
switch(config-dhcp-relay)# server fc00::3
```

You can disable DHCP or DHCPv6 Relay functionality from a specific interface. This will disable both DHCP Relay global and interface mode configurations.

- Example to disable DHCP Relay functionality only.

```
switch(config)# interface vlan3
switch(config-if-Et1)# dhcp relay ipv4 disabled
```

- Example to disable DHCPv6 Relay functionality only.

```
switch(config)# interface Vvlan3
switch(config-if-Et1)# dhcp relay ipv6 disabled
```

13.1.8.1 DHCP Relay Global Configuration Mode Limitations

The DHCP Relay global configuration mode's server commands do not apply to:

- Interfaces in non-default VRF.
- Interfaces without an IP address configured.
- Management and Loopback interfaces.

13.1.8.2 DHCP Relay Global Configuration Mode Show Command

Example

This command displays the VRF specifier for the server:

```
switch# show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is enabled
DHCP Smart Relay is disabled
Interface: Ethernet9
Option 82 Circuit ID: Ethernet9
DHCP Smart Relay is disabled
DHCP servers: 10.40.2.3
10.40.2.3:vrf=qchyh-vrf
```

13.1.9 DHCP Relay Across VRF

The EOS DHCP relay agent supports forwarding of DHCP requests to DHCP servers located in a different VRF to the DHCP client interface VRF. In order to enable VRF support for the DHCP relay agent, Option 82 (DHCP Relay Agent Information Option) must first be enabled. The DHCP relay agent uses Option 82 to pass client specific information to the DHCP server.

These sections describe DHCP Relay across VRF features:

- Global Configuration
- DHCP Relay Global Configuration Mode Show Command

The DHCP relay agent inserts Option 82 information into the DHCP forwarded request, which requires the DHCP server belongs to a network on an interface, and that interface belongs to a different VRF than the DHCP client interface. Option 82 information includes the following:

- **VPN identifier:** The VRF name for the ingress interface of the DHCP request, inserted as sub-option 151.

Table 67: VPN Identifier

SubOpt	Len	ASCII VRF Identifier						
151	7	V	R	F	N	A	M	E

- **Link selection:** The subnet address of the interface that receives the DHCP request, inserted as sub-option 5. When the DHCP smart relay is enabled, the link selection is filled with the subnet of the active address. The relay agent will set the Gateway IP address (gIPaddr) to its own IP address so that DHCP messages can be routed over the network to the DHCP server.

Table 68: Link Selection

SubOpt	Len	Subnet IP Address			
5	4	A1	A2	A3	A4

- **Server identifier override:** The primary IP address of the interface that receives the DHCP request, inserted as sub-option 11. When the DHCP smart relay is enabled, the server identifier is filled with the active address (one of the primary or secondary addresses chosen by smart relay mechanism).

Table 69: Link Selection

SubOpt	Len	Overriding Server Identifier Address			
11	4	B1	B2	B3	B4

- VSS control suboption as suboption 152: The DHCP server will strip out this suboption when sending the response to the relay, indicating that the DHCP server used VPN information to allocate IP address.

Note: The DHCP server must be capable of handling VPN identifier information in option 82.

Direct communication between DHCP client and server may not be possible as they are in separate VRFs. The Server identifier override and Link Selection sub-options set the relay agent to act as the DHCP server, and enable all DHCP communication to flow through the relay agent.

The relay agent adds all the appropriate sub-options, and forwards all (including renew and release) request packets to the DHCP server. When the DHCP server response messages are received by the relay, Option 82 information is removed and the response is forwarded to the DHCP client in the client VRF.

13.1.9.1 Global Configuration

The DHCP relay agent information option is inserted in DHCP messages relayed to the DHCP server. The ip helper-address command enables DHCP relay on an interface; and relays DHCP messages to the specified IPv4 address.

Example

This command enables DHCP relay on the ***interface ethernet 1/2***; and relays DHCP messages to the server at ***1.1.1.1***.

```
switch(config)# interface ethernet 1/2
switch(config-if-Et1/2)# ip helper-address 1.1.1.1
```

```
switch(config-if-Et1/2)#
```

The commands provided in examples below will turn on the attachment of VRF-related tags in the relay agent information option. If both the DHCP client interface and server interface are on the same VRF (default or non-default), then no VRF-related DHCP relay agent information option is inserted.

Examples

- This command configures the DHCP relay to add option 82 information.

```
switch(config)# ip dhcp relay information option
```

- These commands configures two new VRF instances and assign them Route Distinguishers (RDs).

```
switch(config)# vrf instance mtxxg-vrf
switch(config-vrf-mtxxg-vrf)# router bgp 50
switch(config-router-bgp)# vrf mtxxg-vrf
switch(config-router-bgp-vrf-mtxxg-vrf)# rd 5546:5546
switch(config)# vrf instance qchyh-vrf
switch(config-vrf-qchyh-vrf)# router bgp 50
switch(config-router-bgp)# vrf qchyh-vrf
switch(config-router-bgp-vrf-qchyh-vrf)# rd 218:218
```

- This command configures an interface connected to DHCP client in vrf ***mtxxg-vrf*** and assigns an IP address.

```
switch(config)# interface ethernet 9
switch(config-if-Et9)# no switchport
```

- This command configures the DHCP client interface in VRF ***mtxxg-vrf***.

```
switch(config-if-Et9)# vrf mtxxg-vrf
switch(config-if-Et9)# ip address 10.10.0.1/16
```

- This command configures the server interface in VRF ***qchyh-vrf.***

```
switch(config-if-Et11)# vrf qchyh-vrf
switch(config-if-Et11)# ip address 10.40.0.1/16
```

- This command configures a helper address for a DHCP server in VRF ***qchyh-vrf***.

```
switch(config-if-Et11)# ip helper-address 10.40.2.3 vrf qchyh-vrf
```

13.1.9.2 DHCP Relay Global Configuration Mode Show Command

Example

This command displays the VRF specifier for the server:

```
switch# show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is enabled
DHCP Smart Relay is disabled
Interface: Ethernet9
Option 82 Circuit ID: Ethernet9
DHCP Smart Relay is disabled
DHCP servers: 10.40.2.3
10.40.2.3:vrf=qchyh-vrf
```

13.1.10 DHCP Relay in VXLAN EVPN

The ip dhcp relay information option (Global) command enables the configuration of the DHCP server to uniquely identify the origin of the request using a source-interface and the helper address. Source interface is configured with a routable address, which is used by the DHCP server to uniquely identify the DHCP relay agent which forwarded the client's request.

Configuring DHCP Relay in VXLAN EVPN (IPv4)

The following enables DHCP relay information option (***Option 82***) required to specify a source interface.

```
switch (config)# ip dhcp relay information option
```

The following configures a Loopback interface as the source interface.

```
switch (config)# interface Loopback1
switch (config-if-Lo1)# ip address 1.1.1.1/24
```

The following configures the Loopback interface as the specified source interface for the helper address.

```
switch (config)# interface vlan100
switch (config-if-Vl100)# ip helper-address 10.1.1.4 source-interface Loopback1
```

The following configures the Loopback interface when the DHCP server is in a different VRF (***red***). The source interface must be configured in the DHCP server's VRF for the command to take effect.

```
switch (config)# interface Loopback3
switch (config-if-Lo3)# vrf red
switch (config-if-Lo3)# ip address 1.1.1.1/24

switch (config)# interface vlan100
switch (config-if-Vl100)# ip helper-address 10.1.1.4 vrf red source-interface
 Loopback3
```

The following disables the use of source interface along with the helper address.

```
switch (config)# interface vlan100
switch (config-if-Vl100)# no ip helper-address 10.1.1.4 source-interface
 Loopback1
```

Configuring DHCP Relay in VXLAN EVPN (IPv6)

The following configures a local interface.

```
switch (config)# interface Loopback2
switch (config-if-Vl100)# ipv6 address 2001::10:20:30:1/128
```

The following configures the Loopback interface as the local interface for the helper address.

```
switch (config)# interface vlan200
switch (config-if-Vl200)# ipv6 dhcp relay destination 2002::10:20:30:2 local-
interface Loopback2
```

The following configures the Loopback interface when the DHCP server is in a different VRF (***red***). The local interface must be configured in the DHCP server's VRF for the command to take effect.

```
switch (config)# interface Loopback4
switch (config-if-Lo4)# vrf red
```

```
switch (config-if-Lo4)# ipv6 address 2001::10:20:30:1/128

switch (config)# interface vlan200
switch (config-if-Vl200)# ipv6 dhcp relay destination 2002::10:20:30:2 vrf red
 local-interface Loopback4
```

The following disables the use of local interface along with the helper address.

```
switch (config-if-Vl200)# no ipv6 dhcp relay destination 2002::10:20:30:2
 local-interface Loopback4
```

The following displays the status of DHCP relay option (***Option 82***) and lists the configured DHCP servers.

```
switch# show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is enabled
DHCP Smart Relay is disabled
Interface: Vlan100
  Option 82 Circuit ID: Vlan100
  DHCP Smart Relay is disabled
  DHCP servers: 10.1.1.4
Interface: Vlan200
  Option 82 Circuit ID: Vlan100
  DHCP Smart Relay is disabled
  DHCP servers: 2002::10:20:30:2
```

Limitations

Client requests up to a rate of ***130*** packets/second are processed.

13.1.11 DHCP Snooping with Bridging

In this configuration, in addition to sending DHCP packets to relay (after adding information option), the packets will also be bridged within the VLAN. In the bridging mode, the switch intercepts DHCP packets, inserts option-82 if not already present, and bridges the packet within the VLAN. This mode of DHCP snooping can be configured without DHCP relay configuration.

13.1.11.1 Configuring DHCP Snooping with Bridging

Following are the steps to configure DHCP snooping with bridging:

1. Enable DHCP snooping feature using the ip dhcp snooping command.

   ```
   switch# ip dhcp snooping
   ```

2. Enable the insertion of option-82 in DHCP request packets using the ip dhcp snooping information option command. By default, option-82 is not enabled and without this DHCP Snooping is not operational.

   ```
   switch# ip dhcp snooping information option
   ```

3. Enable DHCP snooping on the corresponding VLANs using the ip dhcp snooping vlan command. By default, DHCP snooping is not enabled on any VLAN.

   ```
   switch# ip dhcp snooping vlan
   ```

4. Set the circuit-id information that is sent in option-82. By default, Interface name and VLAN ID are sent. Remote circuit-id will always be the MAC address of the relay agent.

   ```
   switch# ip dhcp snooping information option circuit-id type 2 format
   %h:%p  Hostname and interface name
   ```

```
%p:%v  Interface name and VLAN ID
```

5. Enable bridging capabilities of DHCP snooping using the ip dhcp snooping bridging command. This command will enable DHCP snooping with or without DHCP relay configuration.

```
switch# ip dhcp snooping bridging
```

13.1.11.2 DHCP Snooping with Bridging Limitations

- DHCP snooping with bridging is supported only for IPv4.
- When DHCP snooping is configured without bridging configuration, packets with option-82 already present are dropped.

13.1.11.3 DHCP Snooping with Bridging Show Commands

The show ip dhcp snooping displays the DHCP snooping with bridging information.

```
switch# show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is operational
DHCP Snooping is configured on following VLANs:
 650
DHCP Snooping bridging is operational on following VLANs:
 650
Insertion of Option-82 is enabled
 Circuit-id sub-option Type: 0
 Circuit-id format: Interface name:Vlan ID
 Remote-id: 00:1c:73:8d:eb:67 (Switch MAC)
```

13.1.12 TCP MSS Clamping

TCP MSS clamping limits the value of the Maximum Segment Size (MSS) in the TCP header of TCP SYN packets transiting a specified Ethernet or tunnel interface. Setting the MSS ceiling can avoid IP fragmentation in tunnel scenarios by ensuring that the MSS is low enough to account for the extra overhead of GRE and tunnel outer IP headers. TCP MSS clamping can be used when connecting via GRE to cloud providers that require asymmetric routing.

When MSS clamping is configured on an interface, if the TCP MSS value in a SYN packet transiting that interface exceeds the configured ceiling limit it will be overwritten with the configured limit and the TCP checksum will be recomputed and updated.

TCP MSS clamping is handled by default in the software data path, but the process can be supported through hardware configuration to minimize possible packet loss and a reduction in the number of TCP sessions which the switch can establish per second.

13.1.12.1 Cautions

This feature should be used with caution. When the TCP MSS clamping feature is enabled by issuing the tcp mss ceiling command on any routed interface, *all* routed IPv4 TCP SYN packets (TCP packets with the “SYN” flag set) are sent by default to the CPU and switched through software, even on interfaces where no TCP MSS ceiling has been configured, as long as TCP MSS clamping is enabled. This limits the number of TCP sessions that can be established through the switch per second, and, because throughput for software forwarding is limited, this feature can also cause packet loss if the rate at which TCP SYN packets are sent to the CPU exceeds the limits configured in the control-plane policy map.

Packet loss and TCP session reductions can be minimized by enabling TCP MSS clamping in hardware, but only SYN packets in which MSS is the first TCP option are clamped in the hardware data path; other TCP SYN packets are still switched through software.

To disable MSS clamping, the MSS ceiling must be removed from every interface on which it has been configured by issuing the **no tcp mss ceiling** command on each configured interface.

13.1.12.2 Enabling TCP MSS Clamping

There is no global configuration to enable TCP MSS clamping. It is enabled as soon as an MSS ceiling is configured on at least one interface.

13.1.12.3 Disabling TCP MSS Clamping

To disable TCP MSS clamping, the MSS ceiling configuration must be removed from every interface by using the **no** or **default** form of the tcp mss ceiling command on every interface where a ceiling has been configured.

13.1.12.4 Configuring the TCP MSS Ceiling on an Interface

The TCP MSS ceiling limit is set on an interface using the tcp mss ceiling command. This also enables TCP MSS clamping on the switch as a whole.

Note: Configuring a TCP MSS ceiling on any interface enables TCP MSS clamping on the switch as a whole. Without hardware support, clamping routes all TCP SYN packets through software, even on interfaces where no TCP MSS ceiling has been configured. This significantly limits the number of TCP sessions the switch can establish per second, and can potentially cause packet loss if the CPU traffic exceeds control plane policy limits.

On Sand platform switches (Qumran-MX, Qumran-AX, Jericho, Jericho+), the following limitations apply:

- This command works only on egress.
- TCP MSS ceiling is supported on IPv4 unicast packets entering the switch; the configuration has no effect on GRE transit packets.
- The feature is supported only on IPv4 routed interfaces. It is not supported on L2 (switchport) interfaces or IPv6 routed interfaces.
- The feature is not supported for IPv6 packets even if they are going to be tunneled over an IPv4 GRE tunnel.
- The feature is not supported on VXLAN, loopback or management interfaces.
- The feature is only supported on IPv4 unicast packets entering the switch. The configuration has no effect on GRE transit packets or GRE decap, even if the egress interface has a TCP MSS ceiling configured.

Example

- These commands configure ***interface ethernet 5*** as a routed port, then specify a maximum MSS ceiling value of ***1458*** bytes for TCP SYN packets exiting that port.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no switchport
switch(config-if-Et5)# tcp mss ceiling ipv4 1458 egress
switch(config-if-Et5)#
```

- These commands apply TCP MSS clamping at ***1436*** bytes in the egress direction for IPv6 packets:

```
switch(config)# interface ethernet 26
switch(config)# tcp mss ceiling ipv6 1436 egress
```

- These commands apply TCP MSS clamping at ***1476*** bytes for IPv4 packets and ***1436*** bytes for IPv6 packets in egress direction:

```
switch(config)# interface ethernet 27
switch(config)# tcp mss ceiling ipv4 1476 ipv6 1436 egress
```

13.1.12.5 Verifying the TCP MSS Clamping

If TCP MSS ceiling is configured on an interface and if the command **`show cpu counters queue | nz`** is incrementing in **`CoppSystemL3Ttl1IpOptUcast`** field for Tcp packet with Syn flag, then TCP MSS clamping is being performed in Software.

```
switch# show cpu counters queue | nz
Fap0.1:
CoPP Class                     Queue    Pkts   Octets   DropPkts   DropOctets
Aggregate
-------------------------------------------------------------------------------
CoppSystemL3Ttl1IpOptUcast     TC0      1       82       0          0
```

13.1.12.6 Configuring TCP MSS Clamping

Interface Configuration

You can specify the TCP MSS value under the ***interface configuration mode***. The command syntax is shown below:

`tcp mss ceiling` [ipv4 | ipv6] ***64-65515*** egress

The keyword **egress** specifies that the MSS clamping is applied on packets transmitted out on the interface in egress direction.

The following example applies TCP MSS clamping at ***1436*** bytes in the egress direction for IPv4 packets:

```
switch(config)# interface ethernet 25
switch(config)#tcp mss ceiling ipv4 1436 egress
```

the following example applies TCP MSS clamping at ***1436*** bytes in the egress direction for IPv6 packets:

```
switch(config)# interface ethernet 26
switch(config)# tcp mss ceiling ipv6 1436 egress
```

The following example applies TCP MSS clamping at ***1476*** bytes for IPv4 packets and ***1436*** bytes for IPv6 packets in egress direction:

```
switch(config)# interface ethernet 27
switch(config)# tcp mss ceiling ipv4 1476 ipv6 1436 egress
```

Hardware TCP MSS Clamping Configuration

Hardware MSS clamping requires the system TCAM profile to have TCP MSS clamping enabled. You can achieve this by creating a user defined TCAM profile as described below. The **User Defined PMF Profiles - TOI** provides general guidelines on how to create and configure TCAM profiles.

The system TCAM profile must have the feature **`tcp-mss-ceiling ip`** in it in order to use hardware MSS clamping. This is applicable regardless of whether the TCAM profile is copied from an existing profile or created from scratch.

Step 1: Create the user defined TCAM profile

The following example demonstrates copying any source profile and adding the feature **`tcp-mss-ceiling ip`**. In this example, the profile name is ***Pro1*** and the source profile name is ***Source1***.

```
(config)# hardware tcam
(config-hw-tcam)# profile Pro1 copy Source1
(config-hw-tcam-profile-Pro1)# feature tcp-mss-ceiling ip copy system-feature-
source-profile
```

TCP MSS clamping is supported only for IPv4 routed packets. Set the packet type for the feature as follows. This is optional when using **`copy system-feature-source-profile`**. In this example, the system profile name is ***Pro1*** and the feature name is ***Source1***.

```
(config-hw-tcam-profile-Pro1-feature-Source1)# packet ipv4 forwarding routed
```

Set the key size limit to ***160***. This is also optional when the feature is copied from **`system-feature-source-profile`**. In this example, the system profile name is ***Pro1*** and the feature name is ***Source1***.

```
(config-hw-tcam-profile-Pro1-feature-Source1)# key size limit 160
```

Removing unused features to ensure that the TCP MSS TCAM DB is allocated. In this example, the system profile name is ***Pro1*** and the feature name is ***Source1***.

```
(config-hw-tcam-profile-Pro1-feature-Source1)# exit
(config-hw-tcam-profile-Pro1)# no feature mirror ip
(config-hw-tcam-profile-Pro1)# no feature acl port mac
```

Step 2: Apply the user defined TCAM profile to the system.

The following example sets the profile as the system profile under the *hardware tcam* mode. In this example, the system profile name is ***red***.

```
(config-hw-tcam)# system profile red
```

When the system TCAM profile is changed, it is expected that some agents will restart. Also it might be necessary to remove some unused features from the TCAM profile to ensure that the TCP MSS feature gets allocated a TCAM DB. For more information about configuring TCAM profiles, refer to **User Defined PMF Profiles**.

Note: The hardware clamping only works for TCP packets with MSS as the first TCP option. Packets where MSS is not the first TCP option are still trapped to CPU for clamping in software even if the **`feature tcp-mss-ceiling`** is configured in the system TCAM profile.

Backward Compatibility

The **`tunnel mss ceiling`** command which provides the same functionality is deprecated with the introduction of **`tcp mss ceiling`** command. The configuration option **`tunnel mss ceiling`** was available only on GRE tunnel interfaces, while **`tcp mss ceiling`** is supported on other routed IPv4 interfaces as well.

13.1.12.7 TCP MSS Clamping Limitations

- The TCP-MSS Clamping is not supported on L2 (switchport) interfaces.
- The TCP-MSS Clamping is NOT supported on VXLAN, Loopback and Management interfaces.
- The TCP-MSS Clamping is supported only in the Egress direction.
- The TCP-MSS Clamping is only supported on unicast routed packets entering the switch. The configuration has no effect on GRE transit packets and GRE decap case, even if the Egress interface has TCP MSS ceiling configured.

Software TCP MSS Clamping Limitations

- Once the TCP-MSS Clamping is enabled, all routed TCP-SYN packets will be software switched, even on interfaces where there is no TCP-MSS ceiling configuration.
- TCP SYN packets could get dropped under high CPU usage conditions or due to DOS attack protection mechanisms such as PDP/CoPP. These factors could limit the TCP connection establishment rate, i.e new TCP sessions established per second through the switch.

Hardware MSS Clamping Limitations

- Hardware TCP-MSS clamping is not supported with host routes when the clamping is applied on a non-tunnel interface. This limitation does not apply to GRE tunnel interfaces.
- TCP SYN packets where TCP-MSS is not the first TCP option are trapped to CPU for MSS adjustment even in hardware MSS clamping mode.
- Hardware TCP-MSS clamping is not supported for IPv6 packets.

13.1.12.8 Configuring Hardware Support for TCP MSS Clamping

TCP MSS clamping can be supported in hardware, but some packets are still routed through the software data path, and an MSS ceiling value must be configured on each interface where clamping is to be applied.

Hardware support for clamping is accomplished through the use of a user-defined TCAM profile. The TCAM profile can be created from scratch or copied from an existing profile, but in either case it must include the **`tcp-mss-ceiling ip`** feature.

Guidelines

- When the system TCAM profile is changed, some agents will restart.
- To ensure that the TCP MSS feature is allocated a TCAM DB, it may be necessary to remove some unused features from the TCAM profile.
- Hardware TCP MSS clamping only works for TCP packets with MSS as the first TCP option. Other TCP SYN packets are still trapped to the CPU for clamping in software.
- Hardware TCP MSS clamping is not supported with host routes when the clamping is applied on a non-tunnel interface. This limitation does not apply to GRE tunnel interfaces.
- The maximum MSS ceiling limit with hardware MSS clamping is 32727 even though the CLI allows configuration of much larger values.
- For more information on the creation of user-defined TCAM profiles, see https://www.arista.com/en/support/toi/eos-4-20-5f/13977-user-defined-pmf-profile.

To configure hardware support for TCP MSS clamping, create a TCAM profile that includes the tcp mss ceiling feature, then apply it to the system.

13.1.12.8.1 Creating the TCAM Profile

A TCAM profile that supports TCP MSS clamping can be created from scratch, or the feature can be added to a copy of the default TCAM profile. When creating a profile from scratch, care must be taken to ensure that all needed TCAM features are included in the profile.

Modifying a Copy of the Default TCAM Profile

The following commands create a copy of the default TCAM profile, name it **`tcp-mss-clamping`**, and configure it to enable MSS clamping in hardware, then remove some unused features included in the default profile to ensure that there are sufficient TCAM resources for the clamping feature.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile tcp-mss-clamping copy default
switch(config-hw-tcam-profile-tcp-mss-clampingl)# feature tcp-mss-ceiling ip
 copy
system-feature-source-profile
switch(config-hw-tcam-profile-tcp-mss-clamping-feature-tcp-mss-ceiling)# key
size limit 160
switch(config-hw-tcam-profile-tcp-mss-clamping-feature-tcp-mss-c
eiling)# packet
ipv4 forwarding routed
switch(config-hw-tcam-profile-tcp-mss-clamping-feature-tcp-mss-ceiling)# exit

switch(config-hw-tcam-profile-tcp-mss-clamping)# no feature mirror ip
switch(config-hw-tcam-profile-tcp-mss-clamping)# no feature acl port mac
```

```
switch(config-hw-tcam-profile-tcp-mss-clampingl)# exit

switch(config-hw-tcam)# exit

switch(config)#
```

13.1.12.8.2 Applying the TCAM Profile to the System

The following commands enter Hardware TCAM Configuration Mode and set the **tcp-mss-clamping** profile as the system profile.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# system profile tcp-mss-clamping
switch(config-hw-tcam)#
```

13.1.12.8.3 Verifying the TCAM Profile Configuration

The following command displays hardware TCAM profile information to verify that the user-defined TCAM profile has been applied correctly.

```
switch(config)# show hardware tcam profile

Configuration        Status
FixedSystem          tcp-mss-clamping         tcp-mss-clamping

switch(config)#
```

13.1.13 IPv4 GRE Tunneling

GRE tunneling supports the forwarding over IPv4 GRE tunnel interfaces. The GRE tunnel interfaces act as a logical interface that performs GRE encapsulation or decapsulation.

Note: The forwarding over GRE tunnel interface on DCS-7500R is supported only if all the line cards on the system have Jericho family chip-set.

13.1.13.1 Configuring GRE Tunneling Interface

On a Local Arista Switch

```
switch(config)# ip routing
switch(config)# interface Tunnel 10
switch(config-if-Tu10)# tunnel mode gre
switch(config-if-Tu10)# ip address 192.168.1.1/24
switch(config-if-Tu10)# tunnel source 10.1.1.1
switch(config-if-Tu10)# tunnel destination 10.1.1.2
switch(config-if-Tu10)# tunnel path-mtu-discovery
switch(config-if-Tu10)# tunnel tos 10
switch(config-if-Tu10)# tunnel ttl 10
```

On a Remote Arista Switch

```
switch(config)# ip routing
switch(config)# interface Tunnel 10
switch(config-if-Tu10)# tunnel mode gre
switch(config-if-Tu10)# ip address 192.168.1.2/24
switch(config-if-Tu10)# tunnel source 10.1.1.2
```

```
switch(config-if-Tu10)# tunnel destination 10.1.1.1
switch(config-if-Tu10)# tunnel path-mtu-discovery
switch(config-if-Tu10)# tunnel tos 10
switch(config-if-Tu10)# tunnel ttl 10
```

Alternative Configuration for Tunnel Source IPv4 Address

```
switch(config)# interface Loopback 10
switch(config-if-Lo10)# ip add 10.1.1.1/32
switch(config-if-Lo10)# exit

switch(config)# conf terminal
switch(config)# interface Tunnel 10
switch(config-if-Tu10)# tunnel source interface Loopback 10
```

Configuration for Adding an IPv4 Route over the GRE Tunnel Interface

```
switch(config)# ip route 192.168.100.0/24 Tunnel 10
```

Tunnel Mode

Tunnel Mode needs to be configured as gre, for GRE tunnel interface. Default value is **tunnel mode gre**.

IP Address

Configures the IP address for the GRE tunnel interface. The IP address can be used for routing over the GRE tunnel interface. The configured subnet is reachable over the GRE tunnel interface and the packets to the subnet are encapsulated in the GRE header.

Tunnel Source

Specifies the source IP address for the outer IPv4 encapsulation header for packets going over the GRE tunnel interface. The tunnel source IPv4 address should be a valid local IPv4 address configured on the Arista Switch. The tunnel source can also be specified as any routed interface on the Arista Switch. The routed interface's IPv4 address is assigned as the tunnel source IPv4 address.

Tunnel Destination

Specifies the destination IPv4 address for the outer IPv4 encapsulation header for packets going over the GRE tunnel interface. The tunnel destination IPv4 should be reachable from the Arista Switch.

Tunnel Path Mtu Discovery

Specifies if the "Do not Fragment" flag needs to set in the outer IPv4 encapsulation header for packets going over the GRE tunnel interface.

Tunnel TOS

Specifies the Tunnel Type of Service (ToS) value to be assigned to the outer IPv4 encapsulation header for packets going over the GRE tunnel interface. Default TOS value of ***0*** will be assigned if tunnel TOS is not configured.

Tunnel TTL

Specifies the TTL value to the assigned to the outer IPv4 encapsulation header for packet going over the GRE tunnel interface. The TTL value is copied from the inner IPv4 header if tunnel TTL is not configured. The tunnel TTL configuration requires the tunnel Path MTU Discovery to be configured.

13.1.13.2 Displaying GRE tunnel Information

- The following commands display the tunnel configuration.

```
switch# show interfaces Tunnel 10
Tunnel10 is up, line protocol is up (connected)
 Hardware is Tunnel, address is 0a01.0101.0800
 Internet address is 192.168.1.1/24
 Broadcast address is 255.255.255.255
 Tunnel source 10.1.1.1, destination 10.1.1.2
 Tunnel protocol/transport GRE/IP
   Key disabled, sequencing disabled
   Checksumming of packets disabled
 Tunnel TTL 10, Hardware forwarding enabled
 Tunnel TOS 10
 Path MTU Discovery
 Tunnel transport MTU 1476 bytes
 Up 3 seconds
```

-
```
switch# show gre tunnel static

Name     Index  Source   Destination  Nexthop  Interface
-------- ------ -------- ------------ -------- -----------
Tunnel10 10     10.1.1.1 10.1.1.2     10.6.1.2 Ethernet6/1

switch# show tunnel fib static interface gre 10
Type 'Static Interface', index 10, forwarding Primary
   via 10.6.1.2, 'Ethernet6/1'
      GRE, destination 10.1.1.2, source 10.1.1.1, ttl 10, tos 0xa
```

- Use the **show platform fap tcam summary** command to verify if the TCAM bank is allocated for GRE packet termination lookup.

```
switch# show platform fap tcam summary

           Tcam Allocation (Jericho0)
Bank        Used By     Reserved By
---------- ------------ -----------
0          dbGreTunnel   -
```

- Use the **show ip route** command to verify if the routes over tunnel is setup properly.

```
switch# show ip route

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort is not set

 C      192.168.1.0/24 is directly connected, Tunnel10, Static Interface GRE
 tunnel
index 10, dst 10.1.1.2, src 10.1.1.1, TTL 10, TOS 10
 S      192.168.100.0/24 is directly connected, Tunnel10, Static Interface
 GRE
```

```
tunnel index 10, dst 10.1.1.2, src 10.1.1.1, TTL 10, TOS 10
```

- The following commands are used to verify the tunnel encapsulation programming.

```
switch# show platform fap eedb ip-tunnel gre interface Tunnel 10

-------------------------------------------------------------------------------
|                                                  Jericho0                   |
|                               GRE Tunnel Egress Encapsulation DB

|
|-------------------------------------------------------------------------------|
| Bank/ | OutLIF | Next   | VSI  | Encap | TOS  | TTL | Source | Destination|
OamLIF| OutLIF | Drop|
| Offset|        | OutLIF | LSB  | Mode  |      |     | IP     | IP         | Set
| Profile|     |
|-------------------------------------------------------------------------------|
| 3/0   | 0x6000 | 0x4010 | 0    | 2     | 10   | 10  | 10.1.1.1 | 10.1.1.2 | No
| 0      | No |

switch# show platform fap eedb ip-tunnel

-------------------------------------------------------------------------------
|                                                  Jericho0                    |
|                                  IP Tunnel Egress Encapsulation DB

|
|-------------------------------------------------------------------------------|
| Bank/ | OutLIF | Next   | VSI | Encap| TOS | TTL | Src | Destination | OamLIF
| OutLIF  | Drop|
| Offset|        | OutLIF | LSB | Mode | Idx | Idx | Idx | IP          | Set    |
Profile |     |
|-------------------------------------------------------------------------------|
| 3/0   | 0x6000 | 0x4010 | 0   | 2    | 9   | 0   | 0   | 10.1.1.2    | No     |
0       | No |
```

13.1.14 GRE Tunneling Support

GRE tunneling supports the forwarding over IPv4 GRE tunnel interfaces. The GRE tunnel interfaces act as a logical interface that performs GRE encapsulation or decapsulation. A maximum of 256 GRE-tunnel interfaces are supported.

Note: GRE keepalives are not supported.

To configure a local Arista switch on a GRE-tunnel interface, consider the following an example.

```
switch(config)# ip routing
switch(config)# interface Tunnel 10
switch(config-if-Tu10)# tunnel mode gre
switch(config-if-Tu10)# ip address 192.168.1.1/24
switch(config-if-Tu10)# tunnel source 10.1.1.1
switch(config-if-Tu10)# tunnel destination 10.1.1.2
switch(config-if-Tu10)# tunnel path-mtu-discovery
switch(config-if-Tu10)# tunnel tos 10
switch(config-if-Tu10)# tunnel ttl 10
```

To configure a remote Arista switch on a GRE-tunnel interface, consider the following an example.

```
switch(config)# ip routing
switch(config)# interface Tunnel 10
switch(config-if-Tu10)# tunnel mode gre
switch(config-if-Tu10)# ip address 192.168.1.2/24
switch(config-if-Tu10)# tunnel source 10.1.1.2
switch(config-if-Tu10)# tunnel destination 10.1.1.1underlayVrf
switch(config-if-Tu10)# tunnel path-mtu-discovery
switch(config-if-Tu10)# tunnel tos 10
```

```
switch(config-if-Tu10)# tunnel ttl 10
```

To add a IPv4 route over the GRE-tunnel interface, configure simulare to the following.

```
switch(config)# ip route 192.168.100.0/24 Tunnel 10
```

Note: IPv6 GRE-Tunnels are not supported. This is only a data-plane limitation whereas IS-IS IPv6 (such as control-plane) can still work.

Use the **show interfaces Tunnel** command to display the interface tunnel.

```
switch(config)# show interfaces Tunnel 10
Tunnel10 is up, line protocol is up (connected)
  Hardware is Tunnel, address is 0a01.0101.0800
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  Tunnel source 10.1.1.1, destination 10.1.1.2
  Tunnel protocol/transport GRE/IP
   Key disabled, sequencing disabled
   Checksumming of packets disabled
  Tunnel TTL 10, Hardware forwarding enabled
  Tunnel TOS 10
  Path MTU Discovery
  Tunnel transport MTU 1476 bytes
  Tunnel underlay VRF "underlayVrf"
  Up 3 seconds
```

Use the **show gre tunnel static** command to display a static interface tunnel.

```
switch(config)#show gre tunnel static
Name        Index      Source         Destination       Nexthop     Interface
----------- -------    -----------    -------------     ----------  ----------
Tunnel10    10         10.1.1.1       10.1.1.2          10.6.1.2    Ethernet6/1
```

Use the **show tunnel fib static interface** command to display a fib static interface tunnel.

```
switch(config)# show tunnel fib static interface gre 10
Type 'Static Interface', index 10, forwarding Primary
   via 10.6.1.2, 'Ethernet6/1'
      GRE, destination 10.1.1.2, source 10.1.1.1, ttl 10, tos 0xa
```

Tunnel Mode

Tunnel mode is **GRE** for a GRE-tunnel interface which is also the default tunnel mode.

IP address

Use this IP address for routing over the GRE-tunnel interface. The configuration subnet is reachable over the GRE-tunnel interface, and the packets to the subnet is encapsulated with the GRE header.

Tunnel Source

Specifies the source IP address for the encapsulating IPv4 header of a packet going over the GRE-tunnel interface. The tunnel source IPv4 address is a valid local IPv4 address configured on the Arista switch. It uses any route interface on the Arista switch. The routed interfaces IPv4 address assigns the tunnel source IPv4 address. Maximum of 16 unique tunnel source IPv4 addresses are supported across all GRE-tunnel interfaces.

The following is an example of an interface as a Tunnel source.

```
switch(config)# interface Loopback 10
```

```
switch(config-if-Lo10)# ip add 10.1.1.1/32
switch(config-if-Lo10)# exit
switch(config)# interface Tunnel 10
switch(config-if-Tu10)#  tunnel source interface Loopback 10
```

Note: Coexistence of GRE-tunnel interfaces and Decap-Groups is not supported.

Note: Coexistence of GRE-tunnel interfaces and VxLan is not supported.

Note: GRE-tunnel is not supported with MLAG configuration.

Tunnel Destination

Specifies the destination IPv4 address for the encapsulating IPv4 header of a packet going over the GRE-tunnel interface. The tunnel destination IPv4 is reachable from the Arista switch.

Note: Multicast traffic over GRE-Tunnels is not supported.

Tunnel Path MTU Discovery

The tunnel path Maximum Transmition Unit (MTU) Discovery specifies if the Don't Fragment (DF) flag needs to be set in the encapsulating IPv4 header of a packet going over the GRE-Tunnel interface. MTU configuration on the GRE-tunnel interface is used by control plane protocols and not enforced in hardware for packets forwarded in data-plane. The MTU change on the tunnel interface does not take effect until the tunnel interface is flapped.

Tunnel TOS

The Tunnel TOS specifies the TOS value to be set in the encapsulating IPv4 header of a packet going over the GRE-Tunnel interface. The default value of ***0*** is assigned if tunnel TOS is not configured. Maximum of seven unique tunnel TOS values are supported across all GRE-tunnel interfaces.

Tunnel TTL

The Tunnel TTL specifies the TTL value to be set in the encapsulating IPv4 header of a packet going over the GRE-tunnel interface. The TTL value is copied from the inner IPv4 header if tunnel TTL is not configured. The tunnel TTL configuration requires the tunnel path MTU discovery to be configured. Maximum of four unique tunnel TTL values are supported across all GRE-tunnel interfaces.

VRF Forwarding (Overlay VRF)

The following configuration is an example of overlay VRF, for a GRE tunnel interface.

```
switch(config)# vrf instance overlayVrf
switch(config)# ip routing vrf overlayVrf
switch(config)# interface Tunnel 10
switch(config-if-Tu10)# vrf overlayVrf
```

Note: Both the tunnels source and destination address must be in the underlay VRF. GRE key forwarding is not supported.

The following is an example of a static route configuration, with an overlay VRF.

```
switch(config)# ip route vrf overlayVrf 7.7.7.0/24 192.168.1.2
```

VRF Forwarding (Underlay VRF)

The following is an configuration example of a underlay VRF for a GRE tunnel interface.

```
switch(config)# vrf instance underlayVrf
switch(config)# interface Tunnel 10
switch(config-if-Tu10)# tunnel underlay vrf underlayVrf
```

13.1.14.1 TCAM Bank Allocation

Note: Command to check if Ternary Content-Addressable Memory (TCAM) bank is allocated for GRE packet termination lookup.

```
switch(config)# show platform fap tcam summary

           Tcam Allocation (Jericho0)
Bank       Used By                   Reserved By
---------- ------------------------- -----------
0           dbGreTunnel               -
```

PBR is not supported on GRE terminated packets.

Verifing Tunnel Routes

Use the **show ip route** command to check if the routes over tunnel is setup correctly.

```
switch(config)# show ip route
VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort is not set

 C      192.168.1.0/24 is directly connected, Tunnel10, Static Interface GRE-
Tunnel index 10, dst 10.1.1.2, src 10.1.1.1, TTL 10, TOS 10
 S      192.168.100.0/24 is directly connected, Tunnel10, Static Interface GRE-
Tunnel index 10, dst 10.1.1.2, src 10.1.1.1, TTL 10, TOS 10
```

Verifing Tunnel Encap

Use the **show platform fap eedb ip-tunnel gre interface Tunnel** command to check the tunnel encap programming on the GRE interface.

```
switch(config)# show platform fap eedb ip-tunnel gre interface Tunnel 10
 ------------------------------------------------------------------------------------------------------------------
|                                          Jericho0                                                                |
|                              GRE Tunnel Egress Encapsulation DB                                                  |
|-----------------------------------------------------------------------------------------------------------------|
| Bank/    | OutLIF  | Next    | VSI   | Encap | TOS  | TTL  | Source    | Destination     | OamLIF | OutLIF  | Drop |
| Offset   |         | OutLIF  | LSB   | Mode  |      |      | IP        | IP              | Set    | Profile |      |
|-----------------------------------------------------------------------------------------------------------------|
| 3/0      | 0x6000  | 0x4010  | 0     | 2     | 10   | 10   | 10.1.1.1  | 10.1.1.2        | No     | 0       | No   |
```

Use the **show platform fap eedb ip-tunnel** command to check the tunnel encap programming on the IP-tunnel interface.

```
switch(config)# show platform fap eedb ip-tunnel
```

```
 ------------------------------------------------------------------------------------------------------------
|                                                Jericho0                                                    |
|                                    IP Tunnel Egress Encapsulation DB                                       |
|------------------------------------------------------------------------------------------------------------|
| Bank/   | OutLIF  | Next    | VSI   | Encap | TOS | TTL | Src | Destination     | OamLIF | OutLIF  | Drop |
| Offset  |         | OutLIF  | LSB   | Mode  | Idx | Idx | Idx | IP              | Set    | Profile |      |
|------------------------------------------------------------------------------------------------------------|
| 3/0     | 0x6000  | 0x4010  | 0     | 2     | 9   | 0   | 0   | 10.1.1.2        | No     | 0       | No   |
```

Verifying Tunnel VRF

Use the **show ip interface tunnel** command to check the overlay VRF.

```
switch(config)# show ip interface tunnel 10
Tunnel10 is up, line protocol is up (connected)
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  IPv6 Interface Forwarding : None
  Proxy-ARP is disabled
  Local Proxy-ARP is disabled
  Gratuitous ARP is ignored
  IP MTU 1476 bytes
  VPN Routing/Forwarding "overlayVrf"
switch(config)# show ip route vrf overlayVrf

VRF: overlayVrf
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

Gateway of last resort is not set

 C        1.1.1.0/24 is directly connected, Ethernet1
 S        7.7.7.0/24 [1/0] via 192.168.1.2, Tunnel10, Static Interface GRE-
Tunnel index 10, dst 10.1.1.2, src 10.1.1.1
 C        192.168.1.0/24 is directly connected, Tunnel10, Static Interface GRE-
Tunnel index 10, dst 10.1.1.2, src 10.1.1.1
```

Tunnel underlay VRF Configuration

Use the **show interfaces Tunnel** command to check the underlay VRF.

```
switch(config)# show interfaces Tunnel 10
Tunnel10 is up, line protocol is up (connected)
  Hardware is Tunnel, address is 0a01.0101.0800
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  Tunnel source 10.1.1.1, destination 10.1.1.2
  Tunnel protocol/transport GRE/IP
   Key disabled, sequencing disabled
   Checksumming of packets disabled
  Tunnel TTL 10, Hardware forwarding enabled
  Tunnel TOS 10
  Path MTU Discovery
  Tunnel transport MTU 1476 bytes
  Tunnel underlay VRF "underlayVrf"
  Up 3 seconds
```

Use the **show ip route vrf underlayVrf** command to check the IP route VFR underlayVRF.

```
switch(config)# show ip route vrf underlayVrf
VRF: underlayVrf
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked,

Gateway of last resort is not set

 C        10.1.1.0/24 is directly connected, Ethernet1
```

13.1.15 BfRuntime to use non-default VRFs

Use the following to configure the VRF for the BfRuntime connection for the management interface on the switches that support it. The management interface may be configured on a different VRF from the default one.

Configuring BfRuntime to use Non-default VRFs

The **platform barefoot bfrt vrf** command configures the forwarding plane agent to restart and listen on the configured VRFs for connections.

```
(config)# platform barefoot bfrt vrf <VRF name>
```

If left unconfigured, the default VRF is used for the IP and port for the the BfRuntime server.

The following shows a typical configuraiton.

```
(config)# vrf instance management
(config-vrf-management)# exit
(config)# platform barefoot bfrt 0.0.0.0 50052
(config)# platform barefoot bfrt vrf <VRF name>
(config)# int management1
(config-if-Ma1)# vrf management
```

Displaying BfRuntime Configuration

The **show platform barefoot bfrt** command displays the existing configuration for the BfRuntime server.

```
(switch)# show platform barefoot bfrt
Namespace: management
FixedSystem:0.0.0.0:50052
```

13.1.16 IPv4 Commands

IP Routing and Address Commands

- agent SandL3Unicast terminate
- clear arp inspection statistics
- compress

- ip arp inspection limit
- ip arp inspection logging
- ip arp inspection trust
- ip arp inspection vlan
- ip hardware fib ecmp resilience
- ip hardware fib optimize
- ip hardware fib next-hop resource optimization
- ip icmp redirect
- ip load-sharing
- ip route
- ip routing
- ip source binding
- ip verify
- ip verify source
- ipv4 routable 240.0.0.0/4
- rib fib policy
- show dhcp server
- show hardware capacity
- show ip
- show ip arp inspection vlan
- show ip arp inspection statistics
- show ip hardware fib summary
- show ip interface
- show ip interface brief
- show ip route
- show ip route age
- show ip route gateway
- show ip route host
- show ip route match tag
- show ip route summary
- show ip verify source
- show platform arad ip route
- show platform arad ip route summary
- show rib route ip
- show rib route fib policy excluded
- show rib route summary
- show routing-context vrf
- show vrf
- tcp mss ceiling

IPv4 DHCP Relay

- clear ip dhcp relay counters
- dhcp relay
- ip dhcp relay all-subnets
- ip dhcp relay all-subnets default
- ip dhcp relay always-on
- ip dhcp relay information option (Global)
- ip dhcp relay information option circuit-id
- ip helper-address
- show ip dhcp relay

- show ip dhcp relay counters

IPv4 DHCP Snooping

- clear ip dhcp snooping counters
- ip dhcp snooping
- ip dhcp snooping bridging
- ip dhcp snooping information option
- ip dhcp snooping vlan
- show ip dhcp snooping
- show ip dhcp snooping counters
- show ip dhcp snooping hardware

IPv4 Multicast Counters

- clear ip multicast count
- ip multicast count

ARP Table

- arp
- arp aging timeout
- arp cache persistent
- arp gratuitous accept
- clear arp-cache
- clear arp
- ip local-proxy-arp
- ip proxy-arp
- show arp
- show ip arp

VRF Commands

- cli vrf
- description (VRF)
- platform barefoot bfrt vrf
- show platform barefoot bfrt
- show routing-context vrf
- show vrf
- vrf (Interface mode)
- vrf instance

Trident Forwarding Table Commands

- platform trident forwarding-table partition
- platform trident routing-table partition
- show platform trident forwarding-table partition

IPv4 GRE Tunneling Commands

- interface tunnel
- show interface tunnel
- show platform fap eedb ip-tunnel gre interface tunnel
- show platform fap tcam summary

- show tunnel fib static interface gre
- tunnel

13.1.16.1 agent SandL3Unicast terminate

The **agent SandL3Unicast terminate** command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

Command Mode

Global Configuration

Command Syntax

agent SandL3Unicast terminate

Related Commands

- ip hardware fib optimize enables IPv4 route scale.
- show platform arad ip route shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.
- show platform arad ip route summary shows hardware resource usage of IPv4 routes.

Example

This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)# agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

13.1.16.2 arp

The **arp** command adds a static entry to an Address Resolution Protocol (ARP) cache. The switch uses ARP cache entries to correlate 32-bit IP addresses to 48-bit hardware addresses.

The **no arp** and **default arp** commands remove the ARP cache entry with the specified IP address. When multiple VRFs contain ARP cache entries for identical IP addresses, each entry can only be removed individually.

Command Mode

Global Configuration

Command Syntax

arp [**VRF_INSTANCE**] ***ipv4_addr mac_addr*** **arpa**

no arp [**VRF_INSTANCE**] ***ipv4_addr***

default arp [**VRF_INSTANCE**] ***ipv4_addr***

Parameters

- **VRF_INSTANCE** Specifies the VRF instance being modified.
  - ***no parameter*** Changes are made to the default VRF.
  - **vrf *vrf_name*** Changes are made to the specified user-defined VRF.
- ***ipv4_addr*** IPv4 address of ARP entry.

- ***mac_addr*** Local data-link (hardware) address (48-bit dotted hex notation – H.H.H).

Examples

- This command adds a static entry to the ARP cache in the default VRF.

```
switch(config)# arp 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

- This command adds the same static entry to the ARP cache in the VRF named ***purple***.

```
switch(config)# arp vrf purple 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

13.1.16.3 arp aging timeout

The **`arp aging timeout`** command specifies the duration of dynamic address entries in the Address Resolution Protocol (ARP) cache for addresses learned through the configuration mode interface. The default duration is ***14400*** seconds (four hours).

The **`arp aging timeout`** and **`default arp aging timeout`** commands restores the default ARP aging timeout for addresses learned on the configuration mode interface by deleting the corresponding **`arp aging timeout`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`arp aging timeout` ***arp_time***

`no arp aging timeout`

`default arp aging timeout`

Parameters

arp_time ARP aging timeout period (seconds). Values range from **`60`** to **`65535`**. Default value is ***14400***.

Example

This command specifies an ARP cache duration of ***7200*** seconds (two hours) for dynamic addresses added to the ARP cache that were learned through ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# arp aging timeout 7200
switch(config-if-Vl200)# show active
interface Vlan200
   arp aging timeout 7200
switch(config-if-Vl200)#
```

13.1.16.4 arp cache persistent

The **arp cache persistent** command restores the dynamic entries in the Address Resolution Protocol (ARP) cache after reboot.

The **no arp cache persistent** and **default arp cache persistent** commands remove the ARP cache persistent configuration from the running-config.

Command Mode

Global Configuration

Command Syntax

arp cache persistent

no arp cache persistent

default arp cache persistent

Example

This command restores the ARP cache after reboot.

```
switch(config)# arp cache persistent
switch(config)#
```

13.1.16.5 arp gratuitous accept

The **arp gratuitous accept** command configures the configuration mode interface to accept gratuitous ARP request packets received on that interface. Accepted gratuitous ARP requests are then learned by the ARP table.

The no and **default** forms of the command prevent the interface from accepting gratuitous ARP requests. Configuring gratuitous ARP acceptance on an L2 interface has no effect.

Command Mode

Interface-Ethernet Configuration

Interface-VLAN Configuration

Interface Port-channel Configuration

Command Syntax

arp gratuitous accept

no arp gratuitous accept

default arp gratuitous accept

Example

These commands configure ***interface ethernet 2/1*** to accept gratuitous ARP request packets.

```
switch(config)# interface ethernet 2/1
switch(config-if-Et2/1)# arp gratuitous accept
switch(config-if-Et2/1)#
```

13.1.16.6 clear arp inspection statistics

The **`clear arp inspection statistics`** command clears ARP inspection statistics.

Command Mode

EXEC

Command Syntax

`clear arp inspection statistics`

Related Commands

- ip arp inspection limit
- ip arp inspection logging
- ip arp inspection trust
- ip arp inspection vlan
- show ip arp inspection vlan
- show ip arp inspection statistics

Example

This command clears ARP inspection statistics.

```
switch(config)# clear arp inspection statistics
switch(config)#
```

13.1.16.7 clear arp

The **`clear arp`** command removes the specified dynamic ARP entry for the specified IP address from the Address Resolution Protocol (ARP) table.

Command Mode

Privileged EXEC

Command Syntax

`clear arp` [**VRF_INSTANCE**] ***ipv4_addr*** **Parameters**

- **VRF_INSTANCE** Specifies the VRF instance for which arp data is removed.
  - ***no parameter*** Specifies the context-active VRF.
  - **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **`default`**.
- ***ipv4_addr*** IPv4 address of dynamic ARP entry.

Example

These commands display the ARP table before and after the removal of dynamic ARP entry for IP address ***172.22.30.52***.

```
switch# show arp

Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.52            0  0025.900e.c468  Management1
172.22.30.53            0  0025.900e.c63c  Management1
172.22.30.133           0  001c.7304.3906  Management1

switch# clear arp 172.22.30.52
```

```
switch# show arp

Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.53            0  0025.900e.c63c  Management1
172.22.30.133           0  001c.7304.3906  Management1 switch#
```

13.1.16.8 clear arp-cache

The **clear arp-cache** command refreshes dynamic entries in the Address Resolution Protocol (ARP) cache. Refreshing the ARP cache updates current ARP table entries and removes expired ARP entries not yet deleted by an internal, timer-driven process.

The command, without arguments, refreshes ARP cache entries for all enabled interfaces. With arguments, the command refreshes cache entries for the specified interface. Executing **clear arp-cache** for all interfaces can result in extremely high CPU usage while the tables are resolving.

Command Mode

Privileged EXEC

Command Syntax

clear arp-cache [**VRF_INSTANCE**][**INTERFACE_NAME**]

Parameters

- **VRF_INSTANCE** Specifies the VRF instance for which arp data is refreshed.
  - ***no parameter*** Specifies the context-active VRF.
  - **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **default**.
- **INTERFACE_NAME** Interface upon which ARP cache entries are refreshed. Options include:
  - ***no parameter*** All ARP cache entries.
  - **interface ethernet *e_num*** ARP cache entries of specified Ethernet interface.
  - **interface loopback *l_num*** ARP cache entries of specified loopback interface.
  - **interface management *m_num*** ARP cache entries of specified management interface.
  - **interface port-channel *p_num*** ARP cache entries of specified port-channel Interface.
  - **interface vlan *v_num*** ARP cache entries of specified VLAN interface.
  - **interface vxlan *vx_num*** VXLAN interface specified by ***vx_num***.

Related Commands

The cli vrf command specifies the context-active VRF.

Example

These commands display the ARP cache before and after ARP cache entries are refreshed.

```
switch# show arp

Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.118           0  001c.7301.6015  Management1

switch# clear arp-cache
switch# show arp

Address         Age (min)  Hardware Addr   Interface
```

```
172.22.30.1              0  001c.730b.1d15  Management1 switch#
```

13.1.16.9 clear ip dhcp relay counters

The **clear ip dhcp relay counters** command resets the DHCP relay counters. The configuration mode determines which counters are reset:

Interface configuration: command clears the counter for the configuration mode interface.

Command Mode

Privileged EXEC

Command Syntax

clear ip dhcp relay counters [**INTERFACE_NAME**]

Parameters

INTERFACE_NAME Entity for which counters are cleared. Options include:

- ***no parameter*** Clears counters for the switch and for all interfaces.
- **interface ethernet *e_num*** Clears counters for the specified Ethernet interface.
- **interface loopback *l_num*** Clears counters for the specified loopback interface.
- **interface port-channel *p_num*** Clears counters for the specified port-channel Interface.
- **interface vlan *v_num*** Clears counters for the specified VLAN interface.

Examples

- These commands clear the DHCP relay counters for ***vlan 1045*** and shows the counters before and after the **clear** command.

```
switch# show ip dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
 Vlan1001 |  207  148    0 | 4 days, 19:54:24 ago
 Vlan1045 |  376  277    0 | 4 days, 19:54:24 ago

switch# clear ip dhcp relay counters interface vlan 1045

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  380  380    0 | 4 days, 21:19:17 ago
 All Resp |  281  281    0 |
          |                |
 Vlan1000 |  207  148    0 | 4 days, 21:18:30 ago
 Vlan1045 |    0    0    0 |          0:00:07 ago
```

- These commands clear all DHCP relay counters on the switch.

```
switch(config-if-Vl1045)# exit
switch(config)# clear ip dhcp relay counters
switch(config)# show ip dhcp relay counters

          |  Dhcp Packets  |
```

```
Interface | Rcvd Fwdd Drop | Last Cleared
----------|----- ---- -----|-------------
  All Req |    0    0    0 |  0:00:03 ago
 All Resp |    0    0    0 |
          |                 |
 Vlan1000 |    0    0    0 |  0:00:03 ago
 Vlan1045 |    0    0    0 |  0:00:03 ago
```

13.1.16.10 clear ip dhcp snooping counters

The **`clear ip dhcp snooping counters`** command resets the DHCP snooping packet counters.

Command Mode

Privileged EXEC

Command Syntax

`clear ip dhcp snooping counters` [**COUNTER_TYPE**]

Parameters

COUNTER_TYPE The type of counter that the command resets. Options include:

- ***no parameter*** Counters for each VLAN.
- **debug** Aggregate counters and drop cause counters.

Examples

- This command clears the DHCP snooping counters for each VLAN.

```
switch# clear ip dhcp snooping counters
switch# show ip dhcp snooping counters

     | Dhcp Request Pkts | Dhcp Reply Pkts |
Vlan |  Rcvd  Fwdd  Drop | Rcvd Fwdd  Drop | Last Cleared
-----|------ ----- ------|----- ---- ------|-------------
 100 |     0     0     0 |    0    0     0 |  0:00:10 ago switch#
```

- This command clears the aggregate DHCP snooping counters.

```
switch# clear ip dhcp snooping counters debug
switch# show ip dhcp snooping counters debug

Counter                       Snooping to Relay Relay to Snooping
----------------------------- ----------------- -----------------
Received                                      0                 0
Forwarded                                     0                 0
Dropped - Invalid VlanId                      0                 0
Dropped - Parse error                         0                 0
Dropped - Invalid Dhcp Optype                 0                 0
Dropped - Invalid Info Option                 0                 0
Dropped - Snooping disabled                   0                 0

Last Cleared:  0:00:08 ago switch#
```

13.1.16.11 clear ip multicast count

The **`clear ip multicast count`** command clears all counters associated with the multicast traffic.

Command Mode

Gobal Configuration

Command Syntax

`clear ip multicast count` [***group_address*** [***source_address***]]

Parameters

- ***no parameters*** Clears all counts of the multicast route traffic.
- ***group_address*** Clears the multicast traffic count of the specified group address.
  - ***source_address*** Clears the multicast traffic count of the specified group and source addresses.

Guidelines

This command functions only when the ip multicast count command is enabled.

Examples

- This command clears all counters associated with the multicast traffic.

```
switch(config)# clear ip multicast count
```

- This command clears the multicast traffic count of the specified group address.

```
switch(config)# clear ip multicast count 16.39.24.233
```

13.1.16.12 cli vrf

The **`cli vrf`** command specifies the context-active VRF. The context-active VRF determines the default VRF that VRF-context aware commands use when displaying routing table data.

Command Mode

Privileged EXEC

Command Syntax

`cli vrf` [**VRF_ID**]

Parameters

VRF_ID Name of VRF assigned as the current VRF scope. Options include:

- ***vrf_name*** Name of user-defined VRF.
- **default** System-default VRF.

Guidelines

VRF-context aware commands include:

clear arp-cache

show ip

show ip arp

show ip route

show ip route gateway

show ip route host

Related Commands

The show routing-context vrf command displays the context-active VRF.

Example

These commands specify ***magenta*** as the context-active VRF, then display the context-active VRF.

```
switch# cli vrf magenta
switch# show routing-context vrf
Current VRF routing-context is magenta switch#
```

13.1.16.13 compress

The **compress** command increases the hardware resources available for the specified prefix lengths.

The **no compress** command removes the 2-to-1 compression configuraion from the ***running-config***.

Note: The **compress** command is supported only on 7500R, 7280R, 7500R2 and 7280R2 platforms.

Command Mode

Global Configuration

Command Syntax

ip hardware fib optimize prefix-length ***prefix-length***expand ***prefix-length*** compress

no ip hardware fib optimize prefix-length ***prefix-length***expand ***prefix-length*** compress

Parameters

compress Allows configuring up to one compressed prefix length.

Example

In the following example we are configuring prefix length ***20*** and ***24***, expanding prefix length ***19*** and ***23***, and compressing prefix length ***25***.

```
switch(config)# ip hardware fib optimize prefix-length 20 24 expand 19
 23 compress 25
 ! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

13.1.16.14 description (VRF)

The **description** command adds a text string to the configuration mode VRF. The string has no functional impact on the VRF.

The **no description** and **default description** commands remove the text string from the configuration mode VRF by deleting the corresponding **description** command from ***running-config***.

Command Mode

VRF Configuration

Command Syntax

description ***label_text***

no description

default description

Parameters

label_text Character string assigned to the VRF configuration.

Related Commands

The vrf instance command places the switch in VRF configuration mode.

Example

These commands add description text to the ***magenta*** VRF.

```
switch(config)# vrf instance magenta
switch(config-vrf-magenta)# description This is the first vrf
switch(config-vrf-magenta)# show active
 vrf instance magenta
   description This is the first vrf

switch(config-vrf-magenta)#
```

13.1.16.15 dhcp relay

The **`dhcp relay`** command places the switch in the DHCP relay mode. This command is executed under ***global configuration mode***.

The **`no dhcp relay`** command removes DHCP relay configuration from the ***running-config***.

Command Mode

Global Configuration Mode

Command Syntax

`dhcp relay`

`no dhcp relay`

Example

The **`dhcp relay`** command places the switch in the DHCP relay configuration mode.

```
switch(config)# dhcp relay
switch(config-dhcp-relay)#
```

13.1.16.16 interface tunnel

The **`interface tunnel`** command places the switch in interface-tunnel configuration mode.

Interface-tunnel configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed.

The **`no interface tunnel`** command deletes the specified interface tunnel configuration.

The **`exit`** command returns the switch to the global configuration mode.

Command Mode

Global Configuration

Command Syntax

`interface tunnel` ***number***

`no interface tunnel` ***number***

Parameter

number Tunnel interface number. Values range from **0** to **255**.

Example

This command places the switch in interface-tunnel configuration mode for tunnel interface ***10***.

```
switch(config)# interface tunnel 10
switch(config-if-Tu10)#
```

13.1.16.17 ip arp inspection limit

The **ip arp inspection limit** command err-disables the interface if the incoming ARP rate exceeds the configured value rate limit the incoming ARP packets on an interface.

Command Mode

EXEC

Command Syntax

ip arp inspection limit [**RATE** ***pps***] [**BURST_INTERVAL** ***sec*** | none]

no ip arp inspection limit [**RATE** ***pps***] [**BURST_INTERVAL** ***sec*** | none]

default ip arp inspection limit [**RATE** ***pps***] [**BURST_INTERVAL** ***sec*** | none]

Parameters

- **RATE** Specifies the ARP inspection limit rate in packets per second.
  - ***pps*** ARP inspection limit rate packets per second.
- **BURST_INTERVAL** Specifies the ARP inspection limit burst interval.
  - ***sec*** Burst interval second.

Related Commands

- ip arp inspection limit
- ip arp inspection trust
- ip arp inspection vlan
- show ip arp inspection vlan

Examples

- This command configures the rate limit of incoming ARP packets to errdisable the interface when the incoming ARP rate exceeds the configured value, sets the rate to ***512*** (which is the upper limit for the number of invalid ARP packets allowed per second), and sets the burst consecutive interval over which the interface is monitored for a high ARP rate to ***11*** seconds.

```
switch(config)# ip arp inspection limit rate 512 burst interval 11
switch(config)#
```

- This command displays verification of the interface specific configuration.

```
switch(config)# interface ethernet 3/1
switch(config)# ip arp inspection limit rate 20 burst interval 5
switch(config)# interface Ethernet 3/3
switch(config)# ip arp inspection trust
switch(config)# show ip arp inspection interfaces

 Interface      Trust State  Rate (pps) Burst Interval
```

```
-------------  -----------  ----------  --------------
Et3/1          Untrusted    20          5
Et3/3          Trusted      None        N/A

switch(config)#
```

13.1.16.18 ip arp inspection logging

The **ip arp inspection logging** command enables logging of incoming ARP packets on the interface if the rate exceeds the configured value.

Command Mode

EXEC

Command Syntax

ip arp inspection logging [**RATE** ***pps***][**BURST_INTERVAL** ***sec*** | none]

no ip arp inspection logging [**RATE** ***pps***][**BURST_INTERVAL** ***sec*** | none]

default ip arp inspection logging [**RATE** ***pps***][**BURST_INTERVAL** ***sec*** | none]

Parameters

- **RATE** Specifies the ARP inspection limit rate in packets per second.
  - ***<pps>*** ARP inspection limit rate packets per second.
- **BURST_INTERVAL** Specifies the ARP inspection limit burst interval.
  - ***sec*** Burst interval second.

Related Commands

- ip arp inspection limit
- ip arp inspection trust
- ip arp inspection vlan
- show ip arp inspection vlan

Example

This command enables logging of incoming ARP packets when the incoming ARP rate exceeds the configured value on the interface, sets the rate to ***2048*** (which is the upper limit for the number of invalid ARP packets allowed per second), and sets the burst consecutive interval over which the interface is monitored for a high ARP rate to ***15*** seconds.

```
switch(config)# ip arp inspection logging rate 2048 burst interval 15
switch(config)#
```

13.1.16.19 ip arp inspection trust

The **ip arp inspection trust** command configures the trust state of an interface. By default, all interfaces are untrusted.

Command Mode

EXEC

Command Syntax

ip arp inspection trust

no ip arp inspection trust

default ip arp inspection trust

Related Commands

- ip arp inspection limit
- ip arp inspection logging
- show ip arp inspection vlan
- ip arp inspection vlan

Examples

- This command configures the trust state of an interface.

```
switch(config)# ip arp inspection trust
switch(config)#
```

- This command configures the trust state of an interface to untrusted.

```
switch(config)# no ip arp inspection trust
switch(config)#
```

- This command configures the trust state of an interface to its default (untrusted).

```
switch(config)# default ip arp inspection trust
switch(config)#
```

13.1.16.20 ip arp inspection vlan

The **ip arp inspection vlan** command enables ARP inspection. ARP requests and responses on untrusted interfaces are intercepted on specified VLANs, and intercepted packets are verified to have valid IP-MAC address bindings. All invalid ARP packets are dropped. On trusted interfaces, all incoming ARP packets are processed and forwarded without verification. By default, ARP inspection is disabled on all VLANs.

Command Mode

EXEC

Command Syntax

ip arp inspection vlan [**LIST**]

Parameters

LIST Specifies the VLAN interface number.

Related Commands

- ip arp inspection limit
- ip arp inspection trust
- ip arp inspection vlan

Example

- This command enables ARP inspection on VLANs ***1*** through ***150***.

```
switch(config)# ip arp inspection vlan 1 - 150
switch(config)#
```

- This command disables ARP inspection on VLANs ***1*** through ***150***.

```
switch(config)# no ip arp inspection vlan 1 - 150
switch(config)#
```

- This command sets the ARP inspection default to VLANs ***1*** through ***150***.

```
switch(config)# default ip arp inspection vlan 1 - 150
switch(config)#
```

- These commands enable ARP inspection on multiple VLANs 1 through ***150*** and ***200*** through ***250***.

```
switch(config)# ip arp inspection vlan 1-150,200-250
switch(config)#
```

13.1.16.21 ip dhcp relay all-subnets

The **`ip dhcp relay all-subnets`** command configures the DHCP smart relay status on the configuration mode interface. DHCP smart relay supports forwarding DHCP requests with a client's secondary IP addresses in the gateway address field. Enabling DHCP smart relay on an interface requires that DHCP relay is also enabled on that interface.

By default, an interface assumes the global DHCP smart relay setting as configured by the ip dhcp relay all-subnets default command. The **`ip dhcp relay all-subnets`** command, when configured, takes precedence over the global smart relay setting.

The **`no ip dhcp relay all-subnets`** command disables DHCP smart relay on the configuration mode interface. The **`default ip dhcp relay all-subnets`** command restores the interface's to the default DHCP smart relay setting, as configured by the **`ip dhcp relay all-subnets default`** command, by removing the corresponding **`ip dhcp relay all-subnets`** or **`no ip dhcp relay all-subnets`** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip dhcp relay all-subnets`

`no ip dhcp relay all-subnets`

`default ip dhcp relay all-subnets`

Examples

- This command enables DHCP smart relay on VLAN interface ***100***.

```
switch(config)# interface vlan 100
switch(config-if-Vl100)# ip helper-address 10.4.4.4
switch(config-if-Vl100)# ip dhcp relay all-subnets
switch(config-if-Vl100)# show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCP Smart Relay is enabled
Interface: Vlan100
  DHCP Smart Relay is enabled
```

```
  DHCP servers: 10.4.4.4
switch(config-if-Vl100)#
```

- This command disables DHCP smart relay on VLAN interface ***100***.

```
switch(config-if-Vl100)# no ip dhcp relay all-subnets
switch(config-if-Vl100)# show active
 interface Vlan100
   no ip dhcp relay all-subnets
   ip helper-address 10.4.4.4
switch(config-if-Vl100)# show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCP Smart Relay is enabled
Interface: Vlan100
  DHCP Smart Relay is disabled
  DHCP servers: 10.4.4.4
switch(config-if-Vl100)#
```

- This command enables DHCP smart relay globally, configures VLAN interface ***100*** to use the global setting, then displays the DHCP relay status.

```
switch(config)# ip dhcp relay all-subnets default
switch(config)# interface vlan 100
switch(config-if-Vl100)# ip helper-address 10.4.4.4
switch(config-if-Vl100)# default ip dhcp relay
switch(config-if-Vl100)# show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCP Smart Relay is enabled
Interface: Vlan100
  Option 82 Circuit ID: 333
  DHCP Smart Relay is enabled
  DHCP servers: 10.4.4.4
switch(config-if-Vl100)#
```

13.1.16.22 ip dhcp relay all-subnets default

The **`ip dhcp relay all-subnets default`** command configures the global DHCP smart relay setting. DHCP smart relay supports forwarding DHCP requests with a client's secondary IP addresses in the gateway address field. The default global DHCP smart relay setting is disabled.

The global DHCP smart relay setting is applied to all interfaces for which an ip dhcp relay all-subnets statement is not configured. Enabling DHCP smart relay on an interface requires that DHCP relay is also enabled on that interface.

The **`no ip dhcp relay all-subnets default`** and **`default ip dhcp relay all-subnets default`** commands restore the global DHCP smart relay default setting of disabled by removing the **`ip dhcp relay all-subnets default`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip dhcp relay all-subnets default`

`no ip dhcp relay all-subnets default`

`default ip dhcp relay all-subnets default`

Related Commands

- ip helper-address Enables the DHCP relay agent on a configuration mode interface.
- ip dhcp relay all-subnets Enables the DHCP smart relay agent on a configuration mode interface.

Example

This command configures the global DHCP smart relay setting to **enabled**.

```
switch(config)# ip dhcp relay all-subnets default
switch(config)#
```

13.1.16.23 ip dhcp relay always-on

The **ip dhcp relay always-on** command enables the switch DHCP relay agent on the switch regardless of the DHCP relay agent status on any interface. By default, the DHCP relay agent is enabled only if at least one routable interface is configured with an ip helper-address statement.

The **no ip dhcp relay always-on** and **default ip dhcp relay always-on** commands remove the **ip dhcp relay always-on** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip dhcp relay always-on

no ip dhcp relay always-on

default ip dhcp relay always-on

Related Commands

These commands implement DHCP relay agent.

- ip helper-address
- ip dhcp relay information option (Global)
- ip dhcp relay information option circuit-id

Example

This command enables the DHCP relay agent.

```
switch(config)# ip dhcp relay always-on
switch(config)#
```

13.1.16.24 ip dhcp relay information option (Global)

The **ip dhcp relay information option** command configures the switch to attach tags to DHCP requests before forwarding them to the DHCP servers designated by ip helper-address commands. The command specifies the tag contents for packets forwarded by the interface that it configures.

The **no ip dhcp relay information option** and **default ip dhcp relay information option** commands restore the switch's default setting of not attaching tags to DHCP requests by removing the **ip dhcp relay information option** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip dhcp relay information option`

`no ip dhcp relay information option`

`default ip dhcp relay information option`

Related Commands

These commands implement DHCP relay agent.

- ip helper-address
- ip dhcp relay always-on
- ip dhcp relay information option circuit-id

Example

This command enables the attachment of tags to DHCP requests that are forwarded to DHCP server addresses.

```
switch(config)# ip dhcp relay information option
switch(config)#
```

13.1.16.25 ip dhcp relay information option circuit-id

The **`ip dhcp relay information option circuit-id`** command specifies the content of tags that the switch attaches to DHCP requests before they are forwarded from the configuration mode interface to DHCP server addresses specified by ip helper-address commands. Tags are attached to outbound DHCP requests only if the information option is enabled on the switch (ip dhcp relay information option circuit-id). The default value for each interface is the name and number of the interface.

The **`no ip dhcp relay information option circuit-id`** and **`default ip dhcp relay information option circuit-id`** commands restore the default content setting for the configuration mode interface by removing the corresponding command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip dhcp relay information option circuit-id` ***id_label***

`no ip dhcp relay information option circuit-id`

`default ip dhcp relay information option circuit-id`

Parameters

id_label Tag content. Format is alphanumeric characters (maximum 15 characters).

Related Commands

- ip helper-address
- ip dhcp relay always-on

- ip dhcp relay information option (Global)

Example

This command configures **x-1234** as the tag content for packets send from VLAN ***200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ip dhcp relay information option circuit-id
 x-1234
switch(config-if-Vl200)#
```

13.1.16.26 ip dhcp snooping

The **ip dhcp snooping** command enables DHCP snooping globally on the switch. DHCP snooping is a set of Layer 2 processes that can be configured on LAN switches and used with DHCP servers to control network access to clients with specific IP/MAC addresses. The switch supports Option-82 insertion, which is a DHCP snooping process that allows relay agents to provide remote-ID and circuit-ID information to DHCP reply and request packets. DHCP servers use this information to determine the originating port of DHCP requests and associate a corresponding IP address to that port. DHCP servers use port information to track host location and IP address usage by authorized physical ports.

DHCP snooping uses the information option (Option-82) to include the switch MAC address (router-ID) along with the physical interface name and VLAN number (circuit-ID) in DHCP packets. After adding the information to the packet, the DHCP relay agent forwards the packet to the DHCP server as specified by the DHCP protocol.

DHCP snooping on a specified VLAN requires all of these conditions to be met:

- DHCP snooping is globally enabled.
- Insertion of option-82 information in DHCP packets is enabled.
- DHCP snooping is enabled on the specified VLAN.
- DHCP relay is enabled on the corresponding VLAN interface.

The **no ip dhcp snooping** and **default ip dhcp snooping** commands disables global DHCP snooping by removing the **ip dhcp snooping** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip dhcp snooping

no ip dhcp snooping

default ip dhcp snooping

Related Commands

- ip dhcp snooping information option enables insertion of option-82 snooping data.
- ip helper-address enables the DHCP relay agent on a configuration mode interface.

Example

This command globally enables snooping on the switch, displaying DHCP snooping status prior and after invoking the command.

```
switch(config)# show ip dhcp snooping
DHCP Snooping is disabled
```

```
switch(config)# ip dhcp snooping
switch(config)# show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is not operational
DHCP Snooping is configured on following VLANs:
  None
DHCP Snooping is operational on following VLANs:
  None
Insertion of Option-82 is disabled
switch(config)#
```

13.1.16.27 ip dhcp snooping bridging

The **ip dhcp snooping bridging** command enables the DHCP snooping bridging configuration.

The **no ip dhcp snooping bridging** command removes the DHCP snooping bridging configuration from the ***running-config***.

Command Mode

Global Configuration Mode

Command Syntax

ip dhcp snooping bridging

no ip dhcp snooping bridging

Example

This command configures the DHCP snooping bridging.

```
switch# configure
switch(config)# ip dhcp snooping bridging
```

13.1.16.28 ip dhcp snooping information option

The **ip dhcp snooping information option** command enables the insertion of option-82 DHCP snooping information in DHCP packets on VLANs where DHCP snooping is enabled. DHCP snooping is a layer 2 switch process that allows relay agents to provide remote-ID and circuit-ID information to DHCP reply and request packets. DHCP servers use this information to determine the originating port of DHCP requests and associate a corresponding IP address to that port.

DHCP snooping uses information option (Option-82) to include the switch MAC address (router-ID) along with the physical interface name and VLAN number (circuit-ID) in DHCP packets. After adding the information to the packet, the DHCP relay agent forwards the packet to the DHCP server through DHCP protocol processes.

DHCP snooping on a specified VLAN requires all of these conditions to be met:

- DHCP snooping is globally enabled.
- Insertion of option-82 information in DHCP packets is enabled.
- DHCP snooping is enabled on the specified VLAN.
- DHCP relay is enabled on the corresponding VLAN interface.

When global DHCP snooping is not enabled, the **ip dhcp snooping information option** command persists in ***running-config*** without any operational effect.

The **no ip dhcp snooping information option** and **default ip dhcp snooping information option** commands disable the insertion of option-82 DHCP snooping information in DHCP packets by removing the **ip dhcp snooping information option** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip dhcp snooping information option`

`no ip dhcp snooping information option`

`default ip dhcp snooping information option`

Related Commands

- ip dhcp snooping globally enables DHCP snooping.
- ip helper-address enables the DHCP relay agent on a configuration mode interface.

Example

These commands enable DHCP snooping on DHCP packets from ports on snooping-enabled VLANs. DHCP snooping was previously enabled on the switch.

```
switch(config)# ip dhcp snooping information option
switch(config)# show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is operational
DHCP Snooping is configured on following VLANs:
  100
DHCP Snooping is operational on following VLANs:
  100
Insertion of Option-82 is enabled
  Circuit-id format: Interface name:Vlan ID
  Remote-id: 00:1c:73:1f:b4:38 (Switch MAC)
switch(config)#
```

13.1.16.29 ip dhcp snooping vlan

The **`ip dhcp snooping vlan`** command enables DHCP snooping on specified VLANs. DHCP snooping is a Layer 2 process that allows relay agents to provide remote-ID and circuit-ID information in DHCP packets. DHCP servers use this data to determine the originating port of DHCP requests and associate a corresponding IP address to that port. DHCP snooping is configured on a global and VLAN basis.

VLAN snooping on a specified VLAN requires each of these conditions:

- DHCP snooping is globally enabled.
- Insertion of option-82 information in DHCP packets is enabled.
- DHCP snooping is enabled on the specified VLAN.
- DHCP relay is enabled on the corresponding VLAN interface.

When global DHCP snooping is not enabled, the **`ip dhcp snooping vlan`** command persists in ***running-config*** without any operational affect.

The **`no ip dhcp snooping information option`** and **`default ip dhcp snooping information option`** commands disable DHCP snooping operability by removing the **`ip dhcp snooping information option`** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip dhcp snooping vlan` ***v_range***

`no ip dhcp snooping vlan` ***v_range***

default ip dhcp snooping vlan ***v_range***

Parameters

- ***v_range*** VLANs upon which snooping is enabled. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from **1** to **4094**.
- The ip dhcp snooping command globally enables DHCP snooping.
- The ip dhcp snooping vlan command enables insertion of option-82 snooping data.
- The ip helper-address command enables the DHCP relay agent on a configuration mode interface.

Example

These commands enable DHCP snooping globally, DHCP on VLAN interface ***100***, and DHCP snooping on ***vlan100***.

```
switch(config)# ip dhcp snooping
switch(config)# ip dhcp snooping information option
switch(config)# ip dhcp snooping vlan 100
switch(config)# interface vlan 100
switch(config-if-Vl100)# ip helper-address 10.4.4.4
switch(config-if-Vl100)# show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is operational
DHCP Snooping is configured on following VLANs:
  100
DHCP Snooping is operational on following VLANs:
  100
Insertion of Option-82 is enabled
  Circuit-id format: Interface name:Vlan ID
  Remote-id: 00:1c:73:1f:b4:38 (Switch MAC)
switch(config)#
```

13.1.16.30 ip hardware fib ecmp resilience

The **ip hardware fib ecmp resilience** command enables resilient ECMP for the specified IP address prefix and configures a fixed number of next hop entries in the hardware ECMP table for that prefix. In addition to specifying the maximum number of next hop addresses that the table can contain for the prefix, the command includes a redundancy factor that allows duplication of each next hop address. The fixed table space for the address is the maximum number of next hops multiplied by the redundancy factor.

Resilient ECMP is useful when it is not desirable for routes to be rehashed due to link flap, as when ECMP is being used for load balancing.

The **no ip hardware fib ecmp resilience** and **default ip hardware fib ecmp resilience** commands restore the default hardware ECMP table management by removing the **ip hardware fib ecmp resilience** command from ***running-config*** .

Command Mode

Global Configuration

Command Syntax

ip hardware fib ecmp resilience ***net_addr*** capacity ***nhop_max*** redundancy **duplicates**

no ip hardware fib ecmp resilience ***net_addr***

default ip hardware fib ecmp resilience ***net_addr***

Parameters

- ***net_addr*** IP address prefix managed by command. (CIDR or address-mask).

- ***nhop_max*** Maximum number of nexthop addresses for specified IP address prefix. Value range varies by platform:
  - Helix: <**2** to **64**>
  - Trident: <**2** to **32**>
  - Trident II: <**2** to **64**>
- **duplicates** Specifies the redundancy factor. Value ranges from ***1*** to ***128***.

Example

This command configures a hardware ECMP table space of 24 entries for the IP address ***10.14.2.2/24***. A maximum of six next-hop addresses can be specified for the IP address. When the table contains six next-hop addresses, each appears in the table four times. When the table contains fewer than six next-hop addresses, each is duplicated until the 24 table entries are filled.

```
switch(config)# ip hardware fib ecmp resilience 10.14.2.2/24 capacity
 6 redundancy 4
switch(config)#
```

13.1.16.31 ip hardware fib next-hop resource optimization

The **ip hardware fib next-hop resource optimization** command is used to enable or disable the resource optimization features on the switch. By default, RECMP is enabled on the switch.

The **no hardware fib next-hop resource optimization** command removes all the resource optimization features running on the switch.

Command Mode

Global Configuration Mode

Command Syntax

ip hardware fib next-hop resource optimization ***OPTIONS***

no ip hardware fib next-hop resource optimization ***OPTIONS***

Parameters

- The following two options are allowed to configure with this command:
  - ***disabled*** Disable hardware resource optimization for adjacency programming.
  - ***thresholds*** Utilization percentage for starting or stopping optimization. The resource utilization percentage value ranges from 0 to 100. It can be set to low and high.

Examples

- The following command is used to disable all hardware resource optimization features on the switch:

```
switch# configure terminal
switch(config)# ip hardware fib next-hop resource optimization disabled
```

- The following command is used to configure the thresholds for starting and stopping the optimization:

```
switch(config)# ip hardware fib next-hop resource optimization thresholds
 low 20 high 80
```

13.1.16.32 ip hardware fib optimize

The **ip hardware fib optimize** command enables IPv4 route scale. The platform layer 3 agent is restarted to ensure IPv4 routes are optimized with the agent SandL3Unicast terminate command for the configuration mode interface.

Command Mode

Global Configuration

Command Syntax

ip hardware fib optimize exact-match prefix-length ***prefix-length prefix-length***

ip hardware fib optimize exact-match prefix-length ***prefix-length prefix-length***

Parameters

prefix-length The length of the prefix equal to ***12***, ***16***, ***20***, ***24***, ***28***, or ***32***. One additional prefix-length limited to the prefix-length of ***32*** is optional.

Related Commands

- The agent SandL3Unicast terminate command enables restarting the layer 3 agent to ensure IPv4 routes are optimized.
- The show platform arad ip route command shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.
- The show platform arad ip route summary command shows hardware resource usage of IPv4 routes.

Examples

- This configuration command allows configuring prefix lengths ***12*** and ***32***.

```
switch(config)# ip hardware fib optimize exact-match prefix-length 12 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

- One of the two prefixes in this command is a prefix-length of ***32***, which is required in the instance where there are two prefixes. For this command to take effect, the platform Layer 3 agent must be restarted.
  - This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)# agent SandL3Unicast terminate
SandL3Unicast was terminated
```

- Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.
  - This configuration command allows configuring prefix lengths ***32*** and ***16***.

```
switch(config)# ip hardware fib optimize exact-match prefix-length 32 16
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

- One of the two prefixes in this command is a prefix-length of ***32***, which is required in the instance where there are two prefixes. For this command to take effect, the platform Layer 3 agent must be restarted.
  - This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)# agent SandL3Unicast terminate
SandL3Unicast was terminated
```

- Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

- This configuration command allows configuring prefix length ***24***.

```
switch(config)# ip hardware fib optimize exact-match prefix-length 24
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

- In this instance, there is only one prefix-length, so a prefix-length of ***32*** is not required. For this command to take effect, the platform Layer 3 agent must be restarted.
  - This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)# agent SandL3Unicast terminate
SandL3Unicast was terminated
```

- Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.
  - This configuration command allows configuring prefix length ***32***.

```
switch(config)# ip hardware fib optimize exact-match prefix-length 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are
 optimized
```

- For this command to take effect, the platform Layer 3 agent must be restarted.
  - This configuration command restarts the platform Layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)# agent SandL3Unicast terminate
SandL3Unicast was terminated
```

- Restarting the platform Layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.
  - This configuration command disables configuring prefix lengths ***12*** and ***32***.

```
switch(config)# no ip hardware fib optimize exact-match prefix-length 12
 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are not
 optimized
```

  - One of the two prefixes in this command is a prefix-length of ***32***, which is required in the instance where there are two prefixes. For this command to take effect, the platform Layer 3 agent must be restarted.

13.1.16.33 ip helper-address

The **`ip helper-address`** command enables the DHCP relay agent on the configuration mode interface and specifies a forwarding address for DHCP requests. An interface that is configured with multiple helper-addresses forwards DHCP requests to all specified addresses.

The **`no ip helper-address`** and **`default ip helper-address`** commands remove the corresponding **`ip helper-address`** command from ***running-config***. Commands that do not specify an IP helper-address remove all helper-addresses from the interface.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ip helper-address ***ipv4_addr*** [vrf ***vrf_name***][source-address ***ipv4_addr*** | source-interface **INTERFACES**]

no ip helper-address [***ipv4_addr***]

default ip helper-address [***ipv4_addr***]

Parameters

- **vrf** ***vrf_name*** Specifies the user-defined VRF for DHCP server.
- ***ipv4_addr*** Specifies the DHCP server address accessed by interface.
- **source-address** ***ipv4_addr*** Specifies the source IPv4 address to communicate with DHCP server.
- **source-interface INTERFACES** Specifies the source interface to communicate with DHCP server. Options include:
  - **Ethernet** ***eth_num*** Specifies the Ethernet interface number.
  - **Loopback** ***lpbck_num*** Specifies the loopback interface number. Value ranges from **0** to **1000**.
  - **Management** ***mgmt_num*** Specifies the management interface number. Accepted values are ***1*** and ***2***.
  - **Port-Channel** {***int_num*** | ***sub_int_num***} Specifies the port-channel interface or subinterface number. Value of interface ranges from **1** to **2000**. Value of sub-interface ranges from **1** to **4094**.
  - **Tunnel** ***tnl_num*** Specifies the tunnel interface number. Value ranges from **0** to **255**.
  - **VLAN** ***vlan_num*** Specifies the Ethernet interface number. Value ranges from **1** to **4094**.

Related Commands

- ip dhcp relay always-on
- ip dhcp relay information option (Global)
- ip dhcp relay information option circuit-id

Guidelines

If the source-address parameter is specified, then the DHCP client receives an IPv4 address from the subnet of source IP address. The source-address must be one of the configured addresses on the interface.

Examples

- This command enables DHCP relay on the VLAN interface ***200***; and configure the switch to forward DHCP requests received on this interface to the server at ***10.10.41.15***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ip helper-address 10.10.41.15
switch(config-if-Vl200)# show active
interface Vlan200
   ip helper-address 10.10.41.15
switch(config-if-Vl200)#
```

- This command enables DHCP relay on the ***interface ewthernet 1/2***; and configures the switch to use ***2.2.2.2*** as the source IP address when relaying IPv4 DHCP messages to the server at ***1.1.1.1***.

```
switch(config)# interface ethernet 1/2
switch(config-if-Et1/2)# ip helper-address 1.1.1.1 source-address
 2.2.2.2
switch(config-if-Et1/2)#
```

13.1.16.34 ip icmp redirect

The **ip icmp redirect** command enables the transmission of ICMP redirect messages. Routers send ICMP redirect messages to notify data link hosts of the availability of a better route for a specific destination.

The **no ip icmp redirect** disables the switch from sending ICMP redirect messages.

Command Mode

Global Configuration

Command Syntax

ip icmp redirect

no ip icmp redirect

default ip icmp redirect

Example

This command disables the redirect messages.

```
switch(config)# no ip icmp redirect
switch(config)# show running-config
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
no ip icmp redirect
ip routing
!
               <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

13.1.16.35 ip load-sharing

The **ip load-sharing** command provides the hash seed to an algorithm that the switch uses to distribute data streams among multiple equal-cost routes to an individual IPv4 subnet.

In a network topology using Equal-Cost Multipath routing, all switches performing identical hash calculations may result in hash polarization, leading to uneven load distribution among the data paths. Hash polarization is avoided when switches use different hash seeds to perform different hash calculations.

The **no ip load-sharing** and **default ip load-sharing** commands return the hash seed to the default value of zero by removing the **ip load-sharing** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip load-sharing **HARDWARE** ***seed***

no ip load-sharing **HARDWARE**

default ip load-sharing **HARDWARE**

Parameters

- **HARDWARE** The ASIC switching device. The available option depend on the switch platform. Verify available options with the CLI ? command.
  - **arad**
  - **fm6000**

  - **petraA**
  - **trident**
- ***seed*** The hash seed. Value range varies by switch platform. The default value on all platforms is **0**.
  - when **HARDWARE**=**arad** ***seed*** ranges from **0** to **2**.
  - when **HARDWARE**=**fm6000** ***seed*** ranges from **0** to **39**.
  - when **HARDWARE**=**petraA** ***seed*** ranges from **0** to **2**.
  - when **HARDWARE**=**trident** ***seed*** ranges from **0** to **5**.

Example

This command sets the IPv4 load sharing hash seed to one on FM6000 platform switches.

```
switch(config)# ip load-sharing fm6000 1
switch(config)#
```

13.1.16.36 ip local-proxy-arp

The **ip local-proxy-arp** command enables local proxy ARP (Address Resolution Protocol) on the configuration mode interface. When local proxy ARP is enabled, ARP requests received on the configuration mode interface will return an IP address even when the request comes from within the same subnet.

The **no ip local-proxy-arp** and **default ip local-proxy-arp** commands disable local proxy ARP on the configuration mode interface by removing the corresponding **ip local-proxy-arp** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ip local-proxy-arp

no ip local-proxy-arp

default ip local-proxy-arp

Example

These commands enable local proxy ARP on VLAN interface ***140***.

```
switch(config)# interface vlan 140
switch(config-if-Vl140)# ip local-proxy-arp
switch(config-if-Vl140)# show active
interface Vlan140
   ip local-proxy-arp
switch(config-if-Vl140)#
```

13.1.16.37 ip multicast count

The **ip multicast count** command enables the IPv4 multicast route traffic counter of group and source addresses in either bytes or packets.

The **no ip multicast count** command deletes all multicast counters including the routes of group and source addresses.

The **no ip multicast count *group_address source_address*** command removes the current configuration of the specified group and source addresses. It does not delete the counter because the wildcard is still active.

The **default ip multicast count** command reverts the current counter configuration of multicast route to the default state.

Command Mode

Global Configuration

Command Syntax

ip multicast count [***group_address*** [***source_address***] | bytes | packets]

no ip multicast count [***group_address*** [***source_address***] | bytes | packets]

default ip multicast count [***group_address*** [***source_address***] | bytes | packets]

Parameters

- ***group_address*** Configures the multicast route traffic count of the specified group address.
  - ***source_address*** Configures the multicast route traffic count of the specified group and source addresses.
- **bytes** Configures the multicast route traffic count to bytes.
- **packets** Configures the multicast route traffic count to packets.

Guidelines

This command is supported on the FM6000 platform only.

Examples

- This command configures the multicast route traffic count to bytes.

```
switch(config)# ip multicast count bytes
```

- This command configures the multicast route traffic count of the specified group and source addresses.

```
switch(config)# ip multicast count 10.50.30.23 45.67.89.100
```

- This command deletes all multicast counters including the routes of group and source addresses.

```
switch(config)# no ip multicast count
```

- This command reverts the current multicast route configuration to the default state.

```
switch(config)# default ip multicast count
```

13.1.16.38 ip proxy-arp

The **ip proxy-arp** command enables proxy ARP on the configuration mode interface. Proxy ARP is disabled by default. When proxy ARP is enabled, the switch responds to all ARP requests, including gratuitous ARP requests, with target IP addresses that match a route in the routing table.

The **no ip proxy-arp** and **default ip proxy-arp** commands disable proxy ARP on the configuration mode interface by removing the corresponding **ip proxy-arp** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ip proxy-arp

no ip proxy-arp

default ip proxy-arp

Example

This command enables proxy ARP on ***interface ethernet 4***.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#ip proxy-arp
switch(config-if-Et4)#
```

13.1.16.39 ip route

The **ip route** command creates a static route. The destination is a network segment; the nexthop address is either an IPv4 address or a routable port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

By default, the administrative distance assigned to static routes is 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with an administrative distance value of 200 is overridden by OSPF intra-area routes, which have a default administrative distance of 110.

Tags are used by route maps to filter routes. The default tag value on static routes is 0.

Multiple routes with the same destination and the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic equally through all ECMP route paths. All paths comprising an ECMP are assigned identical tag values; commands that change the tag value of a path change the tag value of all paths in the ECMP.

The **no ip route** and **default ip route** commands delete the specified static route by removing the corresponding **ip route** command from ***running-config***. Commands that do not list a nexthop address remove all **ip route** statements with the specified destination from ***running-config***. If an **ip route** statement exists for the same IP address in multiple VRFs, each must be removed separately. All static routes in a user-defined VRF are deleted when the VRF is deleted.

Command Mode

Global Configuration

Command Syntax

`ip route` [**VRF_INSTANCE**] ***dest_net*** **NEXTHOP** [**DISTANCE**][**TAG_OPTION**][**RT_NAME**]

`no ip route` [**VRF_INSTANCE**] ***dest_net*** [**NEXTHOP**][**DISTANCE**]

`default ip route` [**VRF_INSTANCE**] ***dest_net*** [**NEXTHOP**][**DISTANCE**]

Parameters

- **VRF_INSTANCE** Specifies the VRF instance being modified.
  - ***no parameter*** Changes are made to the default VRF.
  - **vrf *vrf_name*** Changes are made to the specified VRF.
- ***dest_net*** Destination IPv4 subnet (CIDR or address-mask notation).
- **NEXTHOP** Location or access method of next hop device. Options include:
  - ***ipv4_addr*** An IPv4 address.
  - **null0** Null0 interface.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- **DISTANCE** Administrative distance assigned to route. Options include:
  - ***no parameter*** Route assigned default administrative distance of one.
  - **`1-255`** The administrative distance assigned to route.
- **TAG_OPTION** Static route tag. Options include:
  - ***no parameter*** Assigns default static route tag of ***0***.
  - **tag *t_value*** Static route tag value. ***t_value*** ranges from **`0`** to **`4294967295`**.
- **RT_NAME** Associates descriptive text to the route. Options include:
  - ***no parameter*** No text is associated with the route.
  - **name *descriptive_text*** The specified text is assigned to the route.

Related Command

The ip route nexthop-group command creates a static route that specifies a Nexthop Group to determine the Nexthop address.

Example

This command creates a static route in the default VRF.

```
switch(config)# ip route 172.17.252.0/24 vlan 2000
switch(config)#
```

13.1.16.40 ip routing

The **`ip routing`** command enables IPv4 routing. When IPv4 routing is enabled, the switch attempts to deliver inbound packets to destination IPv4 addresses by forwarding them to interfaces or next hop addresses specified by the forwarding table.

The **no ip routing** and **default ip routing** commands disable IPv4 routing by removing the **ip routing** command from ***running-config***. When IPv4 routing is disabled, the switch attempts to deliver inbound packets to their destination MAC addresses. When this address matches the switch's MAC address, the packet is delivered to the CPU. IP packets with IPv4 destinations that differ from the switch's address are typically discarded. The **delete-static-routes** option removes static entries from the routing table.

IPv4 routing is disabled by default.

Command Mode

Global Configuration

Command Syntax

ip routing [**VRF_INSTANCE**]

no ip routing [**DELETE_ROUTES**][**VRF_INSTANCE**

default ip routing [**DELETE_ROUTES**][**VRF_INSTANCE**]

Parameters

- **DELETE_ROUTES** Resolves routing table static entries when routing is disabled.
  - ***no parameter*** Routing table retains static entries.
  - **delete-static-routes** Static entries are removed from the routing table.
- **VRF_INSTANCE** Specifies the VRF instance being modified.
  - ***no parameter*** Changes are made to the default VRF.
  - **vrf *vrf_name*** Changes are made to the specified user-defined VRF.

Example

This command enables IPv4 routing.

```
switch(config)# ip routing
switch(config)#
```

13.1.16.41 ip source binding

IP source guard (IPSG) is supported on Layer 2 Port-Channels, not member ports. The IPSG configuration on port channels supersedes the configuration on the physical member ports. Hence, source IP MAC binding entries should be configured on port channels. When configured on a port channel member port, IPSG does not take effect until this port is deleted from the port channel configuration.

Note: IP source bindings are also used by static ARP inspection.

The **no ip source binding** and **default ip source binding** commands exclude parameters from IPSG filtering, and set the default for **ip source binding**.

Command Mode

Interface-Ethernet Configuration

Command Syntax

ip source binding [***IP_ADDRESS***][***MAC_ADDRESS***] vlan [***VLAN_RANGE***] interface [***INTERFACE***]

no ip source binding [***IP_ADDRESS***][***MAC_ADDRESS***] vlan [***VLAN_RANGE***] interface [***INTERFACE***]

default ip source binding [***IP_ADDRESS***][***MAC_ADDRESS***] vlan [***VLAN_RANGE***] interface [***INTERFACE***]

Parameters

- **IP_ADDRESS** Specifies the IP ADDRESS.
- **MAC_ADDRESS** Specifies the MAC ADDRESS.
- **VLAN_RANGE** Specifies the VLAN ID range.
- **INTERFACE** Specifies the Ethernet interface.

Related Commands

- ip verify source
- show ip verify source

Example

This command configures source IP-MAC binding entries to IP address ***10.1.1.1***, MAC address ***0000.aaaa.1111***, VLAN ID ***4094***, and ***interface ethernet 36***.

```
switch(config)# ip source binding 10.1.1.1 0000.aaaa.1111 vlan 4094
 interface
ethernet 36
switch(config)#
```

13.1.16.42 ip verify source

The **`ip verify source`** command configures IP source guard (IPSG) applicable only to Layer 2 ports. When configured on Layer 3 ports, IPSG does not take effect until this interface is converted to Layer 2.

IPSG is supported on Layer 2 Port-Channels, not member ports. The IPSG configuration on port channels supersedes the configuration on the physical member ports. Therefore, source IP MAC binding entries should be configured on port channels. When configured on a port channel member port, IPSG does not take effect until this port is deleted from the port channel configuration.

The **`no ip verify source`** and **`default ip verify source`** commands exclude VLAN IDs from IPSG filtering, and set the default for **`ip verify source`**.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`ip verify source vlan` [**VLAN_RANGE**]

`no ip verify source` [**VLAN_RANGE**]

`default ip verify source`

Parameters

VLAN_RANGE Specifies the VLAN ID range.

Related Commands

- ip source binding
- show ip verify source

Example

This command excludes VLAN IDs ***1*** through ***3*** from IPSG filtering. When enabled on a trunk port, IPSG filters the inbound IP packets on all allowed VLANs. IP packets received on VLANs ***4***

through ***10*** on ***Ethernet 36*** will be filtered by IPSG, while those received on VLANs ***1*** through ***3*** are permitted.

```
switch(config)#no ip verify source vlan 1-3
switch(config)#interface ethernet 36
switch(config-if-Et36)#switchport mode trunk
switch(config-if-Et36)#switchport trunk allowed vlan 1-10
switch(config-if-Et36)#ip verify source
switch(config-if-Et36)#
```

13.1.16.43 ip verify

The **`ip verify`** command configures Unicast Reverse Path Forwarding (uRPF) for inbound IPv4 packets on the configuration mode interface. uRPF verifies the accessibility of source IP addresses in packets that the switch forwards.

uRPF defines two operational modes: strict mode and loose mode.

- **Strict mode:** uRPF verifies that a packet is received on the interface that its routing table entry specifies for its return packet.
- **Loose mode:** uRPF validation does not consider the inbound packet's ingress interface only that there is a valid return path.

The **`no ip verify`** and **`default ip verify`** commands disable uRPF on the configuration mode interface by deleting the corresponding **`ip verify`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip verify unicast source reachable-via` **RPF_MODE**

`no ip verify unicast`

`default ip verify unicast`

Parameters

RPF_MODE Specifies the uRPF mode. Options include:

- **any** Loose mode.
- **rx** Strict mode.
- **rx allow-default** Strict mode. All inbound packets are forwarded if a default route is defined.

Guidelines

The first IPv4 uRPF implementation briefly disrupts IPv4 unicast routing. Subsequent **`ip verify`** commands on any interface do not disrupt IPv4 routing.

Examples

- This command enables uRPF loose mode on ***VLAN interface 17***.

```
switch(config)#interface vlan 17
```

```
switch(config-if-Vl17)#ip verify unicast source reachable-via any
switch(config-if-Vl17)#show active
 interface Vlan17
   ip verify unicast source reachable-via any
switch(config-if-Vl17)#
```

- This command enables uRPF strict mode on ***VLAN interface 18***.

```
switch(config)#interface vlan 18
switch(config-if-Vl18)#ip verify unicast source reachable-via rx
switch(config-if-Vl18)#show active
 interface Vlan18
   ip verify unicast source reachable-via rx
switch(config-if-Vl18)#
```

13.1.16.44 ipv4 routable 240.0.0.0/4

The **ipv4 routable 240.0.0.0/4** command assignes an class E addresses to an interface. When configured, the class E address traffic are routed through BGP, OSPF, ISIS, RIP, static routes and programmed to the FIB and kernel. By default, this command is disabled.

The **no ipv4 routable 240.0.0.0/4** and **default ipv4 routable 240.0.0.0/4** commands disable IPv4 Class E routing by removing the **ipv4 routable 240.0.0.0/4** command from ***running-config***.

IPv4 routable ***240.0.0.0/4*** routing is disabled by default.

Command Mode

Router General Configuration

Command Syntax

ipv4 routable 240.0.0.0/4

no ipv4 routable 240.0.0.0/4

default ipv4 routable 240.0.0.0/4

Example

These commands configure an IPv4 Class E (***240/4***) address to an interface.

```
switch(config)#router general
switch(config-router-general)#ipv4 routable 240.0.0.0/4
```

13.1.16.45 platform barefoot bfrt vrf

The **platform barefoot bfrt vrf** command configures the forwarding plane agent on supported platforms to restart and listen on the configured VRF for connections. If left unconfigured, the default VRF is used for the IP and port for the the BfRuntime server.

Command Mode

Global Configuration

Command Syntax

platform barefoot bfrt vrf *VRF name*

Parameters

VRF name configured VRF for connections.

Example

These commands configure the forwarding plane agent to restart and listen on the configured VRF for connections.

```
switch(config)#vrf instance management
switch(config-vrf-management)#exit
switch(config)#platform barefoot bfrt 0.0.0.0 50052
switch(config)#platform barefoot bfrt vrf <VRF name>
switch(config)#int management1
switch(config-if-Ma1)#vrf management
```

13.1.16.46 platform trident forwarding-table partition

The **platform trident forwarding-table partition** command provides a shared table memory for L2, L3 and algorithmic LPM entries that can be partitioned in different ways.

Instead of having fixed-size tables for L2 MAC entry tables, L3 IP forwarding tables, and Longest Prefix Match (LPM) routes, the tables can be unified into a single shareable forwarding table.

Note: Changing the Unified Forwarding Table mode causes the forwarding agent to restart, briefly disrupting traffic forwarding on all ports.

The **no platform trident forwarding-table partition** and **default platform trident forwarding-table partition** commands remove the **platform trident forwarding-table partition** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

platform trident forwarding-table partition SIZE

no platform trident forwarding-table partition

default platform trident forwarding-table partition

Parameters

SIZE Size of partition. Options include:

- ***0*** 288k l2 entries, 16k host entries, 16k lpm entries.
- ***1*** 224k l2 entries, 80k host entries, 16k lpm entries.
- ***2*** 160k l2 entries, 144k host entries, 16k lpm entries.
- ***3*** 96k l2 entries, 208k host entries, 16k lpm entries.

The default value is ***2*** (160k l2 entries, 144k host entries, 16k lpm entries).

Examples

- This command sets the single shareable forwarding table to option 2 that supports 160k L2 entries, 144k host entries, and 16k LPM entries.

```
switch(config)#platform trident forwarding-table partition 2
switch(config)
```

- This command sets the single shareable forwarding table to option 3 that supports 96k L2 entries, 208k host entries, and 16k LPM entries. Since the switch was previously configured to option 2, you'll see a warning notice before the changes are implemented.

```
#switch(config)#platform trident forwarding-table partition 3

Warning: StrataAgent will restart immediately
```

13.1.16.47 platform trident routing-table partition

The **platform trident routing-table partition** command manages the partition sizes for the hardware LPM table that stores IPv6 routes of varying sizes.

An IPv6 route of length /64 (or shorter) requires half the hardware resources of an IPv6 route that is longer than /64. The switch installs routes of varying lengths in different table partitions. This command specifies the size of these partitions to optimize table usage.

Note: Changing the routing table partition mode causes the forwarding agent to restart, briefly disrupting traffic forwarding on all ports.

The **no platform trident routing-table partition** and **default platform trident routing-table partition** commands restore the default partitions sizes by removing the **platform trident routing-table partition** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

platform trident routing-table partition SIZE

no platform trident routing-table partition

default platform trident routing-table partition

Parameters

SIZE Size of partition. Options include:

- ***1*** 16k IPv4 entries, 6k IPv6 (/64 and smaller) entries, 1k IPv6 (any prefix length).
- ***2*** 16k IPv4 entries, 4k IPv6 (/64 and smaller) entries, 2k IPv6 (any prefix length).
- ***3*** 16k IPv4 entries, 2k IPv6 (/64 and smaller) entries, 3k IPv6 (any prefix length).

The default value is ***2*** (16k IPv4 entries, 4k IPv6 (/64 and smaller) entries, 2k IPv6 (any prefix length).

Restrictions

Partition allocation cannot be changed from the default setting when uRPF is enabled for IPv6 traffic.

Example

This command sets the shareable routing table to option ***1*** that supports ***6K*** prefixes equal to or shorter than ***/64*** and ***1K*** prefixes longer than ***/64***.

```
switch(config)#platform trident routing-table partition 1
switch(config)
```

13.1.16.48 rib fib policy

The **`rib fib policy`** command enables FIB policy for a particular VRF under router general configuration mode.The FIB policy can be configured to advertise only specific RIB routes and exclude all other routes.

For example, a FIB policy can be configured that will not place routes associated with a specific origin in the routing table. These routes will not be used to forward data packets and these routes are not advertised by the routing protocol to neighbors.

The **`no rib fib policy`** and **`default rib fib policy`** commands restore the switch to its default state by removing the corresponding rib fib policy command from ***running-config*** .

Command Mode

Router General Configuration

Command Syntax

`rib` [*ipv4* | *ipv6*] fib policy *name*

`no rib` [*ipv4* | *ipv6*] fib policy *name*

`default rib` [*ipv4* | *ipv6*] fib policy *name*

Parameters

- ***ipv4*** IPv4 configuration commands.
- ***ipv6*** IPv6 configuration commands.
- ***name*** Route map name.

Example

The following example enables FIB policy for IPv4 in the default VRF, using the route map, ***map1***.

```
Switch(config)#router general
Switch(config-router-general)#vrf default
Switch(config-router-general-vrf-default)#rib ipv4 fib policy map1
```

13.1.16.49 show arp

The **`show arp`** command displays all ARP tables. This command differs from the show ip arp command in that it shows MAC bindings for all protocols, whereas show ip arp only displays MAC address – IP address bindings. Addresses are displayed as their host name by including the ***resolve*** argument.

Command Mode

EXEC

`show arp` [**VRF_INST**][**FORMAT**][**HOST_ADD**][**HOST_NAME**][**INTF**][**MAC_ADDR**][**DATA**]

Parameters

The **VRF_INST** and **FORMAT** parameters are always listed first and second. The **DATA** parameter is always listed last. All other parameters can be placed in any order.

- **VRF_INST** Specifies the VRF instance for which data is displayed.
  - ***no parameter*** Context-active VRF.
  - **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **`default`**.
- **FORMAT** Displays format of host address. Options include:
  - ***no parameter*** Entries associate hardware address with an IPv4 address.
  - **resolve** Enter associate hardware address with a host name (if it exists).

- **HOST_ADD** IPv4 address by which routing table entries are filtered. Options include:
  - ***no parameter*** Routing table entries are not filtered by host address.
  - ***ipv4_addr*** Table entries matching specified IPv4 address.
- **HOST_NAME** Host name by which routing table entries are filtered. Options include:
  - ***no parameter*** Routing table entries are not filtered by host name.
  - **host *hostname*** Entries matching ***hostname*** (text).
- **INTF** Interfaces for which command displays status.
  - ***no parameter*** Routing table entries are not filtered by interface.
  - **interface ethernet *e_num*** Routed Ethernet interface specified by ***e_num***.
  - **interface loopback *l_num*** Routed loopback interface specified by ***l_num***.
  - **interface management *m_num*** Routed management interface specified by ***m_num***.
  - **interface port-channel *p_num*** Routed port channel Interface specified by ***p_num***.
  - **interface vlan *v_num*** VLAN interface specified by ***v_num***.
  - **interface vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- **MAC_ADDR** MAC address by which routing table entries are filtered. Options include:
  - ***no parameter*** Routing table entries are not filtered by interface MAC address.
  - **mac_address *mac_address*** Entries matching ***mac_address*** (dotted hex notation – H.H.H).
- **DATA** Detail of information provided by command. Options include:
  - ***no parameter*** Routing table entries.
  - **summary** Summary of ARP table entries.
  - **summary total** Number of ARP table entries.

Related Commands

The cli vrf command specifies the context-active VRF.

Example

This command displays the ARP table.

```
switch>show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.133           0  001c.7304.3906  Management1 switch>
```

13.1.16.50 show dhcp server

Use the **show dhcp server** command to display DHCP server information.

Command Mode

EXEC

Command Syntax

show dhcp server [ipv4 | ipv6 [leases [**A.B.C.D/E** | **NAME**]]] | [leases [ipv4 | ipv6 | **NAME**]]

Parameters

- **ipv4** Displays details related to IPv4.
- **ipv6** Displays details related to IPv6.
- **leases** Displays active leases.
  - **A.B.C.D/E** IPv4 subnet.

- **NAME** Subnet name.

Examples

- DHCPv4 display example.

```
switch#show dhcp server ipv4
IPv4 DHCP Server is active
Debug log is enabled
DNS server(s): 10.2.2.2
DNS domain name: domainFoo
Lease duration: 1 days 0 hours 0 minutes
TFTP server:
serverFoo (Option 66)
10.0.0.3 (Option 150)
TFTP file: fileFoo
Active Leases: 1
IPv4 DHCP interface status:
   Interface   Status
------------------------------------------------
   Ethernet1   Inactive (Could not determine VRF)
   Ethernet2   Inactive (Not in default VRF)
   Ethernet3   Inactive (Kernel interface not created yet)
   Ethernet4   Inactive (Not up)
   Ethernet5   Inactive (No IP address)
   Ethernet6   Active

Vendor information:
Vendor ID: default
  Sub-options         Data
---------------- ----------------
      1          192.0.2.0, 192.0.2.1

Vendor ID: vendorFoo
  Sub-options       Data
---------------- -----------
      2            192.0.2.2
      3            "Foo"

Subnet: 10.0.0.0/8
Subnet name: subnetFoo
Range: 10.0.0.1 to 10.0.0.10
DNS server(s): 10.1.1.1 10.2.2.2
Lease duration: 3 days 3 hours 3 minutes
Default gateway address: 10.0.0.3
TFTP server:
subnetServerFoo (Option 66)
10.0.0.4 (Option 150)
TFTP boot file: subnetFileFoo
Active leases: 1
Reservations:
MAC address: 1a1b.1c1d.1e1f
IPv4 address: 10.0.0.1

MAC address: 2a2b.2c2d.2e2f
IPv4 address: 10.0.0.2
```

- In this example, DHCPv6 is configured with subnet ***fe80::/10*** while being enabled on ***Ethernet1*** with address ***fe80::1/64*** and on ***Ethernet3*** with address ***fe80::2/64***.

```
switch#show dhcp server ipv6
IPv6 DHCP server is active
Debug log is enabled
```

```
DNS server(s): fe80::6


DNS domain name: testaristanetworks.com


Lease duration: 1 days 3 hours 30 minutes


Active leases: 0


IPv6 DHCP interface status:


   Interface    Status


--------------- ------


   Ethernet1    Active
   Ethernet3    Active









Subnet: fe80::/10


Subnet name: foo




Range: fe80::1 to fe80::3
DNS server(s): fe80::4 fe80::5




Direct: Inactive (Multiple interfaces match this subnet: Ethernet1
 Ethernet3)
Relay: Active
```

```
Active leases: 0
```

- This example illustrates when multiple subnets match an interface. In this example, DHCPv6 is configured with subnets ***fc00::/7*** and **fe80::/10** while being enabled on ***Ethernet1*** with address ***fe80::1/10*** and ***fc00::1/7***.

```
switch#show dhcp server ipv6
IPv6 DHCP server is active


DNS server(s):  fc00::2


DNS domain name: testaristanetworks.com


Lease duration: 1 days 3 hours 30 minutes


Active leases: 0


IPv6 DHCP interface status:


   Interface    Status


--------------- ------


   Ethernet1    Active









Subnet: fc00::/7


Subnet name: foo


Range: fc00::1 to fc00::5
```

```
DNS server(s): fc00::6 fc00::8




Direct: Inactive (This and other subnets match interface Ethernet1)
Relay: Active



Active leases: 0

Subnet: fe80::/10



Subnet name: bar




Direct: Inactive (This and other subnets match interface Ethernet1)
Relay: Active



Active leases: 0
```

- When a subnet is disabled, the **show dhcp server** command displays the disable message with a reason. The number of active leases of the disabled subnets will be ***0***. In this example, there are overlapping subnets.

```
switch#show dhcp server
IPv4 DHCP Server is active
DNS server(s): 10.2.2.2
Lease duration: 1 days 0 hours 0 minutes
Active Leases: 0
IPv4 DHCP interface status:
   Interface   Status
-----------------------------------------------
   Ethernet1   Active

Subnet: 10.0.0.0/24 (Subnet is disabled - overlapping subnet 10.0.0.0/8)
Range: 10.0.0.1 to 10.0.0.10
DNS server(s): 10.3.3.3 10.4.4.4
Default gateway address: 10.0.0.4
Active leases: 0

Subnet: 10.0.0.0/8 (Subnet is disabled - overlapping subnet 10.0.0.0/24)
DNS server(s):
Default gateway address: 10.0.0.3
Active leases: 0
```

- In this example, the display output shows overlapping ranges.

```
switch#show dhcp server
IPv4 DHCP Server is active
DNS server(s): 10.2.2.2
Lease duration: 1 days 0 hours 0 minutes
Active Leases: 0
IPv4 DHCP interface status:
```

```
   Interface   Status
-------------------------------------------------
   Ethernet1   Active

Subnet: 10.0.0.0/8 (Subnet is disabled - range 10.0.0.9-10.0.0.12 overlaps
 with an existing pool)
Range: 10.0.0.1 to 10.0.0.10
Range: 10.0.0.9 to 10.0.0.12
DNS server(s): 10.3.3.3 10.4.4.4
Default gateway address: 10.0.0.4
Active leases: 0
```

- This example shows duplicate static IP address reservation.

```
Subnet: 10.0.0.0/8 (Subnet is disabled - ipv4-address 10.0.0.11 is reserved
 more than once)
Subnet name:
DNS server(s):
Default gateway address: 10.0.0.3
Active leases: 0
Reservations:
MAC address: 1a1b.1c1d.1e1f
IPv4 address: 10.0.0.11

MAC address: 2a2b.2c2d.2e2f
IPv4 address: 10.0.0.11
```

- Use the **`show dhcp server leases`** command to display detailed information about the IP addresses allocated by the DHCP Server (including the IP address, the expected end time for that address, the time when the address is handed out, and the equivalent MAC address).

```
switch#show dhcp server leases
10.0.0.10
End: 2019/06/20 17:44:34 UTC
Last transaction: 2019/06/19 17:44:34 UTC
MAC address: 5692.4c67.460a

2000:0:0:40::b



End: 2019/06/20 18:06:33 UTC



Last transaction: 2019/06/20 14:36:33 UTC



MAC address: 165a.a86d.ffac
```

13.1.16.51 show hardware capacity

The **`show hardware capacity`** command displays the utilization of the hardware resources:

Command Mode

Privileged EXEC

Command Syntax

`show hardware capacity`

Example

- The following command is used to show the utilization of the hardware resources:

```
switch#show hardware capacity
Forwarding Resources Usage

Table          Feature           Chip              Used Used    Free Committed
 Best Case     High
                                                      Entries  (%) Entries
 Entries       Max   Watermark

                     Entries
---------------- ------------------- --------------- ---------- -------
 ---------- ------------ ------------ ---------
ECMP                                                        0   0%    4095
     0      4095        0
ECMP           Mpls                                     0   0%    4095
  0       4095        0
ECMP           Routing                                  0   0%    4095
  0       4095        0
ECMP           VxlanOverlay                             0   0%    4095
  0       4095        0
ECMP           VxlanTunnel                              0   0%    3891
  0       3891        0
```

13.1.16.52 show interface tunnel

The **`show interface tunnel`** command displays the interface tunnel information.

Command Mode

EXEC

Command Syntax

`show interface tunnel` *number*

Parameter

number Specifies the tunnel interface number.

Example

This command displays tunnel interface configuration information for tunnel interface ***10***.

```
switch#show interface tunnel 10

Tunnel10 is up, line protocol is up (connected)
 Hardware is Tunnel, address is 0a01.0101.0800
 Internet address is 192.168.1.1/24
 Broadcast address is 255.255.255.255
 Tunnel source 10.1.1.1, destination 10.1.1.2
 Tunnel protocol/transport GRE/IP
   Key disabled, sequencing disabled
   Checksumming of packets disabled
 Tunnel TTL 10, Hardware forwarding enabled
 Tunnel TOS 10
 Path MTU Discovery
 Tunnel transport MTU 1476 bytes
 Up 3 seconds
```

13.1.16.53 show ip

The **show ip** command displays IPv4 routing, IPv6 routing, IPv4 multicast routing, and VRRP status on the switch.

Command Mode

EXEC

Command Syntax

show ip

Example

This command displays IPv4 routing status.

```
switch>show ip

IP Routing : Enabled
IP Multicast Routing : Disabled
VRRP: Configured on 0 interfaces

IPv6 Unicast Routing : Enabled
IPv6 ECMP Route support : False
IPv6 ECMP Route nexthop index: 5
IPv6 ECMP Route num prefix bits for nexthop index: 10 switch>
```

13.1.16.54 show ip arp

The **show ip arp** command displays ARP cache entries that map an IPv4 address to a corresponding MAC address. The table displays addresses by their host names when the command includes the ***resolve*** argument.

Command Mode

EXEC

Command Syntax

show ip arp [**VRF_INST**][**FORMAT**][**HOST_ADD**][**HOST_NAME**][**INTF**][**MAC_ADDR**][**DATA**]

Parameters

The **VRF_INST** and **FORMAT** parameters are always listed first and second. The **DATA** parameter is always listed last. All other parameters can be placed in any order.

- **VRF_INST** Specifies the VRF instance for which data is displayed.
  - ***no parameter*** Context-active VRF.
  - **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **default**.
- **FORMAT** Displays format of host address. Options include:
  - ***no parameter*** Entries associate hardware address with an IPv4 address.
  - **resolve** Enter associate hardware address with a host name (if it exists).
- **HOST_ADDR** IPv4 address by which routing table entries are filtered. Options include:
  - ***no parameter*** Routing table entries are not filtered by host address.
  - ***ipv4_addr*** Table entries matching specified IPv4 address.
- **HOST_NAME** Host name by which routing table entries are filtered. Options include:
  - ***no parameter*** Routing table entries are not filtered by host name.

- **host *hostname*** Entries matching ***hostname*** (text).

- **INTERFACE_NAME** Interfaces for which command displays status.
  - ***no parameter*** Routing table entries are not filtered by interface.
  - **interface ethernet *e_num*** Routed Ethernet interface specified by ***e_num***.
  - **interface loopback *l_num*** Routed loopback interface specified by ***l_num***.
  - **interface management *m_num*** Routed management interface specified by ***m_num***.
  - **interface port-channel *p_num*** Routed port channel Interface specified by ***p_num***.
  - **interface vlan *v_num*** VLAN interface specified by ***v_num***.
  - **interface vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- MAC_ADDR MAC address by which routing table entries are filtered. Options include:
  - ***no parameter*** Routing table entries are not filtered by interface MAC address.
  - **mac_address *mac_address*** entries matching ***mac_address*** (dotted hex notation – H.H.H).
- **DATA** Detail of information provided by command. Options include:
  - ***no parameter*** Routing table entries.
  - **summary** Summary of ARP table entries.
  - **summary total** Number of ARP table entries.

Related Commands

The cli vrf command specifies the context-active VRF.

Examples

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses.

```
switch>show ip arp

Address         Age (min)  Hardware Addr   Interface
172.25.0.2              0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
172.22.0.2              0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
172.22.0.5              0  f012.3118.c09d  Vlan1000, Port-Channel1
172.22.0.6              0  00e1.d11a.a1eb  Vlan1000, Ethernet5
172.22.0.7              0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
172.22.0.9              0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
 switch>
```

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses. Host names assigned to IP addresses are displayed in place of the address.

```
switch>show ip arp resolve

Address         Age (min)  Hardware Addr   Interface
green-vl101.new         0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
orange-vl1000.n         0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
purple.newcompa         0  f012.3118.c09d  Vlan1000, Port-Channel1
pink.newcompany         0  00e1.d11a.a1eb  Vlan1000, Ethernet5
yellow.newcompa         0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
royalblue.newco         0  0018.910a.1fc5  Vlan1000, Ethernet29
```

```
172.22.0.11              0  0056.cbe9.8510  Vlan1000, Ethernet26
 switch>
```

13.1.16.55 show ip arp inspection statistics

The **show ip arp inspection statistics** command displays the statistics of inspected ARP packets. For a VLAN specified, only VLANs with ARP inspection enabled will be displayed. If no VLAN is specified, all VLANs with ARP inspection enabled are displayed.

Command Mode

EXEC

Command Syntax

show ip arp inspection statistics [vlan [**VID**]|[**INTERFACE**] interface *intf_slot* | *intf_port*]

Parameters

- **VID** Specifies the VLAN interface ID.
- **INTERFACE** Specifies the interface (e.g., Ethernet).
  - ***intf_slot*** Interface slot.
  - ***intf_port*** Interface port.
- **INTF** Specifies the VLAN interface slot and port.

Related Commands

- ip arp inspection limit
- ip arp inspection trust
- ip arp inspection vlan

Examples

- This command display statistics of inspected ARP packets for VLAN ***10***.

```
switch(config)#show ip arp inspection statistics vlan 10

Vlan : 10
--------------
ARP
Req Forwarded = 20
ARP Res Forwarded = 20
ARP Req Dropped = 1
ARP Res Dropped = 1
Last invalid ARP:
Time: 10:20:30 ( 5 minutes ago )
Reason: Bad IP/Mac match
Received on: Ethernet 3/1
Packet:
  Source MAC: 00:01:00:01:00:01
  Dest MAC: 00:02:00:02:00:02
  ARP Type: Request
  ARP Sender MAC: 00:01:00:01:00:01
  ARP Sender IP: 1.1.1

switch(config)#
```

- This command displays ARP inspection statistics for Ethernet interface ***3/1***.

```
switch(config)#show ip arp inspection statistics ethernet interface
 3/1
Interface : 3/1
--------
ARP Req Forwarded = 10
ARP Res Forwarded = 10
ARP Req Dropped = 1
ARP Res Dropped = 1

Last invalid ARP:
Time: 10:20:30 ( 5 minutes ago )
Reason: Bad IP/Mac match
Received on: VLAN 10
Packet:
  Source MAC: 00:01:00:01:00:01
  Dest MAC: 00:02:00:02:00:02
  ARP Type: Request
  ARP Sender MAC: 00:01:00:01:00:01
  ARP Sender IP: 1.1.1

switch(config)#
```

13.1.16.56 show ip arp inspection vlan

The **`show ip arp inspection vlan`** command displays the configuration and operation state of ARP inspection. For a VLAN range specified, only VLANs with ARP inspection enabled will be displayed. If no VLAN is specified, all VLANs with ARP inspection enabled are displayed. The operation state turns to ***Active*** when hardware is ready to trap ARP packets for inspection.

Command Mode

EXEC

Command Syntax

`show ip arp inspection vlan` [**LIST**]

Parameters

LIST Specifies the VLAN interface number.

Related Commands

- ip arp inspection limit
- ip arp inspection trust
- show ip arp inspection statistics

Example

This command displays the configuration and operation state of ARP inspection for VLANs ***1*** through ***150***.

```
switch(config)#show ip arp inspection vlan 1 - 150

VLAN 1
----------
Configuration
: Enabled
Operation State : Active
```

```
VLAN 2
----------
Configuration
: Enabled
Operation State : Active
{...}
VLAN 150
----------
Configuration
: Enabled
Operation State : Active

switch(config)#
```

13.1.16.57 show ip dhcp relay counters

The **show ip dhcp relay counters** command displays the number of DHCP packets received, forwarded, or dropped on the switch and on all interfaces enabled as DHCP relay agents.

Command Mode

EXEC

Command Syntax

show ip dhcp relay counters

Example

This command displays the IP DHCP relay counter table.

```
switch>show ip dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|----------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
 Vlan1000 |    0    0    0 | 4 days, 19:54:24 ago
 Vlan1036 |  376  277    0 | 4 days, 19:54:24 ago switch>
```

13.1.16.58 show ip dhcp relay

The **show ip dhcp relay** command displays the DHCP relay agent configuration status on the switch.

Command Mode

EXEC

Command Syntax

show ip dhcp relay

Example

This command displays the DHCP relay agent configuration status.

```
switch>show ip dhcp relay
```

```
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCP Smart Relay is enabled
Interface: Vlan100
  DHCP Smart Relay is disabled
  DHCP servers: 10.4.4.4 switch>
```

13.1.16.59 show ip dhcp snooping counters

The **show ip dhcp snooping counters** command displays counters that track the quantity of DHCP request and reply packets that the switch receives. Data is either presented for each VLAN or aggregated for all VLANs with counters for packets dropped.

Command Mode

EXEC

Command Syntax

show ip dhcp snooping counters [**COUNTER_TYPE**]

Parameters

COUNTER_TYPE The type of counter that the command resets. Formats include:

- ***no parameter*** Command displays counters for each VLAN.
- **debug** Command displays aggregate counters and drop cause counters.

Examples

- This command displays the number of DHCP packets sent and received on each VLAN.

```
switch>show ip dhcp snooping counters

     | Dhcp Request Pkts | Dhcp Reply Pkts |
Vlan |  Rcvd  Fwdd  Drop | Rcvd Fwdd  Drop | Last Cleared
-----|------ ----- ------|----- ---- ------|-------------
 100 |     0     0     0 |    0    0     0 |  0:35:39 ago switch>
```

- This command displays the number of DHCP packets sent on the switch.

```
switch>show ip dhcp snooping counters debug
Counter                        Snooping to Relay Relay to Snooping
------------------------------ ----------------- -----------------
Received                                       0                 0
Forwarded                                      0                 0
Dropped - Invalid VlanId                       0                 0
Dropped - Parse error                          0                 0
Dropped - Invalid Dhcp Optype                  0                 0
Dropped - Invalid Info Option                  0                 0
Dropped - Snooping disabled                    0                 0

Last Cleared:  3:37:18 ago switch>
```

13.1.16.60 show ip dhcp snooping hardware

The **show ip dhcp snooping hardware** command displays internal hardware DHCP snooping status on the switch.

Command Mode

EXEC

Command Syntax

`show ip dhcp snooping hardware`

Example

This command DHCP snooping hardware status.

```
switch>show ip dhcp snooping hardware
DHCP Snooping is enabled
DHCP Snooping is enabled on following VLANs:
    None
    Vlans enabled per Slice
        Slice:  FixedSystem
        None switch>
```

13.1.16.61 show ip dhcp snooping

The **`show ip dhcp snooping`** command displays the DHCP snooping configuration.

Command Mode

EXEC

Command Syntax

`show ip dhcp snooping`

Related Commands

- ip dhcp snooping globally enables DHCP snooping.
- ip dhcp snooping vlan enables DHCP snooping on specified VLANs.
- ip dhcp relay information option (Global) enables insertion of option-82 snooping data.
- ip helper-address enables the DHCP relay agent on a configuration mode interface.

Example

This command displays the switch's DHCP snooping configuration.

```
switch>show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is operational
DHCP Snooping is configured on following VLANs:
  100
DHCP Snooping is operational on following VLANs:
  100
Insertion of Option-82 is enabled
  Circuit-id format: Interface name:Vlan ID
  Remote-id: 00:1c:73:1f:b4:38 (Switch MAC)
switch>
```

13.1.16.62 show ip hardware fib summary

The **`show ip hardware fib summary`** command displays the statistics of the RECMP.

Command Mode

Privileged EXEC

Command Syntax

`show ip hardware fib summary`

Example

- The following command is used to show the statistics of RECMP:

```
switch#show ip hardware fib summary
Fib summary
-----------
Adjacency sharing: disabled
BFD peer event: enabled
Deletion Delay: 0
Protect default route: disabled
PBR: supported
URPF: supported
ICMP unreachable: enabled
Max Ale ECMP: 600
UCMP weight deviation: 0.0
Maximum number of routes: 0
Fib compression: disabled
Resource optimization for adjacency programming: enabled
Adjacency resource optimization thresholds: low 20, high 80
```

About Output

- The last two lines of the output shows whether the feature is enabled and what are the corresponding threshold values for starting and stopping the optimization process.

13.1.16.63 show ip interface

The **`show ip interface`** command displays the status of specified interfaces that are configured as routed ports. The command provides the following information:

- Interface description
- Internet address
- Broadcast address
- Address configuration method
- Proxy-ARP status
- MTU size

Command Mode

EXEC

Command Syntax

`show ip interface` [**INTERFACE_NAME**][**VRF_INST**]

Parameters

- **INTERFACE_NAME** Interfaces for which command displays status.
  - ***no parameter*** All routed interfaces.
  - ***ipv4_addr*** Neighbor IPv4 address.
  - **ethernet *e_range*** Routed Ethernet interfaces specified by ***e_range***.
  - **loopback *l_range*** Routed loopback interfaces specified by ***l_range***.
  - **management *m_range*** Routed management interfaces specified by ***m_range***.
  - **port-channel *p_range*** Routed port channel Interfaces specified by ***p_range***.
  - **vlan *v_range*** VLAN interfaces specified by ***v_range***.

  - **vxlan *vx_range*** VXLAN interfaces specified by ***vx_range***.
- **VRF_INST** Specifies the VRF instance for which data is displayed.
  - ***no parameter*** Context-active VRF.
  - **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **`default`**.

Example

- This command displays IP status of configured VLAN interfaces numbered between ***900*** and ***910***.

```
switch>show ip interface vlan 900-910
! Some interfaces do not exist
Vlan901 is up, line protocol is up (connected)
  Description: ar.pqt.mlag.peer
  Internet address is 170.23.254.1/30
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  Proxy-ARP is disabled
  MTU 9212 bytes
Vlan903 is up, line protocol is up (connected)
  Description: ar.pqt.rn.170.23.254.16/29
  Internet address is 170.23.254.19/29
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  Proxy-ARP is disabled
  MTU 9212 bytes
```

- This command displays the configured TCP Maximum Segment Size (MSS) ceiling value of ***1436*** bytes for an Ethernet interface ***25***.

```
switch>show ip interface ethernet 25
Ethernet25 is up, line protocol is up (connected)
  Internet address is 10.1.1.1/24
  Broadcast address is 255.255.255.255
  IPv6 Interface Forwarding : None
  Proxy-ARP is disabled
  Local Proxy-ARP is disabled
  Gratuitous ARP is ignored
  IP MTU 1500 bytes
  IPv4 TCP MSS egress ceiling is 1436 bytes
```

13.1.16.64 show ip interface brief

Use the **`show ip interface brief`** command output to display the status summary of the specified interfaces that are configured as routed ports. The command provides the following information for each specified interface:

- IP address
- Operational status
- Line protocol status
- MTU size

Command Mode

EXEC

Command Syntax

`show ip interface` [**INTERFACE_NAME**][**VRF_INST**] brief

Parameters

- **INTERFACE_NAME** Interfaces for which command displays status.
  - ***no parameter*** All routed interfaces.
  - ***ipv4_addr*** Neighbor IPv4 address.
  - **ethernet *e_range*** Routed Ethernet interfaces specified by ***e_range***.
  - **loopback *l_range*** Routed loopback interfaces specified by ***l_range***.
  - **management *m_range*** Routed management interfaces specified by ***m_range***.
  - **port-channel *p_range*** Routed port channel Interfaces specified by ***p_range***.
  - **vlan *v_range*** VLAN interfaces specified by ***v_range***.
  - **vxlan *vx_range*** VXLAN interface range specified by ***vx_range***.
- **VRF_INST** Specifies the VRF instance for which data is displayed.
  - ***no parameter*** Context-active VRF.
  - **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **`default`**.

Example

This command displays the summary status of VLAN interfaces ***900-910***.

```
switch>show ip interface vlan 900-910 brief

! Some interfaces do not exist
Interface              IP Address         Status     Protocol
 MTU
Vlan901                170.33.254.1/30    up         up
 9212
Vlan902                170.33.254.14/29   up         up
 9212
Vlan905                170.33.254.17/29   up         up
 1500
Vlan907                170.33.254.67/29   up         up
 9212
Vlan910                170.33.254.30/30   up         up
 9212
```

13.1.16.65 show ip route

The **`show ip route`** command displays routing table entries that are in the Forwarding Information Base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal-cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **`show running-config`** command displays configured commands not in the FIB.

Command Mode

EXEC

Command Syntax

`show ip route` [**VRF_INSTANCE**][**ADDRESS**][**ROUTE_TYPE**][**INFO_LEVEL**][**PREFIX**]

Parameters

The **VRF_INSTANCE** and **ADDRESS** parameters are always listed first and second, respectively. All other parameters can be placed in any order.

- **VRF_INSTANCE** Specifies the VRF instance for which data is displayed.

  - ***no parameter*** Context-active VRF.
  - **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **`default`**.
- **ADDRESS** Filters routes by IPv4 address or subnet.
  - ***no parameter*** All routing table entries.
  - ***ipv4_addr*** Routing table entries matching specified address.
  - ***ipv4_subnet*** Routing table entries matching specified subnet (CIDR or address-mask).
- **ROUTE_TYPE** Filters routes by specified protocol or origin. Options include:
  - ***no parameter*** All routing table entries.
  - **aggregate** Entries for BGP aggregate routes.
  - **bgp** Entries added through BGP protocol.
  - **connected** Entries for routes to networks directly connected to the switch.
  - ***isis*** Entries added through ISIS protocol.
  - **kernel** Entries appearing in Linux kernel but not added by EOS software.
  - **ospf** Entries added through OSPF protocol.
  - **rip** Entries added through RIP protocol.
  - **static** Entries added through CLI commands.
  - **vrf** Displays routes in a VRF.
- **INFO_LEVEL** Filters entries by next hop connection. Options include:
  - ***no parameter*** Filters routes whose next hops are directly connected.
  - **detail** Displays all routes.
- **PREFIX** Filters routes by prefix.
  - ***no parameter*** Specific route entry that matches the ADDRESS parameter.
  - **longer-prefixes** All subnet route entries in range specified by ADDRESS parameter.

Related Commands

The **cli vrf** command specifies the context-active VRF.

Examples

- This command displays IPv4 routes learned through BGP.

```
switch>show ip route bgp
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, A - Aggregate

 B E    170.44.48.0/23 [20/0] via 170.44.254.78
 B E    170.44.50.0/23 [20/0] via 170.44.254.78
 B E    170.44.52.0/23 [20/0] via 170.44.254.78
 B E    170.44.54.0/23 [20/0] via 170.44.254.78
 B E    170.44.254.112/30 [20/0] via 170.44.254.78
 B E    170.53.0.34/32 [1/0] via 170.44.254.78
 B I    170.53.0.35/32 [1/0] via 170.44.254.2
                             via 170.44.254.13
                             via 170.44.254.20
                             via 170.44.254.67
                             via 170.44.254.35
                             via 170.44.254.98
```

- This command displays the unicast IP routes installed in the system.

```
switch#show ip route
```

```
 VRF name: default
Codes: C - connected, S - static, K - kernel,
 O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
 E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
 N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
 R - RIP, I - ISIS, A B - BGP Aggregate, A O - OSPF Summary,
 NG - Nexthop Group Static Route

Gateway of last resort is not set
 C 10.1.0.0/16 is directly connected, Vlan2659
 C 10.2.0.0/16 is directly connected, Vlan2148
 C 10.3.0.0/16 is directly connected, Vlan2700
 S 172.17.0.0/16 [1/0] via 172.24.0.1, Management1
 S 172.18.0.0/16 [1/0] via 172.24.0.1, Management1
 S 172.19.0.0/16 [1/0] via 172.24.0.1, Management1
 S 172.20.0.0/16 [1/0] via 172.24.0.1, Management1
 S 172.22.0.0/16 [1/0] via 172.24.0.1, Management1
 C 172.24.0.0/18 is directly connected, Management1
```

- This command displays the leaked routes from a source VRF.

```
switch#show ip route vrf VRF2 20.0.0.0/8
...
S L      20.0.0.0/8 [1/0] (source VRF VRF1) via 10.1.2.10, Ethernet1
```

13.1.16.66 show ip route age

The **`show ip route age`** command displays the time when the route for the specified network was present in the routing table. It does not account for the changes in parameters like metric, next-hop etc.

Command Mode

EXEC

Command Syntax

`show ip route` **ADDRESS** age

Parameters

ADDRESS Filters routes by IPv4 address or subnet.

- ***ipv4_addr*** Routing table entries matching specified address.
- ***ipv4_subnet*** Routing table entries matching specified subnet (CIDR or address-mask).

Example

This command shows the amount of time since the last update to ip route ***172.17.0.0/20***.

```
switch>show ip route 172.17.0.0/20 age
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I - ISIS, A - Aggregate

 B E    172.17.0.0/20 via 172.25.0.1, age 3d01h
switch>
```

13.1.16.67 show ip route gateway

The **`show ip route gateway`** command displays IP addresses of all gateways (next hops) used by active routes.

Command Mode

EXEC

Command Syntax

`show ip route` [**VRF_INSTANCE**] gateway

Parameters

VRF_INSTANCE Specifies the VRF instance for which data is displayed.

- ***no parameter*** Context-active VRF.
- **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **`default`**.

Related Commands

The cli vrf command specifies the context-active VRF.

Example

This command displays next hops used by active routes.

```
switch>show ip route gateway
The following gateways are in use:
   172.25.0.1 Vlan101
   172.17.253.2 Vlan3000
   172.17.254.2 Vlan3901
   172.17.254.11 Vlan3902
   172.17.254.13 Vlan3902
   172.17.254.17 Vlan3903
   172.17.254.20 Vlan3903
   172.17.254.66 Vlan3908
   172.17.254.67 Vlan3908
   172.17.254.68 Vlan3908
   172.17.254.29 Vlan3910
   172.17.254.33 Vlan3911
   172.17.254.35 Vlan3911
   172.17.254.105 Vlan3912
   172.17.254.86 Vlan3984
   172.17.254.98 Vlan3992
   172.17.254.99 Vlan3992 switch>
```

13.1.16.68 show ip route host

The **`show ip route host`** command displays all host routes in the host forwarding table. Host routes are those whose destination prefix is the entire address (mask = ***255.255.255.255*** or prefix = ***/32***). Each entry includes a code of the route's purpose:

- **F** static routes from the FIB.
- **R** routes defined because the IP address is an interface address.
- **B** broadcast address.
- **A** routes to any neighboring host for which the switch has an ARP entry.

Command Mode

EXEC

Command Syntax

`show ip route` [**VRF_INSTANCE**] host

Parameters

VRF_INSTANCE Specifies the VRF instance for which data is displayed.

- ***no parameter*** Context-active VRF.
- **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **`default`**.

Related Commands

The cli vrf command specifies the context-active VRF.

Example

This command displays all host routes in the host forwarding table.

```
switch>show ip route host
R - receive B - broadcast F - FIB, A - attached

F   127.0.0.1 to cpu
B   172.17.252.0 to cpu
A   172.17.253.2 on Vlan2000
R   172.17.253.3 to cpu
A   172.17.253.10 on Vlan2000
B   172.17.253.255 to cpu
B   172.17.254.0 to cpu
R   172.17.254.1 to cpu
B   172.17.254.3 to cpu
B   172.17.254.8 to cpu
A   172.17.254.11 on Vlan2902
R   172.17.254.12 to cpu

F   172.26.0.28 via 172.17.254.20 on Vlan3003
                via 172.17.254.67 on Vlan3008
                via 172.17.254.98 on Vlan3492
                via 172.17.254.2 on Vlan3601
                via 172.17.254.13 on Vlan3602
via 172.17.253.2 on Vlan3000
F   172.26.0.29 via 172.25.0.1 on Vlan101
F   172.26.0.30 via 172.17.254.29 on Vlan3910
F   172.26.0.32 via 172.17.254.105 on Vlan3912 switch>
```

13.1.16.69 show ip route match tag

The **`show ip route match tag`** command displays the route tag assigned to the specified IPv4 address or subnet. Route tags are added to static routes for use by route maps.

Command Mode

EXEC

Command Syntax

`show ip route` [**VRF_INSTANCE**] **ADDRESS** match tag

Parameters

- **VRF_INSTANCE** Specifies the VRF instance for which data is displayed.
  - ***no parameter*** Context-active VRF.

  - **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **default**.
- **ADDRESS** Displays routes of specified IPv4 address or subnet.
  - ***ipv4_addr*** Routing table entries matching specified IPv4 address.
  - ***ipv4_subnet*** Routing table entries matching specified IPv4 subnet (CIDR or address-mask).

Example

This command displays the route tag for the specified subnet.

```
switch>show ip route 172.17.50.0/23 match tag
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian

 O E2   172.17.50.0/23 tag 0 switch>
```

13.1.16.70 show ip route summary

The **show ip route summary** command displays the number of routes, categorized by destination prefix, in the routing table.

Command Mode

EXEC

Command Syntax

show ip route [**VRF_INSTANCE**] summary **Parameters**

VRF_INSTANCE Specifies the VRF instance for which data is displayed.

- ***no parameter*** Context-active VRF.
- **vrf *vrf_name*** Specifies name of VRF instance. System default VRF is specified by **default**.

Example

This command displays a summary of the routing table contents.

```
switch>show ip route summary
Route Source         Number Of Routes
-------------------------------------
connected                    15
static                        0
ospf                         74
  Intra-area: 32 Inter-area:33 External-1:0 External-2:9
  NSSA External-1:0 NSSA External-2:0
bgp                           7
  External: 6 Internal: 1
internal                     45
attached                     18
aggregate                     0 switch>
```

13.1.16.71 show ip verify source

The **show ip verify source** command displays the IP source guard (IPSG) configuration, operational states, and IP-MAC binding entries for the configuration mode interface.

Command Mode

EXEC

Command Syntax

show ip verify source [VLAN | DETAIL]

Parameters

- **VLAN** Displays all VLANs configured in **no ip verify source vlan**.
- **DETAIL** Displays all source IP-MAC binding entries configured for IPSG.

Related Commands

- ip source binding
- ip verify source

Examples

- This command verifies the IPSG configuration and operational states.

```
switch(config)#show ip verify source
Interface       Operational State
--------------- ------------------------
Ethernet1       IP source guard enabled
Ethernet2       IP source guard disabled
```

- This command displays all VLANs configured in **no ip verify source vlan**. Hardware programming errors, e.g.,VLAN classification failed, are indicated in the operational state. If an error occurs, this VLAN will be considered as enabled for IPSG. Traffic on this VLAN will still be filtered by IPSG.

```
switch(config)#show ip verify source vlan
IPSG disabled on VLANS: 1-2
VLAN            Operational State
--------------- ------------------------
1               IP source guard disabled
2               Error: vlan classification failed
```

- This command displays all source IP-MAC binding entries configured for IPSG. A source binding entry is considered active if it is programmed in hardware. IP traffic matching any active binding entry will be permitted. If a source binding entry is configured on an interface or a VLAN whose operational state is IPSG disabled, this entry will not be installed in the hardware, in which case an “IP source guard disabled” state will be shown. If a port channel has no member port configured, binding entries configured for this port channel will not be installed in hardware, and a “Port-Channel down” state will be shown.

```
switch(config)#show ip verify source detail
Interface      IP Address  MAC Address     VLAN  State
-------------- ----------- --------------- ----- ------------------------
Ethernet1      10.1.1.1    0000.aaaa.1111   5     active
Ethernet1      10.1.1.5    0000.aaaa.5555   1     IP source guard disabled
Port-Channel1  20.1.1.1    0000.bbbb.1111   4     Port-Channel down
```

13.1.16.72 show platform arad ip route summary

The **`show platform arad ip route summary`** command shows hardware resource usage of IPv4 routes.

Command Mode

EXEC

Command Syntax

`show platform arad ip route summary`

Related Commands

- The agent SandL3Unicast terminate command enables restarting the layer 3 agent to ensure IPv4 routes are optimized.
- The ip hardware fib optimize command enables IPv4 route scale.
- The show platform arad ip route command shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.

Example

This command shows hardware resource usage of IPv4 routes.

```
switch(config)#show platform arad ip route summary
Total number of VRFs: 1
Total number of routes: 25
Total number of route-paths: 21
Total number of lem-routes: 4

switch(config)#
```

13.1.16.73 show platform arad ip route

The **`show platform arad ip route`** command shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.

Command Mode

EXEC

Command Syntax

`show platform arad ip route`

Examples

- This command displays the platform unicast forwarding routes. In this example, the ACL label field in the following table is ***4094*** by default for all routes. If an IPv4 egress RACL is applied to an SVI, all routes corresponding to that VLAN will have an ACL label value. In this case, the ACL Label field value is *2*.

```
switch#show platform arad ip route
 Tunnel Type: M(mpls), G(gre)

-------------------------------------------------------------------------------
|                               Routing Table                                 |
|
|-----------------------------------------------------------------------------
|VRF|   Destination    |      |                   |     | Acl   |             |
ECMP| FEC | Tunnel
| ID|   Subnet         | Cmd  |       Destination  | VID | Label |  MAC / CPU
Code |Index|Index|T Value
```

```
-------------------------------------------------------------------------------
|0  |0.0.0.0/8          |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1031 | -
|0  |10.1.0.0/16        |TRAP | CoppSystemL3DstMiss|2659 | - | ArpTrap | - |1030 | -
|0  |10.2.0.0/16        |TRAP | CoppSystemL3DstMiss|2148 | - | ArpTrap | - |1026 | -
|0  |172.24.0.0/18      |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1032 | -
|0  |0.0.0.0/0          |TRAP | CoppSystemL3LpmOver|0    | - | SlowReceive | -
|1024 | -
|0  |10.1.0.0/32*       |TRAP | CoppSystemIpBcast  |0    | - | BcastReceive | -
|1027 | -
|0  |10.1.0.1/32*       |TRAP | CoppSystemIpUcast  |0    | - | Receive | - |32766| -
|0  |10.1.255.1/32*     |ROUTE| Po1                |2659 |4094 | 00:1f:5d:6b:ce:45
| - |1035 | -
|0  |10.1.255.255/32*   |TRAP | CoppSystemIpBcast  |0    | - | BcastReceive | -
|1027 | -
|0  |10.3.0.0/32*       |TRAP | CoppSystemIpBcast  |0    | - | BcastReceive | -
|1027 | -
|0  |10.3.0.1/32*       |TRAP | CoppSystemIpUcast  |0    | - | Receive | - |32766| -
|0  |10.3.255.1/32*     |ROUTE| Et18               |2700 |2 | 00:1f:5d:6b:00:01
| - |1038 | -
.......................................................
```

- **Related Commands**
  - The agent SandL3Unicast terminate command enables restarting the Layer 3 agent to ensure IPv4 routes are optimized.
  - The ip hardware fib optimize command enables IPv4 route scale.
  - The show platform arad ip route summary command shows hardware resource usage of IPv4 routes.
- This command shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.

```
switch(config)#show platform arad ip route
Tunnel Type: M(mpls), G(gre)
* - Routes in LEM

-------------------------------------------------------------------------------
|                              Routing Table                    |             |
|-----------------------------------------------------------------------------
|VRF|  Destination |     |                  |    |Acl  |                |ECMP
| FEC | Tunnel
|ID |    Subnet    | Cmd |    Destination    |VID |Label| MAC / CPU Code
|Index|Index|T Value

-------------------------------------------------------------------------------
|0  |0.0.0.0/8       |TRAP |CoppSystemL3DstMiss|0   | -   |ArpTrap          |  -
|1030 |   -
|0  |100.1.0.0/32    |TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive     |  -
|1032 |   -
|0  |100.1.0.0/32    |TRAP |CoppSystemIpUcast  |0   | -   |Receive          |  -
|32766|   -
|0  |100.1.255.255/32|TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive     |  -
|1032 |   -
|0  |200.1.255.255/32|TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive     |  -
|1032 |   -
|0  |200.1.0.0/16    |TRAP |CoppSystemL3DstMiss|1007| -   |ArpTrap          |  -
|1029 |   -
|0  |0.0.0.0/0       |TRAP |CoppSystemL3LpmOver|0   | -   |SlowReceive      |  -
|1024 |   -
|0  |4.4.4.0/24*     |ROUTE|Et10               |1007| -   |00:01:00:02:00:03|  -
|1033 |   -
|0  |10.20.30.0/24*  |ROUTE|Et9                |1006| -   |00:01:00:02:00:03|  -
|1027 |   -

switch(config)#
```

13.1.16.74 show platform barefoot bfrt

The **show platform barefoot bfrt** command displays information about the current BfRuntime server configuration.

Command Mode

EXEC

Command Syntax

show platform barefoot bfrt

Parameters

no parameter state of the system.

Example

The following output is for a system where the BfRuntime server has been configured.

```
(switch)#show platform barefoot bfrt
Namespace: management
FixedSystem:0.0.0.0:50052
```

13.1.16.75 show platform fap eedb ip-tunnel gre interface tunnel

The **`show platform fap eedb ip-tunnel gre interface tunnel`** command verifies the tunnel encapsulation programming for the tunnel interface.

Command Mode

EXEC

Command Syntax

`show platform fap eedb ip-tunnel gre interface tunnel` *number*

Parameter

number Specifies the tunnel interface number.

Example

These commands verify the tunnel encapsulation programming for the ***tunnel interface 10***.

```
switch#show platform fap eedb ip-tunnel gre interface tunnel 10

-------------------------------------------------------------------------------
|                                                 Jericho0                    |
|                                GRE Tunnel Egress Encapsulation DB

|
|-------------------------------------------------------------------------------|
| Bank/ | OutLIF | Next   | VSI  | Encap | TOS  | TTL | Source | Destination|
OamLIF| OutLIF | Drop|
| Offset|        | OutLIF | LSB  | Mode  |      |     | IP     | IP         | Set
| Profile|     |
|-------------------------------------------------------------------------------|
| 3/0   | 0x6000 | 0x4010 | 0    | 2     | 10   | 10  | 10.1.1.1 | 10.1.1.2 | No
| 0      | No  |

switch#show platform fap eedb ip-tunnel
-------------------------------------------------------------------------------
|                                                 Jericho0                    |
|                                    IP Tunnel Egress Encapsulation DB

|
|-------------------------------------------------------------------------------
| Bank/ | OutLIF | Next   | VSI | Encap| TOS | TTL | Src | Destination | OamLIF
| OutLIF  | Drop|
| Offset|        | OutLIF | LSB | Mode | Idx | Idx | Idx | IP          | Set    |
Profile |     |
|-------------------------------------------------------------------------------
| 3/0   | 0x6000 | 0x4010 | 0   | 2    | 9   | 0   | 0   | 10.1.1.2    | No     |
0       | No  |
```

13.1.16.76 show platform fap tcam summary

The **show platform fap tcam summary** command displays information about the TCAM bank that is allocated for GRE packet termination lookup.

Command Mode

EXEC

Command Syntax

show platform fap tcam summary

Example

This command verifies if the TCAM bank is allocated for GRE packet termination lookup.

```
switch# show platform fap tcam summary

Tcam Allocation (Jericho0)
Bank        Used By                Reserved By
----------- ---------------------- -----------
0           dbGreTunnel             -
```

13.1.16.77 show platform trident forwarding-table partition

The **show platform trident forwarding-table partition** command displays the size of the L2 MAC entry tables, L3 IP forwarding tables, and Longest Prefix Match (LPM) routes.

Command Mode

Privileged EXEC

Command Syntax

show platform trident forwarding-table partition

show platform trident forwarding-table partition flexible

Example

The **show platform trident forwarding-table partition** command shows the Trident forwarding table information.

```
switch(config)# show platform trident forwarding-table partition
L2 Table Size: 96k
L3 Host Table Size: 208k
LPM Table Size: 16k
switch(config)#
```

The **show platform trident forwarding-table partition flexible**shows the banks allocated for ALPM as well.

```
switch(config)# show platform trident forwarding-table partition
 flexible
------------------------------------------------
Minimum L2 entries             = 32768
Minimum L3 entries             = 16384
Maximum L2 entries             = 262144
Maximum L3 entries             = 262144
Maximum Exact Match entries    = 131072
```

```
L2 entries per bucket          = 4
L3 entries per bucket          = 4
Exact Match entries per bucket = 2
Maximum entries per bucket     = 4
Maximum shared buckets         = 65536
Maximum entries per bank       = 32768
Maximum shared banks           = 8
ALPM entries per bank          = 46080
ALPM                           = Enabled
--------------------
# UFT bank details #
--------------------
S - Shared UFT bank, D - Dedicated UFT bank
+-------------+------------+------+------------+--------------+
| Physical ID |  Feature   | Type | Logical ID | Hash Offset  |
+-------------+------------+------+------------+--------------+
|      0      |     L2     |  D   |     0      |     0x4      |
|      1      |     L2     |  D   |     1      |     0xe      |
|      2      |    ALPM    |  S   |    N/A     |      0       |
|      3      |    ALPM    |  S   |    N/A     |      0       |
|      4      |    ALPM    |  S   |    N/A     |      0       |
|      5      |    ALPM    |  S   |    N/A     |      0       |
|      6      |     L2     |  S   |     2      |     0xc      |
|      7      | ExactMatch |  S   |     0      |     0xc      |
|      8      | ExactMatch |  S   |     1      |     0xf      |
|      9      |     L3     |  S   |     2      |     0xc      |
|      10     |     L3     |  D   |     0      |     0x0      |
|      11     |     L3     |  D   |     1      |     0x8      |
+-------------+------------+------+------------+--------------+
```

13.1.16.78 show rib route ip

The **`show rib route ip`** command displays a list of IPv4 Routing Information Base (RIB) routes.

Command Mode

EXEC

Command Syntax

`show rib route ip` [vrf ***vrf_name***][**PREFIX**][**ROUTE TYPE**]

Parameters

- **vrf *vrf_name*** Displays RIB routes from the specified VRF.
- **PREFIX** dDisplays routes filtered by the specified IPv4 information. Options include:
  - ***ip_address*** Displays RIB routes filtered by the specified IPv4 address.
  - ***ip_subnet_mask*** Displays RIB routes filtered by the specified IPv4 address and subnet mask.
  - ***ip_prefix*** Displays RIB routes filtered by the specified IPv4 prefix.
- **ROUTE TYPE** Displays routes filtered by the specified route type. Options include:
  - **bgp** Displays RIB routes filtered by BGP.
  - **connected** Displays RIB routes filtered by connected routes.
  - **dynamicPolicy** Displays RIB routes filtered by dynamic policy routes.
  - **host** Displays RIB routes filtered by host routes.
  - **isis** Displays RIB routes filtered by ISIS routes.
  - **ospf** Displays RIB routes filtered by OSPF routes.
  - **ospf3** Displays RIB routes filtered by OSPF3 routes.
  - **reserved** Displays RIB routes filtered by reserved routes.

- **route-input** Displays RIB routes filtered by route-input routes.
- **static** Displays RIB routes filtered by static routes.
- **vrf** Displays routes in a VRF.
- **vrf-leak** Displays leaked routes in a VRF.

Examples

- This command displays IPv4 RIB static routes.

```
switch# show rib route ip static
VRF name: default, VRF ID: 0xfe, Protocol: static
Codes: C - Connected, S - Static, P - Route Input
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis
       > - Best Route, * - Unresolved Nexthop
       L - Part of a recursive route resolution loop
>S    10.80.0.0/12 [1/0]
         via 172.30.149.129 [0/1]
            via Management1, directly connected
>S    172.16.0.0/12 [1/0]
         via 172.30.149.129 [0/1]
            via Management1, directly connected switch#
```

- This command displays IPv4 RIB connected routes.

```
switch# show rib route ip connected
VRF name: default, VRF ID: 0xfe, Protocol: connected
Codes: C - Connected, S - Static, P - Route Input
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis
       > - Best Route, * - Unresolved Nexthop
       L - Part of a recursive route resolution loop
>C    10.1.0.0/24 [0/1]
         via 10.1.0.102, Ethernet1
>C    10.2.0.0/24 [0/1]
         via 10.2.0.102, Ethernet2
>C    10.3.0.0/24 [0/1]
         via 10.3.0.102, Ethernet3 switch#
```

- This command displays routes leaked through VRF leak agent.

```
switch# show rib route ip vrf VRF2 vrf-leak
VRF: VRF2, Protocol: vrf-leak
...
>VL    20.0.0.0/8 [1/0] source VRF: VRF1
          via 10.1.2.10 [0/0] type ipv4
             via 10.1.2.10, Ethernet1
```

13.1.16.79 show rib route fib policy excluded

The **show rib route fib policy excluded** command displays the RIB routes filtered by FIB policy. The **fib policy excluded** option displays the RIB routes that have been excluded from being programmed into FIB, by FIB policy.

Command Mode

EXEC

Command Syntax

show rib route [***ipv4*** | ***ipv6***] fib policy excluded

Example

The following example displays the RIB routes excluded by the FIB policy using the fib policy excluded option of the **show rib route** command.

```
switch# show rib route ipv6 fib policy excluded
switch# show rib route ip bgp fib policy excluded

VRF name: default, VRF ID: 0xfe, Protocol: bgp
Codes: C - Connected, S - Static, P - Route Input
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis
       > - Best Route, * - Unresolved Nexthop
       L - Part of a recursive route resolution loop
>B    10.1.0.0/24 [200/0]
         via 10.2.2.1 [115/20] type tunnel
            via 10.3.5.1, Ethernet1
         via 10.2.0.1 [115/20] type tunnel
            via 10.3.4.1, Ethernet2
            via 10.3.6.1, Ethernet3
>B    10.1.0.0/24 [200/0]
         via 10.2.2.1 [115/20] type tunnel
            via 10.3.5.1, Ethernet1
         via 10.2.0.1 [115/20] type tunnel
            via 10.3.4.1, Ethernet2
            via 10.3.6.1, Ethernet3
```

13.1.16.80 show rib route summary

The **show rib route summary** command displays information about the routes present in the Routing Information Base.

Command Mode

EXEC

Command Syntax

show rib route summary [**INFO_LEVEL**]

Parameters

- **no parameter** variable displays data in one table with the summary of all routes in the RIB for default VRF.
- **brief** keyword displays one table with the summary of all routes across all configured VRFs.
- **ip** keyword displays one table with the summary of all IPv4 in the RIB for default VRF.
- **ipv6** keyword displays one table with the summary of all IPv4 in the RIB for default VRF.
- **vrf *vrf_Name*** keyword displays one table with the summary of all routes in the Routing Information Base for the specified VRF.
- **vrf all** keyword displays one table with the summary of all routes in the Routing Information Base for each configured VRF.
- **INFO_LEVEL** amount of information that is displayed. Options include:
  - **Display Values**
    - **VRF** VRF RIB displayed.
    - **Route Source** Source for the route.
    - **Number of Routes** Number of routes for each source.

Examples

- The following displays data in one table with the summary of all routes in the RIB for default VRF.

```
switch> show rib route summary
VRF: default
Route Source         Number Of Routes
-------------------- ----------------
BGP                                1
Connected                          4
Dynamic policy                     0
IS-IS                              0
OSPF                               0
OSPFv3                             0
RIP                                0
Route input                        2
Static                             0
VRF leak                           0
```

- The following displays data in one table with the summary of all routes across all configured VRFs.

```
switch> show rib route summary brief
Route Source         Number Of Routes
-------------------- ----------------
BGP                                2
Connected                          8
Dynamic policy                     0
IS-IS                              0
OSPF                               0
OSPFv3                             0
RIP                                0
Route input                        4
Static                             0
VRF leak                           0
```

- The following displays data in one table with the summary of all IPv4 routes in the RIB for default VRF.

```
switch> show rib route summary ip
VRF: default
Route Source         Number Of Routes
-------------------- ----------------
BGP                                1
Connected                          4
Dynamic policy                     0
IS-IS                              0
OSPF                               0
OSPFv3                             0
RIP                                0
Route input                        2
Static                             0
VRF leak                           0
```

- The following displays data in one table with the summary of all IPv6 routes in the RIB for default VRF.

```
switch> show rib route summary ipv6
VRF: default
Route Source         Number Of Routes
-------------------- ----------------
BGP                                0
Connected                          0
Dynamic policy                     0
```

```
IS-IS                                0
OSPF                                 0
OSPFv3                               0
RIP                                  0
Route input                          0
Static                               0
VRF leak                             0
```

- The following displays data in one table with the summary of all routes in the RIB for the VRF named ***red***.

```
switch> show rib route summary vrf red
VRF: red
Route Source          Number Of Routes
--------------------- ----------------
BGP                                  1
Connected                            4
Dynamic policy                       0
IS-IS                                0
OSPF                                 0
OSPFv3                               0
RIP                                  0
Route input                          2
Static                               0
VRF leak                             0
```

- The following displays data in one table with the summary of all routes in the RIB for each configured VRF.

```
switch> show rib route summary vrf all
VRF: red
Route Source          Number Of Routes
--------------------- ----------------
BGP                                  1
Connected                            4
Dynamic policy                       0
IS-IS                                0
OSPF                                 0
OSPFv3                               0
RIP                                  0
Route input                          2
Static                               0
VRF leak                             0

VRF: default
Route Source          Number Of Routes
--------------------- ----------------
BGP                                  1
Connected                            4
Dynamic policy                       0
IS-IS                                0
OSPF                                 0
OSPFv3                               0
RIP                                  0
Route input                          2
Static                               0
VRF leak                             0
```

13.1.16.81 show routing-context vrf

The **show routing-context vrf** command displays the context-active VRF. The context-active VRF determines the default VRF that VRF-context aware commands use when displaying routing table data from a specified VRF.

Command Mode

EXEC

Command Syntax

show routing-context vrf

Related Commands

The cli vrf command specifies the context-active VRF.

Example

This command displays the context-active VRF.

```
switch> show routing-context vrf
Current VRF routing-context is PURPLE switch>
```

13.1.16.82 show tunnel fib static interface gre

The **show tunnel fib static interface gre** command displays the Forwarding Information Base (FIB) information for a static interface GRE tunnel.

Command Mode

EXEC

Command Syntax

show tunnel fib static interface gre *number*

Parameter

number Specifies the tunnel index number.

Example

This command display the interface tunnel configuration with GRE configured.

```
switch# show tunnel fib static interface gre 10

Type 'Static Interface', index 10, forwarding Primary
   via 10.6.1.2, 'Ethernet6/1'
      GRE, destination 10.1.1.2, source 10.1.1.1, ttl 10, tos 0xa
```

13.1.16.83 show vrf

The **show vrf** command displays the VRF name, RD, supported protocols, state and included interfaces for the specified VRF or for all VRFs on the switch.

Command Mode

EXEC

Command Syntax

`show vrf` [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE Specifies the VRF instance to display.

- ***no parameter*** Information is displayed for all VRFs.
- **vrf *vrf_name*** Information is displayed for the specified user-defined VRF.

Example

This command displays information for the VRF named ***purple***.

```
switch> show vrf purple
Vrf      RD          Protocols  State       Interfaces
-------- ----------- ---------- ----------- --------------
purple   64496:237   ipv4       no routing  Vlan42, Vlan43 switch>
```

13.1.16.84 tcp mss ceiling

The **`tcp mss ceiling`** command configures the Maximum Segment Size (MSS) limit in the TCP header on the configuration mode interface and enables TCP MSS clamping.

The **`no tcp mss ceiling`** and the **`default tcp mss ceiling`** commands remove any MSS ceiling limit previously configured on the interface.

Note: Configuring a TCP MSS ceiling on any Ethernet or tunnel interface enables TCP MSS clamping on the switch as a whole. Without hardware support, clamping routes all TCP SYN packets through software, even on interfaces where no TCP MSS ceiling has been configured. This significantly limits the number of TCP sessions the switch can establish per second, and can potentially cause packet loss if the CPU traffic exceeds control plane policy limits.

Command Mode

Interface-Ethernet Configuration

Subinterface-Ethernet Configuration

Interface-Port-channel Configuration

Subinterface-Port-channel Configuration

Interface-Tunnel Configuration

Interface-VLAN Configuration

Command Syntax

`tcp mss ceiling` {ipv4 ***segment size*** | ipv6 ***segment size***}{egress | ingress}

`no tcp mss ceiling`

`default tcp mss ceiling`

Parameters

- **ipv4** ***segment size*** The IPv4 segment size value in bytes. Values range from **`64`** to **`65515`**.
- **ipv6** ***segment size*** The IPv6 segment size value in bytes. Values range from **`64`** to **`65495`**. This option is not supported on Sand platform switches (Qumran-MX, Qumran-AX, Jericho, Jericho+).
- **egress** The TCP SYN packets that are forwarded from the interface to the network.
- **ingress** The TCP SYN packets that are received from the network to the interface. Not supported on Sand platform switches.

Guidelines

- On Sand platform switches (Qumran-MX, Qumran-AX, Jericho, Jericho+), this command works only for egress, and is supported only on IPv4 unicast packets entering the switch.
- Clamping can only be configured in one direction per interface and works only on egress on Sand platform switches.
- To configure ceilings for both IPv4 and IPv6 packets, both configurations must be included in a single command; re-issuing the command overwrites any previous settings.
- Clamping configuration has no effect on GRE transit packets.

Example

These commands configure ***interface ethernet 5*** as a routed port, then specify a maximum MSS ceiling value of ***1458*** bytes in TCP SYN packets exiting that port. This enables TCP MSS clamping on the switch.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no switchport
switch(config-if-Et5)# tcp mss ceiling ipv4 1458 egress
switch(config-if-Et5)#
```

13.1.16.85 tunnel

The **`tunnel`** command configures options for protocol-over-protocol tunneling. Because interface-tunnel configuration mode is not a group change mode, ***running-config*** is changed immediately after commands are executed. The **`exit`** command does not affect the configuration.

The **`no tunnel`** command deletes the specified tunnel configuration.

Command Mode

Interface-tunnel Configuration

Command Syntax

`tunnel` ***options***

`no tunnel` ***options***

Parameters

- ***options*** Specifies the various tunneling options as listed below.
  - **destination** Destination address of the tunnel.
  - **ipsec** Secures the tunnel with the IPsec address.
  - **key** Sets the tunnel key.
  - **mode** Tunnel encapsulation method.
  - **path-mtu-discovery** Enables the Path MTU discovery on tunnel.
  - **source** Source of the tunnel packets.
  - **tos** Sets the IP type of service value.
  - **ttl** Sets time to live value.
  - **underlay** Tunnel underlay.

Example

These commands place the switch in interface-tunnel configuration mode for ***interface Tunnel 10*** and with GRE tunnel configured on the interfaces specified.

```
switch(config)# ip routing
switch(config)# interface Tunnel 10
switch(config-if-Tu10)# tunnel mode gre
switch(config-if-Tu10)# ip address 192.168.1.1/24
switch(config-if-Tu10)# tunnel source 10.1.1.1
switch(config-if-Tu10)# tunnel destination 10.1.1.2
switch(config-if-Tu10)# tunnel path-mtu-discovery
switch(config-if-Tu10)# tunnel tos 10
switch(config-if-Tu10)# tunnel ttl 10
```

13.1.16.86 vrf (Interface mode)

The **`vrf`** command adds the configuration mode interface to the specified VRF. You must create the VRF first, using the vrf instance command.

The **`no vrf`** and **`default vrf`** commands remove the configuration mode interface from the specified VRF by deleting the corresponding **`vrf`** command from ***running-config***.

All forms of the **`vrf`** command remove all IP addresses associated with the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`vrf` [***vrf_name***]

`no vrf` [***vrf_name***]

`default vrf` [***vrf_name***]

Parameters

vrf_name Name of configured VRF.

Examples

- These commands add the configuration mode interface (***vlan 20***) to the VRF named ***purple***.

```
switch(config)# interface vlan 20
switch(config-if-Vl20)# vrf purple
switch(config-if-Vl20)#
```

- These commands remove the configuration mode interface from VRF ***purple***.

```
switch(config)#  interface vlan 20
switch(config-if-Vl20)# no vrf purple
switch(config-if-Vl20)#
```

13.1.16.87 vrf instance

The **vrf instance** command places the switch in VRF configuration mode for the specified VRF. If the named VRF does not exist, this command creates it. The number of user-defined VRFs supported varies by platform.

To add an interface to the VRF once it is created, use the vrf (Interface mode) command.

The **no vrf instance** and **default vrf instance** commands delete the specified VRF instance by removing the corresponding **vrf instance** command from ***running-config***. This also removes all IP addresses associated with interfaces that belong to the deleted VRF.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

vrf instance [***vrf_name***]

no vrf instance [***vrf_name***]

default vrf instance [***vrf_name***]

Parameters

vrf_name Name of VRF being created, deleted or configured. The names ***main*** and ***default*** are reserved.

Example

This command creates a VRF named ***purple*** and places the switch in VRF configuration mode for that VRF.

```
switch(config)# vrf instance purple
switch(config-vrf-purple)#
```

13.2 IPv6

Arista switches support Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) for routing packets across network boundaries. This section describes Arista's implementation of IPv6 and includes these topics:

- Introduction
- IPv6 Description
- Configuring IPv6
- IPv6 Commands

13.2.1 Introduction

Routing transmits network layer data packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range.

Connected subnets have IP address ranges that do not overlap. A router is a network device connecting multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address.

IPv4 and IPv6 are Internet layer protocols that define packet-switched inter-networking, including source-to-destination datagram transmission across multiple networks. The switch supports IP version 4 (IPv4) and IP version 6 (IPv6).

IPv6 is described by ***RFC 2460: Internet Protocol, Version 6 (IPv6) Specification***. ***RFC 2463*** describes ICMPv6 for IPv6. ICMPv6 is a core protocol of the Internet Protocol suite.

13.2.2 IPv6 Description

Internet Protocol version 6 (IPv6) is a communications protocol used for relaying network packets across a set of connected networks using the Internet Protocol suite. Each network device is assigned a 128 bit IP address that identifies its network location.

IPv6 specifies a packet format that minimizes router processing of packet headers. Since the IPv4 and IPv6 packet headers differ significantly, the protocols are not interoperable. Many transport and application-layer protocols require little or no change to operate over IPv6.

- IPv6 Address Format
- IPv6 DHCP Snooping
- Neighbor Discovery Protocol

13.2.2.1 IPv6 Address Format

IPv6 addresses have 128 bits, represented by eight 16-bit hexadecimal numbers separated by colons. IPv6 addresses are abbreviated as follows:

- Leading zeros in each 16-bit number may be omitted.
- One set of consecutive 16-bit numbers that equal zero may be replaced by a double colon.

Example

The following three IPv6 hexadecimal number representations refer to the same address:

```
d28e:0000:0000:0000:0234:812f:61ed:4419
d28e:0:0:0:234:812f:61ed:4419
d28e::234:812f:61ed:4419
```

IPv6 addresses typically denote a 64-bit network prefix and a 64-bit host address.

Unicast and Anycast Addressing

Unicast addressing defines a one-to-one association between the destination address and a network endpoint. Each destination address uniquely identifies a single receiver endpoint. Anycast addressing defines a one-to-one-of-many association: packets to a single member of a group of potential receivers identified by the same destination address.

Unicast and anycast addresses are typically composed as follows:

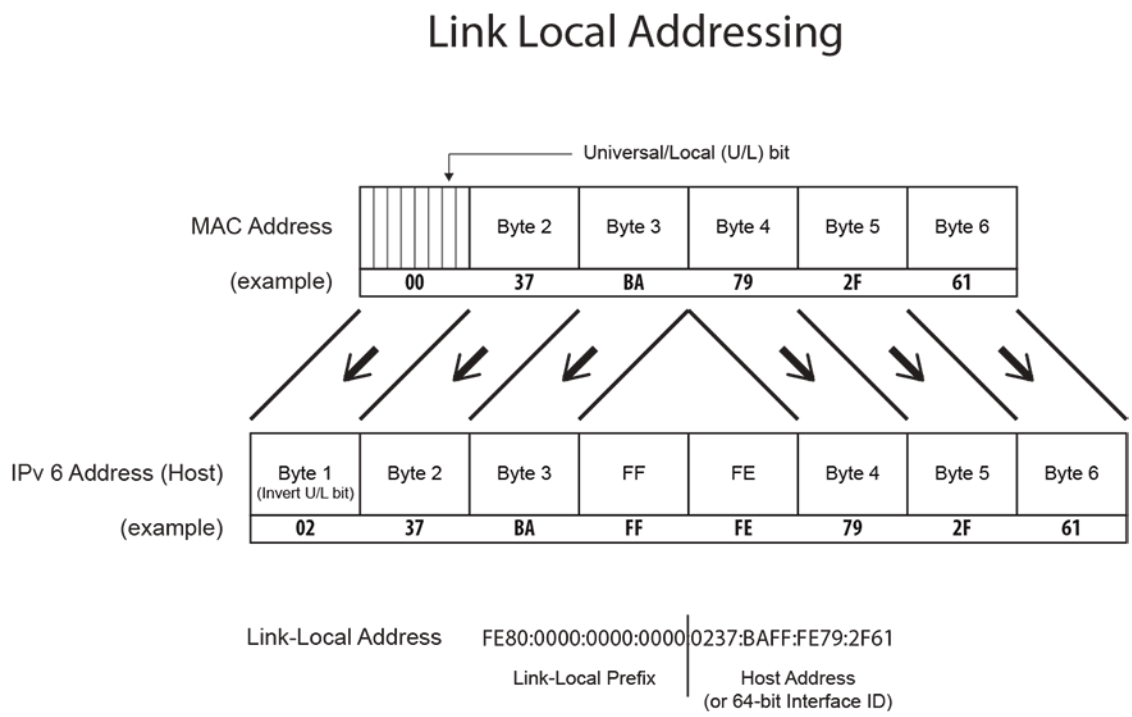
- A 64-bit network prefix that identifies the network segment.
- A 64-bit interface identifier that is based on interface MAC address.

The format of a network address identifies the scope of the address:

- Global address: valid in all networks and connect with other addresses with global scope anywhere or to addresses with link-local scope on the directly attached network.
- Link-local address: scope extends only to the link to which the interface is directly connected. Link-local addresses are not routable off the link.

Link-local addresses are created by the switch and are not configurable. The following figure depicts the switch's link local address derivation method.

Figure 13-1: Link Local Address Derivation



Multicast Addressing

Multicast addressing defines a one-to-many association: packets are simultaneously routed from a single sender to multiple endpoints in a single transmission. The network replicates packets as required by network links that contain a recipient endpoint. One multicast address is assigned to an interface for each multicast group to which the interface belongs.

A solicited-node multicast address is an IPv6 multicast address whose scope extends only to the link to which the interface is directly connected. All IPv6 hosts have at least one such address per interface. Solicited-node multicast addresses are used by the Neighbor Discovery Protocol to obtain Layer 2 link-layer addresses of other nodes.

13.2.2.2 IPv6 DHCP Snooping

Dynamic Host Configuration Protocol (DHCP) snooping is a Layer 2 feature that is configured on LAN switches. The Arista EOS switch supports Option-37 insertion that allows relay agents to provide remote-id information in DHCP request packets. DHCP servers use this information to determine the originating port of DHCP requests and associate a corresponding IP address to that port. DHCP servers use port information to track host location and IP address usage by authorized physical ports.

DHCP snooping uses the information option (Option-37) to include the switch MAC address (router-id) along with the physical interface name and VLAN number (remote-id) in DHCP packets. After adding the information to the packet, the DHCP relay agent forwards the packet to the DHCP server as specified by the DHCP protocol.

13.2.2.3 Neighbor Discovery Protocol

The Neighbor Discovery Protocol (***RFC 4861***) operates with IPv6 to facilitate the following tasks for nodes within a specified prefix space:

- autoconfiguring a node's IPv6 address
- sensing other nodes on the link
- discovering the link-local addresses of other nodes on the link
- detecting duplicate addresses
- discovering available routers
- discovering DNS servers
- discovering the link's address prefix
- maintaining path reachability data to other active neighbor nodes

The Neighbor Discovery Protocol protocol defines five different ICMPv6 packet types:

- Router Solicitation
- Router Advertisement
- Neighbor Solicitation
- Neighbor Advertisement
- Redirect

13.2.3 Configuring IPv6

These sections describe IPv6 configuration tasks:

- Configuring IPv6 on the Switch
- Configuring IPv6 on an Interface
- Configuring IPv6 DHCP Snooping
- Viewing IPv6 Network Components
- DHCP Relay Agent for IPv6
- TCP MSS Clamping for IPv6

13.2.3.1 Configuring IPv6 on the Switch

13.2.3.1.1 Enabling IPv6 Unicast Routing on the Switch

The ipv6 unicast-routing command enables the forwarding of IPv6 unicast packets. When routing is enabled, the switch attempts to deliver inbound packets to destination addresses by forwarding them to interfaces or next hop addresses specified by the IPv6 routing table.

Example

This command enables IPv6 unicast-routing.

```
switch(config)# ipv6 unicast-routing
switch(config)#
```

13.2.3.1.2 Configuring Default and Static IPv6 Routes

The ipv6 route command creates an IPv6 static route. The destination is a IPv6 prefix; the source is an IPv6 address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

By default, the administrative distance assigned to static routes is ***1***. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of ***200*** is overridden by OSPF intra-area routes, which have a default distance of ***110***.

Example

This command creates an IPv6 static route.

```
switch(config)# ipv6 route 10:23:31:00:01:32:93/24 vlan 300
switch(config)#
```

The default route denotes the packet forwarding rule that takes effect when no other route is configured for a specified IPv6 address. All packets with destinations that are not established in the routing table are sent to the destination specified by the default route.

The IPv6 default route source is ***::/0***. The default route destination is referred to as the default gateway.

Example

This command creates a default route and establishes ***fd7a:629f:52a4:fe61::2*** as the default gateway address.

```
switch(config)# ipv6 route ::/0 fd7a:629f:52a4:fe61::2
switch(config)#
```

13.2.3.1.3 IPv6 ECMP

Multiple routes that are configured to the same destination with the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic across all ECMP route paths equally. All ECMP paths are assigned the same tag value; commands that change the tag value of any ECMP path change the tag value of all paths in the ECMP.

Resilient ECMP is available for IPv6 routes. Equal Cost Multipath Routing (ECMP) and Load Sharing describes resilient ECMP. The ipv6 hardware fib ecmp resilience command implements IPv6 resilient ECMP.

Example

This command implements IPv6 resilient ECMP by configuring a hardware ECMP table space of 15 entries for IPv6 address ***2001:db8:0::/64***. A maximum of five nexthop addresses can be specified for the address. When the table contains five addresses, each appears in the table three times. When the table contains fewer than five addresses, each is duplicated until the 15 table entries are filled.

```
switch(config)# ipv6 hardware fib ecmp resilience 2001:db8:0::/64 capacity 5
redundancy 3
switch(config)#
```

13.2.3.2 Configuring IPv6 on an Interface

13.2.3.2.1 Enabling IPv6 on an Interface

The ipv6 enable command enables IPv6 on the configuration mode interface if it does not have a configured IPv6 address. It also configures the interface with an IPv6 address.

The **no** ipv6 enable command disables IPv6 on a configuration mode interface not configured with an IPv6 address. Interfaces configured with an IPv6 address are not disabled by this command.

Example

This command enables IPv6 on ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-vl200)# ipv6 enable
switch(config-vl200)#
```

13.2.3.2.2 Assigning an IPv6 Address to an Interface

The ipv6 address command enables IPv6 on the configuration mode interface, assigns a global IPv6 address to the interface, and defines the prefix length. This command is supported on routable interfaces. Multiple global IPv6 addresses can be assigned to an interface.

Example

These commands configure an IPv6 address with subnet mask for ***vlan 200***:

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 address 10:23:31::1:32:93/64
switch(config-if-Vl200)#
```

13.2.3.2.3 IPv6 Neighbor Discovery

The IPv6 Neighbor Discovery protocol defines a method for nodes to perform the following network maintenance tasks:

- determine Layer 2 addresses for neighbors known to reside on attached links.
- detect changed Layer 2 addresses.
- purge invalid values from the neighbor cache table.
- (hosts) find neighboring routers to forward packets.
- track neighbor reachability status.

IPv6 Neighbor Discovery is defined by ***RFC 2461***. IPv6 Stateless Address Autoconfiguration is described by ***RFC 2462***.

The following sections describe Neighbor Discovery configuration tasks.

13.2.3.2.3.1Reachable Time

The ipv6 nd reachable-time command specifies the time period that the switch includes in the reachable time field of Router Advertisements (RAs) sent from the configuration mode interface. The reachable time defines the period that a remote IPv6 node is considered reachable after a reachability confirmation event.

Example

These commands configure the entry of ***25000*** (25 seconds) in the reachable time field of RAs sent from ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd reachable-time 25000
switch(config-if-Vl200)# show active
   interface Vlan200
   ipv6 address fd7a:4321::1/64
   ipv6 nd reachable-time 25000
switch(config-if-Vl200)#
```

13.2.3.2.3.2Router Advertisement Interval

The ipv6 nd ra interval command configures the interval between IPv6 RA transmissions from the configuration mode interface.

Example

These commands configure a RA transmission interval of ***60*** seconds on ***interface vlan 200***, then displays the interface status.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra interval 60
```

```
switch(config-if-Vl200)# show active
interface Vlan200
   ipv6 nd ra interval 60
switch(config-if-Vl200)#
```

13.2.3.2.3.3Router Lifetime

The ipv6 nd ra lifetime command specifies the value that the switch places in the ***router lifetime*** field of IPv6 RAs sent from the configuration mode interface.

If the value is set to ***0***, IPv6 peers connected to the specified interface will remove the switch from their lists of default routers. Values greater than ***0*** indicate the time in seconds that peers should keep the router on their default router lists without receiving further RAs from the switch. Unless the value is ***0***, the router lifetime value should be equal to or greater than the interval between unsolicited RAs sent on the interface.

Example

This command configures the switch to enter ***2700*** in the router lifetime field of RAs transmitted from ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra lifetime 2700
switch(config-if-Vl200)# show active
interface Vlan200
   ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#
```

13.2.3.2.3.4Router Advertisement Prefix

The ipv6 nd prefix command configures neighbor discovery router advertisement prefix inclusion for RAs sent from the configuration mode interface.

By default, all prefixes configured as IPv6 addresses are advertised in the interface's RAs. The **ipv6 nd prefix** command with the **no-advertise** option prevents advertising of the specified prefix without affecting the advertising of other prefixes specified as IPv6 addresses. When an interface configuration includes at least one **ipv6 nd prefix** command that enables prefix advertising, RAs advertise only prefixes specified through **ipv6 nd prefix** commands.

Commands enabling prefix advertising also specify the advertised valid and preferred lifetime periods. Default periods are ***2,592,000*** (valid) and ***604,800*** (preferred) seconds.

Example

These commands enable neighbor discovery advertising for IPv6 address ***3012:D678::/64***, specifying a valid lifetime of ***1,296,000*** seconds and the default preferred lifetime.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd prefix 3012:D678::/64 1296000
switch(config-if-Vl200)#
```

13.2.3.2.3.5Router Advertisement Suppression

The ipv6 nd ra disabled command suppress IPv6 RA transmissions on the configuration mode interface. By default, only unsolicited RAs that are transmitted periodically are suppressed. The **all** option configures the switch to suppress all RAs, including those responding to a router solicitation.

Example

This command suppresses all RAs on ***interface vlan 200***.

```
switch(config)# interface vlan 200
```

```
switch(config-vl200)# ipv6 nd ra disabled all
switch(config-vl200)#
```

13.2.3.2.3.6Router Advertisement MTU Suppression

The ipv6 nd ra mtu suppress command suppresses the router advertisement MTU option on the configuration mode interface. The MTU option causes an identical MTU value to be advertised by all nodes on a link. By default, the router advertisement MTU option is not suppressed.

Example

This command suppresses the MTU option on ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-vl200)# ipv6 nd ra mtu suppress
switch(config-vl200)#
```

13.2.3.2.3.7Router Advertisement Flag Configuration

The following commands set the specified configuration flag in IPv6 RAs transmitted from the configuration mode interface:

- The ipv6 nd managed-config-flag command sets the **managed address configuration** flag. This bit instructs hosts to use stateful address autoconfiguration.
- The ipv6 nd other-config-flag command sets the **other stateful configuration** flag. This bit indicates availability of autoconfiguration information, other than addresses. Hosts should use stateful autoconfiguration when available. The setting of this flag has no effect if the **managed address configuration** flag is set.
- These commands configure the switch to set the **managed address configuration** flag in advertisements sent from ***interface vlan 200***.

  ```
  switch(config)# interface vlan 200
  switch(config-if-Vl200)# ipv6 nd managed-config-flag
  switch(config-if-Vl200)#
  ```

- These commands configure the switch to set the **other stateful configuration** flag in advertisements sent from ***interface vlan 200***.

  ```
  switch(config)# interface vlan 200
  switch(config-if-Vl200)# ipv6 nd other-config-flag
  switch(config-if-Vl200)#
  ```

13.2.3.2.4 IPv6 Router Preference

The IPv6 Router Preference protocol supports an extension to RA messages for communicating default router preferences and more specific routes from routers to hosts. This provides assistance to hosts when selecting a router. RFC 4191 describes the IPv6 Router Preference Protocol.

The ipv6 nd router-preference command specifies the value that the switch enters in the Default Router Preference (DRP) field of RAs that it sends from the configuration mode interface. The default field entry value is **medium**.

Example

This command configures the switch as a medium preference router on RAs sent from ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd router-preference medium
switch(config-if-Vl200)#
```

13.2.3.2.5 uRPF Configuration

Unicast Reverse Path Forwarding (uRPF) verifies the accessibility of source IP addresses in packets that the switch forwards. Unicast Reverse Path Forwarding (uRPF) describes uRPF. uRPF is enabled for IPv6 packets entering the configuration mode interface through the ipv6 verify command.

uRPF defines two operational modes: strict mode and loose mode.

- **Strict mode:** uRPF verifies that a packet is received on the interface that its routing table entry specifies for its return packet.
- **Loose mode:** uRPF validation does not consider the inbound packet's ingress interface only that there is a valid return path.

Example

This command enables uRPF strict mode on ***interface vlan 100***. If a default route is configured on the interface, all inbound packets will pass the uRPF check as valid.

```
switch(config)# interface vlan 100
switch(config-if-Vl100)# ipv6 verify unicast source reachable-via rx allow-
default
switch(config-if-Vl100)# show active
 interface Vlan100
   ipv6 verify unicast source reachable-via rx allow-default
switch(config-if-Vl100)#
```

13.2.3.3 Configuring IPv6 DHCP Snooping

13.2.3.3.1 Enabling IPv6 DHCP Snooping on the Switch

The ipv6 dhcp snooping command enables DHCP snooping globally on the switch. DHCP snooping is a Layer 2 feature that can be configured on LAN switches. The Arista switch supports Option-37 insertion that allows relay agents to provide remote-ID information in DHCP request packets.

Note: DHCPv6 VLAN classification and DHCPv4 VLAN classification share same hardware resource.

Examples

- The following configuration enables IPv6 DHCP snooping feature at the global level.

```
switch(config)# ipv6 dhcp snooping
switch(config)# ipv6 dhcp snooping remote-id option
switch(config)# ipv6 dhcp snooping vlan <vlan|vlan-range>
```

- The following command display IPv6 DHCP snooping state.

```
switch(config)# ipv6 dhcp snooping
switch(config)# show ipv6 dhcp snooping
DHCPv6 Snooping is enabled
DHCPv6 Snooping is operational
DHCPv6 Snooping is configured on following VLANs:
  2789-2790
DHCPv6 Snooping is operational on following VLANs:
  2789
Insertion of Option-37 is enabled
```

13.2.3.4 Viewing IPv6 Network Components

13.2.3.4.1 Displaying RIB Route Information

Use the show rib route ipv6 command view the IPv6 Routing Information Base (RIB) information.

Example

This command displays IPv6 RIB BGP routes.

```
switch# show rib route ipv6 bgp
VRF name: default, VRF ID: 0xfe, Protocol: bgp
Codes: C - Connected, S - Static, P - Route Input
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis
       > - Best Route, * - Unresolved Nexthop
       L - Part of a recursive route resolution loop
 B    2001:10:1::/64 [200/42]
         via 2001:10:1::100 [0/1]
            via Ethernet1, directly connected
>B    2001:10:100::/64 [200/200]
         via 2001:10:1::100 [0/1]
            via Ethernet1, directly connected
>B    2001:10:100:1::/64 [200/0]
         via 2001:10:1::100 [0/1]
            via Ethernet1, directly connected
>B    2001:10:100:2::/64 [200/42]
         via 2001:10:1::100 [0/1]
            via Ethernet1, directly connected switch#
```

13.2.3.4.2 Displaying the FIB and Routing Table

The show ipv6 route command displays routing table entries that are in the Forwarding Information Base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

Example

This command displays a route table entry for a specific IPv6 route.

```
switch> show ipv6 route fd7a:3418:52a4:fe18::/64
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:3418:52a4:fe18::/64 [10/20]
     via f180::21c:73ff:fe00:1319, Vlan3601
     via f180::21c:73ff:fe00:1319, Vlan3602
     via f180::21c:73ff:fe00:1319, Vlan3608
     via f180::21c:73ff:fe0f:6a80, Vlan3610
     via f180::21c:73ff:fe00:1319, Vlan3611 switch>
```

13.2.3.4.3 Displaying the Route Age

The show ipv6 route age command displays the IPv6 route age to the specified IPv6 address or prefix.

Example

This command displays the route age for the specified prefix.

```
switch> show ipv6 route 2001::3:0/11 age
```

```
IPv6 Routing Table - 74 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

C 2001::3:0/11 age 00:02:34 switch>
```

13.2.3.4.4 Displaying Host Routes

The show ipv6 route host command displays all host routes in the IPv6 host forwarding table. Host routes are those whose destination prefix is the entire address (prefix = ***/128***). Each displayed host route is labeled with its purpose:

- **F** static routes from the FIB.
- **R** routes defined because the IP address is an interface address.
- **A** routes to any neighboring host for which the switch has an ARP entry.

Example

This command displays all IPv6 host routes in the host forwarding table.

```
switch# show ipv6 route host
R - receive F - FIB, A - attached

 F  ::1 to cpu
 A  fee7:48a2:0c11:1900:400::1 on Vlan102
 R  fee7:48a2:0c11:1900:400::2 to cpu
 F  fee7:48a2:0c11:1a00::b via fe80::21c:73ff:fe0b:a80e on Vlan3902
 R  fee7:48a2:0c11:1a00::17 to cpu
 F  fee7:48a2:0c11:1a00::20 via fe80::21c:73ff:fe0b:33e on Vlan3913
 F  fee7:48a2:0c11:1a00::22 via fe80::21c:73ff:fe01:5fe1 on Vlan3908
                            via fe80::21c:73ff:fe01:5fe1 on Vlan3902 switch#
```

13.2.3.4.5 Displaying Route Summaries

The show ipv6 route summary command displays the current number of routes of the IPv6 routing table in summary format.

Example

This command displays the route source and the corresponding number of routes in the IPv6 routing table.

```
switch> show ipv6 route summary
Route Source    Number Of Routes
----------------- ----------------
   connected                    2
   static                       0
   ospf                         5
   bgp                          7
   isis                         0
   internal                     1
   attached                     0
   aggregate                    2

   Total Routes                 17 switch>
```

13.2.3.5 DHCP Relay Agent for IPv6

13.2.3.5.1 Configuring IPv6 DHCP Relay

13.2.3.5.1.1 Configuring the IPv6 DHCP Relay Agent (Global)

The ipv6 dhcp relay always-on command enables the switch DHCP relay agent globally regardless of the DHCP relay agent status ond any interface. The DHCP relay agent is enabled by default if at least one routable interface is configured with an ipv6 dhcp relay destination statement.

Example

This command enables the DHCP relay agent.

```
switch(config)# ipv6 dhcp relay always-on
switch(config)#
```

13.2.3.5.1.2 Configuring DHCP for IPv6 Relay Agent

The ipv6 dhcp relay destination command enables the DHCPv6 relay agent function and specifies the client message destination address on an interface.

Example

This command enables the DHCPv6 relay agent function and sets the client message destination address to ***2001:0db8:0:1::1*** on ***interface ethernet 4***.

```
switch(config)# interface ethernet 4
switch(config-if-Et4)# ipv6 dhcp relay destination 2001:0db8:0:1::1
switch(config-if-Et4)
```

13.2.3.5.1.3 Configuring the Client Link Layer Address for the IPv6 DHCP Relay Agent

The ipv6 dhcp relay option link-layer address command enables the DHCPv6 relay agent to configure the client link layer address option to solicit and request messages. In other words, the command enables the link layer address option (79) in the global configuration mode. The **no** ipv6 dhcp relay option link-layer address command disables the link layer address option (79) in the global configuration mode.

Example

This command enables the insertion of link layer address option (79) in the global configuration mode.

```
switch(config)# ipv6 dhcp relay option link-layer address
```

13.2.3.5.1.4 Clearing IPv6 DHCP Relay Counters

The clear ipv6 dhcp relay counters command resets the DHCP relay counters. The configuration mode determines which counters are reset:

- **Global configuration:** the command clears the counters for the switch and for all interfaces.
- **Interface configuration:** the command clears the counter for the configuration mode interface.

Examples

- These commands clear all DHCP relay counters on the switch.

```
switch(config-if-Et4)# exit
switch(config)# clear ipv6 dhcp relay counters
switch(config)#
```

- These commands clear the DHCP relay counters for ***interface ethernet 4***.

```
switch(config)# interface ethernet 4
switch(config-if-Et4)# clear ipv6 dhcp relay counters
switch(config)#
```

13.2.3.5.2 Viewing IPv6 DHCP Relay Information

13.2.3.5.2.1IPv6 DHCP Status

The **show ip dhcp relay** command displays the status of DHCP relay agent parameters on the switch and each interface where at least one feature parameter is listed. The command displays the status for both global and interface configurations.

Example

This command displays the DHCP Agent Relay parameter status.

```
switch(config)# interface ethernet 1/2
switch(config-if-Et1/2)# show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCPv6 Relay Link-layer Address Option (79) is disabled
DHCP Smart Relay is disabled
Interface: Ethernet1/2
  DHCP Smart Relay is disabled
  DHCP servers: 1::1
                2001:db8:0:1::1
switch(config-if-Et1/2)#
```

13.2.3.5.2.2IPv6 DHCP Relay Counters

The show ipv6 dhcp relay counters command displays the number of DHCP packets received, forwarded, or dropped on the switch and on all interfaces enabled as DHCP relay agents.

Example

This command displays the IP DHCP relay counter table.

```
switch> show ipv6 dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
Ethernet4 |  207  148    0 | 4 days, 19:54:24 ago switch>
```

13.2.3.6 TCP MSS Clamping for IPv6

TCP MSS clamping limits the value of the Maximum Segment Size (MSS) in the TCP header of TCP SYN packets transiting a specified Ethernet or tunnel interface. Setting the MSS ceiling can avoid IP fragmentation in tunnel scenarios by ensuring that the MSS is low enough to account for the extra overhead of GRE and tunnel outer IP headers. TCP MSS clamping can be used when connecting via GRE to cloud providers that require asymmetric routing.

When MSS clamping is configured on an interface, if the TCP MSS value in a SYN packet transiting that interface exceeds the configured ceiling limit it will be overwritten with the configured limit and the TCP checksum will be recomputed and updated.

TCP MSS clamping is handled by default in the software data path, but the process can be supported through hardware configuration to minimize possible packet loss and a reduction in the number of TCP sessions which the switch can establish per second.

13.2.3.6.1 Configuring the TCP MSS Ceiling on an IPv6 Interface

The TCP MSS ceiling limit is set on an interface using command **`tcp mss ceiling ipv6`**. This also enables TCP MSS clamping on the switch.



Note: Clamping routes all TCP SYN packets through software, even on interfaces where no TCP MSS ceiling has been configured without any hardware support. This significantly limits the number of TCP sessions the switch can establish per second, and can potentially cause packet loss if the CPU traffic exceeds control plane policy limits.

On platform switches ***DCS-7280R3, DCS-7500R3 Line Cards and DCS-7800R3 Line Cards***, the following limitations apply:

- This command works only on egress.
- TCP MSS ceiling is supported on unicast packets entering the switch; the configuration has no effect on GRE transit packets.
- It is not supported on L2 (switchport) interfaces.
- This is not supported on VXLAN, Loopback or management interfaces.
- This is only supported on unicast packets entering the switch. The configuration has no effect on GRE transit packets or GRE decap, even if the egress interface has a TCP MSS ceiling configured.
- This cannot co-exist with Policy Based Routing (PBR) on switches running releases ***EOS-4.21.5F*** or older.
- Only hardware TCP MSS clamping is supported from release ***EOS-4.26.1F***.

Example

These commands configure ***interface ethernet 26*** as a routed port, then specify a maximum MSS ceiling value of ***1436*** bytes for TCP SYN packets exiting that port.

```
switch(config)# interface ethernet 26
switch(config-if-Et5)# no switchport
switch(config-if-Et5)# tcp mss ceiling ipv6 1436 egress
switch(config-if-Et5)#
```

13.2.4 IPv6 Commands

Global Configuration Commands

- ipv6 dhcp relay always-on
- ipv6 dhcp relay option link-layer address
- ipv6 hardware fib aggregate-address
- ipv6 hardware fib ecmp resilience
- ipv6 hardware fib nexthop-index
- ipv6 neighbor
- ipv6 neighbor cache persistent
- ipv6 route
- ipv6 unicast-routing

Interface Configuration Commands

- ipv6 address
- ipv6 dhcp relay destination
- ipv6 dhcp snooping
- ipv6 enable
- ipv6 nd managed-config-flag
- ipv6 nd ns-interval
- ipv6 nd other-config-flag
- ipv6 nd prefix
- ipv6 nd ra dns-server
- ipv6 nd ra dns-servers lifetime
- ipv6 nd ra dns-suffix
- ipv6 nd ra dns-suffixes lifetime
- ipv6 nd ra hop-limit
- ipv6 nd ra interval
- ipv6 nd ra lifetime
- ipv6 nd ra mtu suppress
- ipv6 nd ra disabled
- ipv6 nd reachable-time
- ipv6 nd router-preference
- ipv6 verify
- pim ipv6 sparse-mode

Privileged EXEC Commands

- clear ipv6 dhcp relay counters
- clear ipv6 dhcp snooping counters
- clear ipv6 neighbors

EXEC Commands

- show ipv6 dhcp relay counters
- show ipv6 dhcp snooping
- show ipv6 dhcp snooping counters
- show ipv6 dhcp snooping hardware
- show ipv6 hardware fib aggregate-address
- show ipv6 interface
- show ipv6 nd ra internal state
- show ipv6 neighbors
- show ipv6 route
- show ipv6 route age
- show ipv6 route host
- show ipv6 route interface
- show ipv6 route match tag
- show ipv6 route summary
- show platform fap mroute ipv6
- show rib route ipv6

13.2.4.1 clear ipv6 dhcp relay counters

The **clear ipv6 dhcp relay counters** command resets the DHCP relay counters. When no port is specified, the command clears the counters for the switch and for all interfaces. Otherwise, the command clears the counter for the specified interface.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 dhcp relay counters [**PORT**]

Parameters

PORT Interface through which neighbor is accessed. Options include:

- ***no parameter*** all dynamic entries are removed.
- **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **interface loopback *l_num*** Loopback interface specified by ***l_num***.
- **interface port-channel *p_num*** Port-channel interface specified by ***p_num p_num***.
- **interface vlan *v_num*** VLAN interface specified by ***v_num***.

Example

These commands clear the DHCP relay counters for ***interface ethernet 4*** and shows the counters before and after the **clear** command.

```
switch(config)# show ipv6 dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
Ethernet4 |  207  148    0 | 4 days, 19:54:24 ago


switch(config)# interface ethernet 4
switch(config-if-Et4)# clear ipv6 dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  380  380    0 | 4 days, 21:19:17 ago
 All Resp |  281  281    0 |
          |                |
Ethernet4 |    0    0    0 |4 days, 21:18:30 ago
These commands clear all DHCP relay counters on the switch.
switch(config-if-Et4)# exit

switch(config)# clear ipv6 dhcp relay counters

switch(config)# show ipv6 dhcp relay counters
          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop | Last Cleared
----------|----- ---- -----|-------------
  All Req |    0    0    0 |  0:00:03 ago
 All Resp |    0    0    0 |
          |                |
Ethernet4 |    0    0    0 | 0:00:03 ago
switch(config)#
```

13.2.4.2 clear ipv6 dhcp snooping counters

The **clear ipv6 dhcp snooping counters** command resets the DHCP snooping packet counters.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 dhcp snooping counters [**COUNTER_TYPE**]

Parameters

- **COUNTER_TYPE** The type of counter that the command resets.
- ***no parameter*** command clears the counters for each VLAN.
- **debug** command clears aggregate counters and drop cause counters.

Examples

- This command clears the number of DHCP packets sent and received on each VLAN.

```
switch# clear ipv6 dhcp snooping counters
switch# show ipv6 dhcp snooping counters

     | Dhcpv6 Request Pkts | Dhcpv6 Reply Pkts |
Vlan |  Rcvd   Fwdd   Drop |  Rcvd  Fwdd  Drop | Last Cleared
-----|------ ------ -------|------ ----- ------|-------------
2789 |     1      1      0 |     1     1     0 |  0:03:09 ago
```

- This command clears the number of DHCP packets sent on the switch.

```
switch# clear ipv6 dhcp snooping counters debug
switch# show ipv6 dhcp snooping counters debug

Counter                         Snooping to Relay Relay to Snooping
------------------------------ ----------------- -----------------
Received                                        1                 1
Forwarded                                       1                 1
Dropped - Invalid VlanId                        0                 0
Dropped - Parse error                           0                 0
Dropped - Invalid Dhcp Optype                   0                 0
Dropped - Invalid Remote-ID Option              0                 0
Dropped - Snooping disabled                     0                 0

Last Cleared:  0:04:29 ago
```

13.2.4.3 clear ipv6 neighbors

The **clear ipv6 neighbors** command removes the specified dynamic IPv6 neighbor discovery cache entries. Commands that do not specify an IPv6 address remove all dynamic entries for the listed interface. Commands that do not specify an interface remove all dynamic entries.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 neighbors [**PORT**][**DYNAMIC_IPV6**]

Parameters

- **PORT** Interface through which neighbor is accessed. Options include:
  - ***no parameter*** all dynamic entries are removed.

- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-channel interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.

- **DYNAMIC_IPV6** Address of entry removed by the command. Options include:
  - ***no parameter*** All dynamic entries for specified interface are removed.
  - ***ipv6_addr*** IPv6 address of entry.

Example

This command removes all dynamic neighbor entries for ***vlan 200***.

```
switch# clear ipv6 neighbors vlan 200
switch#
```

13.2.4.4 ipv6 address

The **ipv6 address** command assigns a global IPv6 address to the IPv6 interface, and defines the prefix length. This command is supported on routable interfaces. Multiple global IPv6 addresses can be assigned to an interface.

The **no ipv6 address** and **default ipv6 address** commands remove the IPv6 address assignment from the configuration mode interface by deleting the corresponding **ipv6 address** command from ***running-config***. If the command does not include an address, all address assignments are removed from the interface. IPv6 remains enabled on the interface after the removal of all IPv6 addresses only if an **ipv6 enable** command is configured on the interface.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 address [***ipv6_prefix***]

no ipv6 address [***ipv6_prefix***]

default ipv6 address [***ipv6_prefix***]

Parameter

ipv6_prefix address assigned to the interface (CIDR notation).

Guidelines

This command is supported on routable interfaces.

Example

These commands configure an IPv6 address and prefix length for ***interface vlan 200***:

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 address 10:23:31:00:01:32:93/64
switch(config-if-Vl200)#
```

13.2.4.5 ipv6 dhcp relay always-on

The **iv6p dhcp relay always-on** command enables the switch DHCP relay agent on the switch regardless of the DHCP relay agent status on any interface. By default, the DHCP relay agent is enabled only if at least one routable interface is configured with an **ipv6 dhcp relay destination** statement.

The **no ipv6 dhcp relay always-on** and **default ipv6 dhcp relay always-on** commands remove the **ipv6 dhcp relay always-on** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ipv6 dhcp relay always-on

no ipv6 dhcp relay always-on

default ipv6 dhcp relay always-on

Example

This command enables the DHCP relay agent.

```
switch(config)# ipv6 dhcp relay always-on
switch(config)#
```

13.2.4.6 ipv6 dhcp relay destination

The **ipv6 dhcp relay destination** command enables the DHCPv6 relay agent and sets the destination address on the configuration mode interface.

The **no ipv6 dhcp relay destination** and **default ipv6 dhcp relay destination** commands remove the corresponding **ipv6 dhcp relay destination** command from ***running-config***. When the commands do not list an IPv6 address, all **ipv6 dhcp relay destination** commands are removed from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 dhcp relay destination [***ipv6_addr***][***source-address ipv6_addr***]

no ipv6 dhcp relay destination [***ipv6_addr***]

default ipv6 dhcp relay destination [***ipv6_addr***]

Parameters

- ***ipv6_addr*** DCHP Server's IPv6 address.
- **source-address *ipv6_addr*** specify the source IPv6 address to communicate with DHCP server.

Guidelines

If the source-address parameter is specified, then the DHCP client receives an IPv6 address from the subnet of source IP address. The source-address must be one of the configured addresses on the interface.

Example

This command enables the DHCPv6 relay agent and sets the destination address to ***2001:0db8:0:1::1*** on ***interface ethernet 4***.

```
switch(config)# interface ethernet 4
switch(config-if-Et4)# ipv6 dhcp relay destination 2001:0db8:0:1::1
switch(config-if-Et4)# show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCPv6 Relay Link-layer Address Option (79) is disabled
DHCP Smart Relay is disabled
Interface: Ethernet4
  DHCP Smart Relay is disabled
  DHCP servers: 1::1
                2001:db8:0:1::1
switch(config-if-Et4)#
```

13.2.4.7 ipv6 dhcp relay option link-layer address

The **`ipv6 dhcp relay option link-layer address`** command enables the DHCPv6 relay agent to configure the client link layer address option to solicit and request messages. In other words, the command enables the link layer address option (79) in the global configuration mode.

The **`no ipv6 dhcp relay option link-layer address`** command disables the link layer address option (79) in the global configuration mode.

Command Mode

Global Configuration

Command Syntax

`ipv6 dhcp relay option link-layer address`

`no ipv6 dhcp relay option link-layer address`

`default ipv6 dhcp relay option link-layer address`

Example

This command enables the insertion of link layer address option (79) in the global configuration mode.

```
switch(config)# ipv6 dhcp relay option link-layer address
```

13.2.4.8 ipv6 enable

The **`ipv6 enable`** command enables IPv6 on the configuration mode interface. Assigning an IPv6 address to an interface also enables IPv6 on the interface.

The **`no ipv6 enable`** and **`default ipv6 enable`** command remove the corresponding **`ipv6 enable`** command from ***running-config***. This action disables IPv6 on interfaces that are not configured with an IPv6 address.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 enable

no ipv6 enable

default ipv6 enable

Example

This command enables IPv6 on ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-vl200)# ipv6 enable
switch(config-vl200)#
```

13.2.4.9 ipv6 hardware fib aggregate-address

The **ipv6 hardware fib aggregate-address** command specifies the routing table repository of specified IPv6 route.

By default, routes that are created statically through the CLI or dynamically through routing protocols are initially stored in software routing tables, then entered in the hardware routing table by the routing agent. This command prevents the entry of the specified route into the hardware routing table. Specified routes that are in the hardware routing table are removed by this command. Specific routes that are encompassed within the specified route prefix are affected by this command.

The **no ipv6 hardware fib aggregate-address** and **default ipv6 hardware fib aggregate-address** commands remove the restriction from the hardware routing table for the specified routes by removing the corresponding **ipv6 hardware fib aggregate-address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ipv6 hardware fib aggregate-address ***ipv6_prefix*** summary-only software-forward

no ipv6 hardware fib aggregate-address ***ipv6_prefix***

default ipv6 hardware fib aggregate-address ***ipv6_prefix***

Parameters

ipv6_prefix IPv6 prefix that is restricted from the hardware routing table (CIDR notation).

Example

These commands configure a hardware routing restriction for an IPv6 prefix, then displays that restriction.

```
switch(config)# ipv6 hardware fib aggregate-address fd77:4890:5313:ffed::/64
summary-only software-forward
switch(config)# show ipv6 hardware fib aggregate-address
Codes: S - Software Forwarded
S  fd77:4890:5313:ffed::/64

switch(config)#
```

13.2.4.10 ipv6 hardware fib ecmp resilience

The **ip hardware fib ecmp resilience** command configures a fixed number of next hop entries in the hardware ECMP table for the specified IPv6 address prefix. In addition to specifying the maximum number of next hop addresses that the table can contain for the prefix, the command introduces a redundancy factor that

allows duplication of each next hop address. The fixed table space for the address is the maximum number of next hops multiplied by the redundancy factor.

The default method of adding or removing next hop entries when required by the active hashing algorithm leads to inefficient management of the ECMP table, which can result in the rerouting of packets to different next hops that breaks TCP packet flows. Implementing fixed table entries for a specified IP address allows data flows that are hashed to a valid next hop number to remain intact. Additionally, traffic is evenly distributed over a new set of next hops.

The **no ip hardware fib ecmp resilience** and **default ip hardware fib ecmp resilience** commands restore the default hardware ECMP table management by removing the **ip hardware fib ecmp resilience** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ipv6 hardware fib ecmp resilience ***net_prfx*** capacity *nhop_max* redundancy ***duplicates***

no ipv6 hardware fib ecmp resilience ***net_addr***

default ipv6 hardware fib ecmp resilience ***net_addr***

Parameters

- ***net_prfx*** IPv6 address prefix managed by command.
- ***nhop_max*** Specifies maximum number of nexthop entries for specified IP address prefix. Value range varies by platform:
  - Helix: <**2** to **64**>
  - Trident: <**2** to **32**>
  - Trident II: <**2** to **64**>
- ***duplicates*** Specifies the redundancy factor. Value ranges from ***1*** to ***128***.

Example

This command configures a hardware ECMP table space of 15 entries for the IPv6 address ***2001:db8:0::/64***. A maximum of five nexthop addresses can be specified for the address. When the table contains five nexthop addresses, each appears in the table three times. When the table contains fewer than five nexthop addresses, each is duplicated until the 15 table entries are filled.

```
switch(config)#ipv6 hardware fib ecmp resilience 2001:db8:0::/64 capacity 5
 redundancy 3
```

13.2.4.11 ipv6 hardware fib nexthop-index

The **ipv6 hardware fib nexthop-index** command deterministically selects the next hop used for ECMP routes. By default, routes that are created statically through the CLI or dynamically through routing protocols are initially stored in software routing tables, then entered in the hardware routing table by the routing agent. This command specifies the method of creating an index-offset number that points to the next hop from the list of the route's ECMP next hops.

The index-offset is calculated by adding the next hop index to a prefix offset.

- **Next hop index:** specified in the command.
- **Prefix offset:** the least significant bits of the route's prefix.

The command specifies the number of bits that comprise the prefix offset. The prefix offset is set to the prefix when the command specifies a prefix size larger than the prefix. If the command specifies an prefix size of zero, the prefix-offset is also zero and the index-offset is set to the next hop index.

When the index-offset is greater than the number of next hops in the table, the position of the next hop is the remainder of the division of the index-offset by the number of next hop entries.

The **no ipv6 hardware fib nexthop-index** and **default ipv6 hardware fib nexthop-index** commands remove the specified nexthop used for ECMP routes by removing the **ipv6 hardware fib nexthop-index** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ipv6 hardware fib nexthop ***nxthop_index*** [**PREFIX**]

no ipv6 hardware fib nexthop

default ipv6 hardware fib nexthop

Parameters

- ***nxthop_index*** specifies the next hop index. Value ranges from ***0*** to ***32***.
- **PREFIX** Number of bits of the route's prefix to use as the prefix-offset. Value ranges from ***0*** to ***64***.
  - ***no parameter*** The prefix offset is set to zero.
  - **prefix-bits *0 to 64*** Specifies the number bits to use as the prefix-offset.

Example

This command specifies the next hop from the list of ECMP next hops for the route.

```
switch(config)# ipv6 hardware fib nexthop-index 5 prefix-bits 10

switch> show ip
IP Routing : Enabled
IP Multicast Routing : Disabled
VRRP: Configured on 0 interfaces

IPv6 Unicast Routing : Enabled
IPv6 ECMP Route support : False
IPv6 ECMP Route nexthop index: 5
IPv6 ECMP Route num prefix bits for nexthop index: 10 switch>
```

13.2.4.12 ipv6 nd managed-config-flag

The **ipv6 nd managed-config-flag** command causes the **managed address configuration** flag to be set in IPv6 RA packets transmitted from the configuration mode interface.

The **no ipv6 nd managed-config-flag** and **default ipv6 nd managed-config-flag** commands restore the default setting where the **managed address configuration** flag is not set in IPv6 RA packets transmitted by the interface by removing the corresponding **ipv6 nd managed-config-flag** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd managed-config-flag

no ipv6 nd managed-config-flag

default ipv6 nd managed-config-flag

Example

These commands cause the managed address configuration flag to be set in IPv6 RA packets sent from ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd managed-config-flag
switch(config-if-Vl200)#
```

13.2.4.13 ipv6 nd ns-interval

The **ipv6 nd ns-interval** command configures the interval between IPv6 Neighbor Solicitation (NS) transmissions from the configuration mode interface.

The **no ipv6 nd ns-interval** and **default ipv6 nd ns-interval** commands return the IPv6 NS transmission interval for the configuration mode interface to the default value of ***1000*** milliseconds by removing the corresponding **ipv6 nd ns-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd ns-interval ***period*** **no ipv6 nd ns-interval**

default ipv6 nd ns-interval

Parameter

period interval in milliseconds between successive IPv6 neighbor solicitation transmissions. Values range from ***1000*** to ***4294967295***. The default period is ***1000*** milliseconds.

Example

This command configures a neighbor solicitation transmission interval of ***30*** seconds on ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ns-interval 30000
switch(config-if-Vl200)#
```

13.2.4.14 ipv6 nd other-config-flag

The **ipv6 nd other-config-flag** command configures the configuration mode interface to send IPv6 RAs with the **other stateful configuration** flag set.

The **no ipv6 nd other-config-flag** and **default ipv6 nd other-config-flag** commands restore the default setting by removing the corresponding **ipv6 nd other-config-flag** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ipv6 nd other-config-flag`

`no ipv6 nd other-config-flag`

`default ipv6 nd other-config-flag`

Example

These commands configure the switch to set the other stateful configuration flag in advertisements sent from ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd other-config-flag
switch(config-if-Vl200)#
```

13.2.4.15 ipv6 nd prefix

The **`ipv6 nd prefix`** command configures neighbor discovery Router Advertisements (RAs) prefix inclusion for RAs sent from the configuration mode interface.

By default, all prefixes configured as IPv6 addresses are advertised in the interface's RAs. The **`ipv6 nd prefix`** command with the ***no-advertise*** option prevents advertising of the specified prefix without affecting the advertising of other prefixes specified as IPv6 addresses. When an interface configuration includes at least one **`ipv6 nd prefix`** command that enables prefix advertising, RAs advertise only prefixes specified through **`ipv6 nd prefix`** commands.

Commands enabling prefix advertising also specify the advertised valid and preferred lifetime periods. Default periods are ***2,592,000*** (valid) and ***604,800*** (preferred) seconds.

The **`no ipv6 nd prefix`** and **`default ipv6 nd prefix`** commands remove the corresponding **`ipv6 nd prefix`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ipv6 nd prefix` ***ipv6_prefix*** **LIFETIME** [**FLAGS**]

`ipv6 nd prefix` ***ipv6_prefix*** no-advertise

`no ipv6 nd prefix` ***ipv6_prefix***

`default ipv6 nd prefix` ***ipv6_prefix***

Parameters

- ***ipv6_prefix*** IPv6 prefix (CIDR notation).

- **no-advertise** Prevents advertising of the specified prefix.
- **LIFETIME** Period that the specified IPv6 prefix is advertised (seconds). Options include:
  - **valid preferred** Two values that set the *valid* and *preferred* lifetime periods.
  - **valid** One value that sets the *valid* lifetime. The *preferred* lifetime is set to the default value.
  - ***no parameter*** The *valid* and *preferred* lifetime periods are set to their default values.

Options for **valid**: ***0*** **to** ***4294967295*** and **infinite**. Default value is ***2592000*** Options for **preferred**: ***0*** **to** ***4294967295*** and **infinite**. The default value is ***604800***. The maximum value (***4294967295***) and **infinite** are equivalent settings.

- **FLAGS** **on-link** and **autonomous address-configuration** flag values in RAs.
  - ***no parameter*** both flags are set.
  - **no-autoconfig** **autonomous address-configuration** flag is reset.
  - **no-onlink** **on-link** flag is reset.
  - **no-autoconfig no-onlink** both flags are reset.
  - **no-onlink no-autoconfig** both flags are reset.

Example

These commands enable neighbor discovery advertising for IPv6 address ***3012:D678::/64***, on ***interface vlan 200***, specifying a valid lifetime of ***1,296,000*** seconds and the default preferred lifetime.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd prefix 3012:D678::/64 1296000
```

13.2.4.16 ipv6 nd ra disabled

The **ipv6 nd ra disabled** command suppress IPv6 Router Advertisement (RA) transmissions on the configuration mode interface. By default, only unsolicited RAs that are transmitted periodically are suppressed. The **all** option configures the switch to suppress all RAs, including those responding to a router solicitation.

The **no ipv6 nd ra disabled** and **default ipv6 nd ra disabled** commands restore the transmission of RAs on the configuration mode interface by deleting the corresponding **ipv6 nd ra disabled** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd ra disabled [**SCOPE**]

no ipv6 nd ra disabled

default ipv6 nd ra disabled

Parameters

SCOPE specifies the RAs that are suppressed.

- ***no parameter*** Periodic unsolicited RAs are suppressed.
- **all** All RAs are suppressed.

Example

This command suppresses all RAs on ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-vl200)# ipv6 nd ra disabled all
switch(config-vl200)#
```

13.2.4.17 ipv6 nd ra dns-server

The **ipv6 nd ra dns-server** command configures the IPv6 address of a preferred Recursive DNS Server (RDNSS) for the command mode interface to include in its neighbor-discovery Router Advertisements (RAs). Including RDNSS information in RAs provides DNS server configuration for connected IPv6 hosts without requiring DHCPv6.

Multiple servers can be configured on the interface by using the command repeatedly. A lifetime value for the RDNSS can optionally be specified with this command, and overrides any default value configured for the interface using the **ipv6 nd ra dns-servers lifetime** command.

The **no ipv6 nd ra dns-server** and **default ipv6 nd ra dns-server** commands remove the corresponding **ipv6 nd ra dns-server** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd ra dns-server ***ipv6_addr*** **SERVER_LIFE**

no ipv6 nd ra dns-server ***ipv6_addr***

default ipv6 nd ra dns-server ***ipv6_addr***

Parameters

- ***ipv6_addr*** RDNSS address to be included in RAs from the command mode interface.
- **SERVER_LIFE** maximum lifetime value for the specified RDNSS entry. This value overrides any default lifetime value. Value should be between the RA interval configured on the interface and two times that interval. Options include:
  - ***no parameter*** lifetime period is the default lifetime period configured on the interface. If no lifetime period is configured on the interface, the default value is 1.5 times the maximum RA interval set by the **ipv6 nd ra interval** command.
  - **lifetime 0** the configured RDNSS is not to be used.
  - **lifetime *1 to 4294967295*** specifies the lifetime period for this RDNSS in seconds.

Example

This command configures the RDNSS at ***2001:0db8:0:1::1*** as a preferred RDNSS for ***interface vlan 200*** to include in its neighbor-discovery route advertisements, and sets its lifetime value to ***300*** seconds.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra dns-server 2001:0db8:0:1::1 lifetime 300
switch(config-if-Vl200)#
```

13.2.4.18 ipv6 nd ra dns-servers lifetime

The **ipv6 nd ra dns-servers lifetime** command sets the default value that the configuration mode interface uses for the lifetime of any Recursive DNS Server (RDNSS) configured on the interface. A lifetime value set for an individual RDNSS overrides this value. The lifetime value is the maximum amount of time after a route advertisement packet is sent that the RDNSS referenced in the packet may be used for name resolution.

The **no ipv6 nd ra dns-servers lifetime** and **default ipv6 nd ra dns-servers lifetime** commands remove the default lifetime value from the interface by removing the corresponding **ipv6 nd ra dns-servers lifetime** command from ***running-config***. When there is no default RDNSS lifetime value configured on the interface, an RDNSS without a custom lifetime value will default to 1.5 times the RA interval configured on the interface. A lifetime of zero seconds means that the RDNSS must not be used for name resolution.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd ra dns-servers lifetime ***period***

no ipv6 nd ra dns-servers lifetime

default ipv6 nd ra dns-servers lifetime

Parameters

period the RDNSS lifetime value for the configuration mode interface. Options include:

- ***0*** any RDNSS configured on the command mode interface without a custom lifetime value must not be used.
- ***1 to 4294967295*** maximum RDNSS lifetime value for the configuration mode interface. This value is overridden by any lifetime value set with the **ipv6 nd ra dns-server** command. Should be between the router advertisement interval configured on the interface and two times that interval.

Example

This command sets the default RDNSS maximum lifetime value for ***interface vlan 200*** to ***350*** seconds.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra dns-servers lifetime 350
switch(config-if-Vl200)#
```

13.2.4.19 ipv6 nd ra dns-suffix

The **ipv6 nd ra dns-suffix** command creates a DNS Search List (DNSSL) for the command mode interface to include in its neighbor-discovery Router Advertisements as defined in RFC 6106 . The DNSSL contains the domain names of DNS suffixes for IPv6 hosts to append to short, unqualified domain names for DNS queries.

Multiple DNS domain names can be added to the DNSSL by using the command repeatedly. A lifetime value for the DNSSL can optionally be specified with this command, and overrides any default value configured for the interface using the ipv6 nd ra dns-suffixes lifetime command.

The **no ipv6 nd ra dns-suffix** and **default ipv6 nd ra dns-suffix** commands remove the corresponding **ipv6 nd ra dns-suffix** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd ra dns-suffix ***domain*** **SUFFIX_LIFE**

no ipv6 nd ra dns-suffix ***ipv6_addr***

default ipv6 nd ra dns-suffix ***ipv6_addr***

Parameters

- ***domain*** domain suffix for IPv6 hosts to append to short, unqualified domain names for DNS queries. Suffix must contain only alphanumeric characters, "**.**" and "**-**" and must begin and end with an alphanumeric character.
- **SUFFIX_LIFE** maximum lifetime value for the specified domain suffix. This value overrides any default lifetime value. Value should be between the RA interval configured on the interface and two times that interval. Options include:
  - ***no parameter*** lifetime period is the default lifetime period configured on the interface. If no lifetime period is configured on the interface, the default value is 1.5 times the maximum RA interval set by the ipv6 nd ra interval command.
  - **lifetime 0** the configured domain suffix is not to be used.
  - **lifetime *1 to 4294967295*** specifies the lifetime period for this domain suffix in seconds.

Example

These commands create a DNSSL for ***interface vlan 200*** to include in its neighbor-discovery route advertisements, and set its lifetime value to ***300*** seconds.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra dns-suffix test.com lifetime 300
switch(config-if-Vl200)#
```

13.2.4.20 ipv6 nd ra dns-suffixes lifetime

The **ipv6 nd ra dns-suffixes lifetime** command sets the default value that the configuration mode interface uses for the lifetime of any DNS Search List (DNSSL) configured on the interface. A lifetime value set for an individual DNSSL overrides this value. The lifetime value is the maximum amount of time after a route advertisement packet is sent that the DNSSL included in the packet may be used for name resolution.

The **no ipv6 nd ra dns-suffixes lifetime** and **default ipv6 nd ra dns-suffixes lifetime** commands remove the default lifetime value from the interface by removing the corresponding **ipv6 nd ra dns-suffixes lifetime** command from ***running-config***. When there is no default DNSSL lifetime value configured on the interface, a DNSSL without a custom lifetime value will default to 1.5 times the RA interval configured on the interface. A lifetime of zero seconds means that the DNSSL must not be used for name resolution.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ipv6 nd ra dns-suffixes lifetime` ***period***

`no ipv6 nd ra dns-suffixes lifetime`

`default ipv6 nd ra dns-suffixes lifetime`

Parameters

period the DNSSL lifetime value for the configuration mode interface. Options include:

- ***0*** any DNSSL configured on the command mode interface without a custom lifetime value must not be used.
- ***1 to 4294967295*** maximum DNSSL lifetime value for the configuration mode interface. This value is overridden by any lifetime value set with the **`ipv6 nd ra dns-suffix`** command. Should be between the RA interval configured on the interface and two times that interval.

Example

This command sets the default DNSSL maximum lifetime value for ***interface vlan 200*** to ***350*** seconds.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra dns-suffixes lifetime 350
switch(config-if-Vl200)#
```

13.2.4.21 ipv6 nd ra hop-limit

The **`ipv6 nd ra hop-limit`** command sets a suggested hop-limit value to be included in Router Advertisement (RA) packets. The hop-limit value is to be used by attached hosts in outgoing packets.

The **`no ipv6 nd ra hop-limit`** and **`default ipv6 nd ra hop-limit`** commands remove the corresponding **`ipv6 nd ra hop-limit`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ipv6 nd ra hop-limit` ***quantity***

`no ipv6 nd ra hop-limit lifetime`

`default ipv6 nd ra hop-limit lifetime`

Parameters

quantity the hop-limit value to be included in RA packets sent by the configuration mode interface. Options include:

- ***0*** indicates that outgoing packets from attached hosts are to be immediately discarded.
- ***1 to 255*** number of hops. The default value is ***64***.

Example

These commands include a hop-limit value of ***100*** in RA packets sent by ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra hop-limit
switch(config-if-Vl200)#
```

13.2.4.22 ipv6 nd ra interval

The **ipv6 nd ra interval** command configures the interval between IPv6 Router Advertisement transmissions from the configuration mode interface.

The **no ipv6 nd ra interval** and **default ipv6 nd ra interval** commands return the IPv6 RA transmission interval for the configuration mode interface to the default value of ***200*** seconds by removing the corresponding **ipv6 nd ra interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd ra interval [**SCALE**] ***ra_period*** [***minimum_period***]

no ipv6 nd ra interval

default ipv6 nd ra interval

Parameters

- **SCALE** timescale in which command parameter values are expressed.
  - ***no parameter*** seconds.
  - **msec** milliseconds.
- ***ra_period*** maximum interval between successive IPv6 RA transmissions. The default period is ***200*** seconds.
  - ***4 - 1800*** valid range when **SCALE** is set to default value (seconds).
  - ***500 - 1800000*** valid range when **SCALE** is set to **msec**.
- ***minimum_period*** minimum interval between successive IPv6 RA transmissions. Must be smaller than ***ra_period***. By default, a minimum period is not defined.
  - ***no parameter*** Command does not specify a minimum period.
  - ***3 - 1799*** valid range when **scale** is set to default value (seconds).
  - ***375 - 1799999*** valid range when **scale** is set to **msec**.

Example

These commands configure a RA transmission interval of ***60*** seconds on VLAN interface ***interface vlan 200***, then displays the interface status.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra interval 60
switch(config-if-Vl200)# show active
interface Vlan200
   ipv6 nd ra interval 60
```

```
switch(config-if-Vl200)#
```

13.2.4.23 ipv6 nd ra lifetime

The **ipv6 nd ra lifetime** command specifies the value that the switch places in the **router lifetime** field of IPv6 Router Advertisements sent from the configuration mode interface.

If the value is set to ***0***, IPv6 peers connected to the specified interface will remove the switch from their lists of default routers. Values greater than ***0*** indicate the time in seconds that peers should keep the router on their default router lists without receiving further RAs from the switch. Unless the value is ***0***, the router lifetime value should be equal to or greater than the interval between unsolicited RAs sent on the interface.

The **no ipv6 nd ra lifetime** and **default ipv6 nd ra lifetime** commands return the IPv6 RA lifetime data entry filed for the configuration mode interface to the default value of ***1800*** seconds by removing the corresponding **ipv6 nd ra lifetime** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd ra lifetime ***ra_lifetime***

no ipv6 nd ra lifetime

default ipv6 nd ra lifetime

Parameters

ra_lifetime router lifetime period (seconds). Default value is 1800. Options include:

- ***0*** Router should not be considered as a default router.
- ***1 - 65535*** Lifetime period advertised in RAs. Should be greater than or equal to the interval between IPv6 RA transmissions from the configuration mode interface as set by the **ipv6 nd ra interval** command.

Example

This command configures the switch to enter ***2700*** in the router lifetime field of RAs transmitted from ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd ra lifetime 2700
switch(config-if-Vl200)# show active
interface Vlan20
   ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#
```

13.2.4.24 ipv6 nd ra mtu suppress

The **ipv6 nd ra mtu suppress** command suppresses the Router Advertisement (RA) MTU option on the configuration mode interface. The MTU option causes an identical MTU value to be advertised by all nodes on a link. By default, the RA MTU option is not suppressed.

The **no ipv6 nd ra mtu suppress** and **default ipv6 nd ra mtu suppress** commands restores the MTU option setting to enabled by for the configuration mode interface by removing the corresponding **ipv6 nd ra mtu suppress** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd ra mtu suppress

no ipv6 nd ra mtu suppress

default ipv6 nd ra mtu suppress

Example

This command suppresses the MTU option on ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-vl200)# ipv6 nd ra mtu suppress
switch(config-vl200)#
```

13.2.4.25 ipv6 nd reachable-time

The **ipv6 nd reachable-time** command specifies the time period that the switch includes in the reachable time field of RAs sent from the configuration mode interface. The reachable time defines the period that a remote IPv6 node is considered reachable after a reachability confirmation event.

RAs that advertise zero seconds indicate that the router does not specify a reachable time. The default advertisement value is 0 seconds. The switch reachability default period is 30 seconds.

The **no ipv6 nd reachable-time** and **default ipv6 nd reachable-time** commands restore the entry of the default value (***0***) in RAs sent from the configuration mode interface by deleting the corresponding **ipv6 nd reachable-time** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 nd reachable-time ***period***

no ipv6 nd reachable-time

default ipv6 nd reachable-time

Parameter

period Reachable time value (milliseconds). Value ranges from ***0 to 4294967295***. Default is ***0***.

Example

These commands configure the entry of ***25000*** (25 seconds) in the reachable time field of RAs sent from ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd reachable-time 25000
interface Vlan200
   ipv6 address fd7a:4321::1/64
   ipv6 nd reachable-time 25000
switch(config-if-Vl200)#
```

13.2.4.26 ipv6 nd router-preference

The **`ipv6 nd router-preference`** command specifies the value that the switch enters in the Default Router Preference (DRP) field of Router Advertisements (RAs) that it sends from the configuration mode interface. The default field entry value is **medium**.

The **`no ipv6 nd router-preference`** and **`default ipv6 nd router-preference`** commands restore the switch to enter the default DRP field value of **medium** in RAs sent from the configuration mode interface by deleting the corresponding **`ipv6 nd router-preference`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`ipv6 nd router-preference` **RANK**

`no ipv6 nd router-preference`

`default ipv6 nd router-preference`

Parameters

RANK Router preference value. Options include:

- **high**
- **low**
- **medium**

Example

This command configures the switch as a medium preference router on RAs sent from ***interface vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 nd router-preference medium
switch(config-if-Vl200)#
```

13.2.4.27 ipv6 neighbor cache persistent

The **`ipv6 neighbor cache persistent`** command restores the IPv6 neighbor cache after reboot.

The **`no ipv6 neighbor cache persistent`** and **`default ipv6 neighbor cache persistent`** commands remove the ARP cache persistant configuration from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ipv6 neighbor cache persistent`

`no ipv6 neighbor cache persistent`

`default ipv6 neighbor cache persistent`

Example

This command restores the ipv6 neighbor cache after reboot.

```
switch(config)# ipv6 neighbor cache persistent
switch(config)#
```

13.2.4.28 ipv6 neighbor

The **`ipv6 neighbor`** command creates an IPv6 neighbor discovery cache static entry. The command converts pre-existing dynamic cache entries for the specified address to static entries.

The **`no ipv6 neighbor`** and **`default ipv6 neighbor`** commands remove the specified static entry from the IPV6 neighbor discovery cache and delete the corresponding **`ipv6 neighbor`** command from ***running-config***. These commands do not affect any dynamic entries in the cache.

Command Mode

Global Configuration

Command Syntax

`ipv6 neighbor` ***ipv6_addr*** **PORT** ***mac_addr***

`no ipv6 neighbor` ***ipv6_address*** **PORT**

`default ipv6 neighbor` ***ipv6_addr*** **PORT**

Parameters

- ***ipv6_addr*** Neighbor's IPv6 address.
- **PORT** Interface through which the neighbor is accessed. Options include:
  - **ethernet** ***e_num*** Ethernet interface specified by ***e_num***.
  - **loopback** ***l_num*** Loopback interface specified by ***l_num***.
  - **management** ***m_num*** Management interface specified by ***m_num***.
  - **port-channel** ***p_num*** Port-channel interface specified by ***p_num***.
  - **vlan** ***v_num*** VLAN interface specified by ***v_num***.
  - **vxlan** ***vx_num*** VXLAN interface specified by ***vx_num***.
  - ***mac_addr*** Neighbor's data-link (hardware) address. (48-bit dotted hex notation – H.H.H).

Example

This command will add a static entry to the neighbor discovery cache for the neighbor located at ***3100:4219::3EF2*** with hardware address ***0100.4EA1.B100*** and accessible through ***vlan 200***.

```
switch(config)# ipv6 neighbor 3100:4219::3EF2 vlan 200 0100.4EA1.B100
switch(config)#
```

13.2.4.29 ipv6 route

The **ipv6 route** command creates an IPv6 static route. The destination is a IPv6 prefix; the source is an IPv6 address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

By default, the administrative distance assigned to static routes is ***1***. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of ***200*** is overridden by OSPF intra-area routes, which have a default distance of ***110***.

The command provides these methods of designating the nexthop location:

- **null0:** Traffic to the specified destination is dropped.
- **IPv6 gateway:** Switch identifies egress interface by recursively resolving the next-hop.
- **Egress interface:** Switch assumes destination subnet is directly connected to interface; when routing to any subnet address, the switch sends an ARP request to find the MAC address for the first packet.
- **Combination Egress interface and IPv6 gateway:** Switch does not assume subnet is directly connected to interface; the only ARP traffic is for the nexthop address for the first packet on the subnet. Combination routes are not recursively resolved.

Multiple routes that are configured to the same destination with the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic across all ECMP route paths equally. All ECMP paths are assigned the same tag value; commands that change the tag value of any ECMP path change the tag value of all paths in the ECMP.

The **no ipv6 route** and **default ipv6 route** commands delete static routes by removing the corresponding **ipv6 route** statements from ***running-config***. Commands not including a source delete all statements to the destination. Only statements with parameters that match specified command arguments are deleted. Parameters that are not in the command line are not evaluated.

Command Mode

Global Configuration

Command Syntax

ipv6 route ***dest_prefix*** **NEXTHOP** [**DISTANCE**][**TAG_OPT**][**RT_NAME**]

no ipv6 route ***dest_prefix*** [***nexthop_addr***][**DISTANCE**]

default ipv6 route ***dest_prefix*** [***nexthop_addr***][**DISTANCE**]

Parameters

- ***dest_prefix*** Destination IPv6 prefix (CIDR notation).
- **NEXTHOP** Access method of next hop device. Options include:
  - **null0** Null0 interface – route is dropped.
  - ***nexthop_addr*** IPv6 address of nexthop device.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
  - **ethernet *e_num nexthop_addr*** Combination route (Ethernet interface and gateway).
  - **loopback *l_num nexthop_addr*** Combination route (loopback interface and gateway).
  - **management *m_num nexthop_addr*** Combination route (management interface and gateway).
  - **port-channel *p_num nexthop_addr*** Combination route (port channel interface and gateway).
  - **vlan *v_num nexthop_addr*** Combination route (VLAN interface and gateway).
  - **vxlan *vx_num nexthop_addr*** Combination route (VXLAN interface and gateway).

- **DISTANCE** administrative distance assigned to route. Options include:
  - ***no parameter*** route assigned default administrative distance of one.
  - ***1 to 255*** The administrative distance assigned to route.
- **TAG_OPT** static route tag. Options include:
  - ***no parameter*** assigns default static route tag of ***0***.
  - **tag *0 to 4294967295*** Static route tag value.
- **RT_NAME** Associates descriptive text to the route. Options include:
  - ***no parameter*** No text is associated with the route.
  - **name *descriptive_text*** The specified text is assigned to the route.

Example

This command creates an IPv6 static route.

```
switch(config)# ipv6 route 10:23:31:00:01:32:93/24 vlan 300
```

13.2.4.30 ipv6 unicast-routing

The **ipv6 unicast-routing** command enables the forwarding of IPv6 unicast packets. When routing is enabled, the switch attempts to deliver inbound packets to destination addresses by forwarding them to interfaces or next hop addresses specified by the IPv6 routing table.

The **no ipv6 unicast-routing** and default **ip ipv6 unicast-routing** commands disable IPv6 unicast routing by removing the **ipv6 unicast-routing** command from ***running-config***. Dynamic routes added by routing protocols are removed from the routing table. Static routes are preserved by default; the **delete-static-routes** option removes static entries from the routing table.

IPv6 unicast routing is disabled by default.

Command Mode

Global Configuration

Command Syntax

ipv6 unicast-routing

no ipv6 unicast-routing [**DELETE_ROUTES**]

default ipv6 unicast-routing [**DELETE_ROUTES**]

Parameters

DELETE_ROUTES Resolves routing table static entries when routing is disabled.

- ***no parameter*** Routing table retains static entries.
- **delete-static-routes** Static entries are removed from the routing table.

Example

This command enables IPv6 unicast-routing.

```
switch(config)# ipv6 unicast-routing
switch(config)#
```

13.2.4.31 ipv6 verify

The **ipv6 verify** command configures Unicast Reverse Path Forwarding (uRPF) for inbound IPv6 packets on the configuration mode interface. uRPF verifies the accessibility of source IP addresses in packets that the switch forwards.

uRPF defines two operational modes: strict mode and loose mode.

- **Strict mode**: uRPF also verifies that a packet is received on the interface that its routing table entry specifies for its return packet.
- **Loose mode**: uRPF validation does not consider the inbound packet's ingress interface.

The **no ipv6 verify** and **default ipv6 verify** commands disable uRPF on the configuration mode interface by deleting the corresponding **ipv6 verify** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 verify unicast source reachable-via RPF_MODE

no ipv6 verify unicast

default ipv6 verify unicast

Parameters

RPF_MODE Specifes the uRPF mode. Options include:

- **any** Loose mode.
- **rx** Strict mode.
- **rx allow-default** Strict mode. All inbound packets are forwarded if a default route is defined.

Guidelines

The first IPv6 uRPF implementation briefly disables IPv6 unicast routing. Subsequent **ip verify** commands on any interface do not disable IPv6 routing.

Example

This command enables uRPF strict mode on ***interface vlan 100***. When a default route is configured on the interface, all inbound packets are checked as valid.

```
switch(config)# interface vlan 100
switch(config-if-Vl100)# ipv6 verify unicast source reachable-via rx allow-
default
switch(config-if-Vl100)# show active
 interface Vlan100
   ipv6 verify unicast source reachable-via rx allow-default
switch(config-if-Vl100)#
```

13.2.4.32 ipv6 dhcp snooping

The **ipv6 dhcp snooping** command enables DHCP snooping globally on the switch.

The **no ipv6 dhcp snooping** and **default ipv6 dhcp snooping** commands disable global DHCP snooping by removing the ipv6 dhcp snooping command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ipv6 dhcp snooping [remote-id option | vlan [$ | vlan-range]]

no ipv6 dhcp snooping [remote-id option | vlan [$ | vlan-range]]

default ipv6 dhcp snooping [remote-id option | vlan [$ | vlan-range]]

Parameters

- **remote-id option** configures the remote ID option.
- **vlan** enables IPv6 DHCP snooping for a specific VLAN. Numbers range from 1 to 4094.
- **$** end of range.
- **vlan-range** VLANs based on the snooping enabled. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.

Examples

- The following configuration enables IPv6 DHCP snooping feature at the global level.

```
switch(config)# ipv6 dhcp snooping
switch(config)# ipv6 dhcp snooping remote-id option
switch(config)# ipv6 dhcp snooping vlan <vlan|vlan-range>
```

- The following command display IPv6 DHCP snooping state.

```
switch(config)# ipv6 dhcp snooping
switch(config)# show ipv6 dhcp snooping
DHCPv6 Snooping is enabled
DHCPv6 Snooping is operational
DHCPv6 Snooping is configured on following VLANs:
  2789-2790
DHCPv6 Snooping is operational on following VLANs:
  2789
Insertion of Option-37 is enabled
```

13.2.4.33 pim ipv6 sparse-mode

The **pim ipv6 sparse-mode** command enables PIM Sparse Mode (PIM-SM) and IGMP (router mode) on the configuration mode interface.

Note: PIM and multicast border router (MBR) must be mutually exclusive on an interface. If the interface is configured as an MBR, do not enable PIM on the interface.

The **no pim ipv6 sparse-mode** and **default pim ipv6 sparse-mode** commands restore the default PIM and IGMP (router mode) settings of ***disabled*** on the configuration mode interface by removing the **pim ipv6 sparse-mode** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel

Configuration Interface-VLAN Configuration

Command Syntax

pim ipv6 sparse-mode

no pim ipv6

no pim ipv6 sparse-mode

default pim ipv6

default pim ipv6 sparse-mode

Example

This command enables **pim ipv6 sparse-mode** on ***interface vlan 4*** interface.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# pim ipv6 sparse-mode
switch(config-if-Vl4)#
```

13.2.4.34 show ipv6 dhcp relay counters

The **show ipv6 dhcp relay counters** command displays the number of DHCP packets received, forwarded, or dropped on the switch and on all interfaces enabled as DHCP relay agents.

Command Mode

EXEC

Command Syntax

show ipv6 dhcp relay counters

Example

This command displays the IP DHCP relay counter table.

```
switch> show ipv6 dhcp relay counters
           |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |          Last Cleared
----------|----- ---- -----|----------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
Ethernet4 |  207  148    0 | 4 days, 19:54:24 ago

switch>
```

13.2.4.35 show ipv6 dhcp snooping

The **show ipv6 dhcp snooping** command displays information about the DHCP snooping configuration.

Command Mode

EXEC

Command Syntax

show ipv6 dhcp snooping

Related Commands

- ipv6 dhcp snooping
- show ipv6 dhcp snooping counters
- show ipv6 dhcp snooping hardware
- clear ipv6 dhcp snooping counters

Example

This command displays the switch's DHCP snooping configuration.

```
switch# show ipv6 dhcp snooping
DHCPv6 Snooping is enabled
DHCPv6 Snooping is operational
DHCPv6 Snooping is configured on following VLANs:
  2789-2790
```

```
DHCPv6 Snooping is operational on following VLANs:
  2789
Insertion of Option-37 is enabled
```

13.2.4.36 show ipv6 dhcp snooping counters

The **show ipv6 dhcp snooping counters** command displays counters that track the quantity of DHCP request and reply packets that the switch receives. Data is either presented for each VLAN or aggregated for all VLANs with counters for packets dropped.

Command Mode

EXEC

Command Syntax

show ipv6 dhcp snooping counters [**COUNTER_TYPE**]

Parameters

- **COUNTER_TYPE** The type of counter that the command displays.
- ***no parameter*** command displays counters for each VLAN.
- **debug** command displays aggregate counters and drop cause counters.

Examples

- This command displays the number of DHCP packets sent and received on each VLAN.

```
switch# show ipv6 dhcp snooping counters

     | Dhcpv6 Request Pkts | Dhcpv6 Reply Pkts |
Vlan |  Rcvd   Fwdd   Drop |  Rcvd  Fwdd  Drop | Last Cleared
-----|------ ------ -------|------ ----- ------|-------------
2789 |     1      1      0 |     1     1     0 |  0:03:09 ago
```

- This command displays the number of DHCP packets sent on the switch.

```
switch# show ipv6 dhcp snooping counters debug

Counter                        Snooping to Relay Relay to Snooping
------------------------------ ----------------- -----------------
Received                                       1                 1
Forwarded                                      1                 1
Dropped - Invalid VlanId                       0                 0
Dropped - Parse error                          0                 0
Dropped - Invalid Dhcp Optype                  0                 0
Dropped - Invalid Remote-ID Option             0                 0
Dropped - Snooping disabled                    0                 0

Last Cleared:  0:04:29 ago
```

13.2.4.37 show ipv6 dhcp snooping hardware

The **show ipv6 dhcp snooping hardware** command displays internal hardware DHCP snooping status on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 dhcp snooping hardware

Example

This command displays DHCP snooping hardware status.

```
switch# show ipv6 dhcp snooping hardware
DHCPv6 Snooping is enabled
DHCPv6 Snooping is enabled on following VLANs:
    2789
    Vlans enabled per Slice
        Slice:  Linecard0-0
        2789
        Slice:  Linecard0-1
        2789
        Slice:  Linecard0-2
        2789
        Slice:  Linecard0-3
        2789
```

13.2.4.38 show ipv6 hardware fib aggregate-address

The **show ipv6 hardware fib aggregate-address** command displays the IPv6 prefixes that are restricted from entry into the hardware routing table. The **ipv6 hardware fib aggregate-address** command configures IPv6 prefix restictions.

Command Mode

EXEC

Command Syntax

show ipv6 address fib aggregate-address [**ADDRESS**][**RESTRICTION**]

Parameters

- **ROUTE_FILTER** filters by IPv6 address. Options include:
  - ***no parameter*** Displays all routes.
  - ***ipv6_addr*** Command displays only specified address.
  - ***ipv6_prefix*** Command displays addresses filtered by specified prefix (CIDR notation).
- **RESTRICTION** filters by route restriction.
  - ***no parameter*** displays routes restricted from the hardware routing table.
  - **software-forward** displays routes restricted from the hardware routing table.

Example

This command displays the routes that are restricted from the hardware routing table.

```
switch> show ipv6 hardware fib aggregate-address
Codes: S - Software Forwarded
S  fd77:4890:5313:aaed::/64
S  fd77:4890:5313:ffed::/64 switch>
```

13.2.4.39 show ipv6 interface

The **ipv6 interface** command displays the status of specified routed interfaces that are configured for IPv6.

Command Mode

EXEC

Command Syntax

show ipv6 interface [**INTERFACE_NAME**][**INFO_LEVEL**]

Parameters

- **INTERFACE_NAME** interfaces for which command displays status.
  - ***no parameter*** all routed interfaces.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- **INFO_LEVEL** amount of information that is displayed. Options include:
  - ***no parameter*** command displays data block for each specified interface.
  - **brief** command displays table that summarizes IPv6 interface data.

Example

This command displays the status of ***interface vlan 903***.

```
switch> show ipv6 interface vlan 903
Vlan903 is up, line protocol is up (connected)
  IPv6 is enabled, link-local is fe80::21c:73ff:fe01:21e/64
  Global unicast address(es):
    fd7a:629f:52a4:fe10::3, subnet is fd7a:629f:52a4:fe10::/64
  Joined group address(es):
    ff02::1
    ff02::1:ff01:21e
    ff02::1:ff00:3
    ff01::2 switch>
```

13.2.4.40 show ipv6 nd ra internal state

The **ipv6 nd ra internal state** command displays the state of the IPv6 Router Advertisement (RA) daemon for the specified routable interface.

Command Mode

EXEC

Command Syntax

show ipv6 nd ra internal state [**INTERFACE_NAME**]

Parameters

INTERFACE_NAME interfaces for which command displays status.

- ***no parameter*** all routed interfaces.
- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.

Example

This command displays the IPv6 RA daemon for ***vlan 1243***.

```
switch> show ipv6 nd ra internal state vlan 1243
```

```
INTERFACE: Vlan3908
                                ifindex : 0x00000021
                                    mtu : 9212
                             numIpv6Addr : 2
                      numPrefixToAdvertise : 0
                       numPrefixToSuppress : 0
                                RaSuppress : 0
                             RsRspSuppress : 0
                          raIntervalMaxMsec : 200000
                          raIntervalMinMsec : 0
                          managedConfigFlag : 0
                            otherConfigFlag : 0
                              raMtuSuppress : 0
                                 raLifetime : 1800
                             reacheableTime : 0
                           routerPreference : 0
                                 lastRaTime : 2012-05-01 09:22:57.020634
                          lastRsRspSentTime :
                                nextTimeout : 171.474535 (sec)
                    raNotSentIntfNotReady : 0
                                 numRaSent : 219
                                 numRsRcvd : 0
                            numRsSuppressed : 0
                               numRsRspSent : 0
               numRsDroppedInvalidHopLimit : 0
               numPktDroppedUnexpectedType : 0
                             initialized : 1 switch>
```

13.2.4.41 show ipv6 neighbors

The **`show ipv6 neighbors`** command displays the IPv6 neighbor discovery cache. The command provides filters to restrict the list to a specified IPv6 address or routable interface.

Command Mode

EXEC

Command Syntax

`show ipv6 neighbors` [**PORT**][**SOURCE**][**INFO_LEVEL**]

Parameters

- **PORT** Filters by interface through which neighbor is accessed. Options include:
  - ***no parameter*** all routed interfaces.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- **SOURCE** Filters by neighbor IPv6 address. Options include:
  - ***no parameter*** all IPv6 neighbors.
  - ***ipv6_addr*** IPv6 address of individual neighbor.
- **INFO_LEVEL** amount of information that is displayed. Options include:
  - ***no parameter*** command displays the discovery cache for the specified interfaces.
  - **summary** command displays summary information only.

Example

This command displays the IPv6 neighbor discovery cache for IPv6 address ***fe80::21c:73ff:fe01:5fe1***.

```
switch> show ipv6 neighbors fe80::21c:73ff:fe01:5fe1
IPv6 Address                              Age Hardware Addr    State Interface
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Et12
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Po999
```

```
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl102
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl103
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl205
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl207
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl3901
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl3902
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl3903
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl3904
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl3905
fe80::21c:73ff:fe01:5fe1                  0 001c.d147.8214   REACH Vl3996
```

13.2.4.42 show ipv6 route

The **show ipv6 route** command displays IPv6 routing table entries that are in the Forwarding Information Base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **show running-config** command displays all configured routes.

Command Mode

EXEC

Command Syntax

show ipv6 route [**ADDRESS**][**ROUTE_TYPE**][**INFO_LEVEL**]

Parameters

ADDRESS, when present, is always listed first. All other parameters can be placed in any order.

- **ADDRESS** filters routes by IPv6 address or prefix.
  - ***no parameter*** all routing table entries.
  - ***ipv6_address*** routing table entries matching specified IPv6 address.
  - ***ipv6_prefix*** routing table entries matching specified IPv6 prefix (CIDR notation).
- **ROUTE_TYPE** filters routes by specified protocol or origin.
  - ***no parameter*** all routing table entries.
  - **aggregate** entries for BGP aggregate routes.
  - **bgp** entries added through BGP protocol.
  - **connected** entries for routes to networks directly connected to the switch.
  - **kernel** entries appearing in Linux kernel but not added by EOS software.
  - **isis** entries added through IS-IS protocol.
  - **ospf** entries added through OSPF protocol.
  - ***static*** entries added through CLI commands.
- **INFO_LEVEL** Filters entries by next hop connection.
  - ***no parameter*** filters routes whose next hops are directly connected.
  - **detail** displays all routes.

Example

This command displays a route table entry for a specific IPv6 route.

```
switch> show ipv6 route fd7a:3418:52a4:fe18::/64
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:3418:52a4:fe18::/64 [10/20]
     via fe80::21c:73ff:fe00:1319, Vlan3601
     via fe80::21c:73ff:fe00:1319, Vlan3602
     via fe80::21c:73ff:fe00:1319, Vlan3608
```

```
     via fe80::21c:73ff:fe0f:6a80, Vlan3610
     via fe80::21c:73ff:fe00:1319, Vlan3611 switch>
```

13.2.4.43 show ipv6 route age

The **show ipv6 route age** command displays the IPv6 route age to the specified IPv6 address or prefix.

Command Mode

EXEC

Command Syntax

show ipv6 route ADDRESS age

Parameters

ADDRESS filters routes by IPv6 address or prefix.

- ***ipv6_address*** routing table entries matching specified address (A:B:C:D:E:F:G:H).
- ***ipv6_prefix*** routing table entries matching specified IPv6 prefix (A:B:C:D:E:F:G:H/PL).

Example

This command displays the route age for the specified prefix.

```
switch>show ipv6 route 2001::3:0/11 age
IPv6 Routing Table - 74 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

C 2001::3:0/11 age 00:02:34 switch>
```

13.2.4.44 show ipv6 route host

The **show ipv6 route host** command displays all host routes in the IPv6 host forwarding table. Host routes are those whose destination prefix is the entire address (prefix =***/128***). Each displayed host route is labeled with its purpose:

- **F** static routes from the FIB.
- **A** routes to any neighboring host for which the switch has an ARP entry.
- **R** routes defined because the IP address is an interface address.

Command Mode

EXEC

Command Syntax

show ipv6 route host

Example

This command displays all IPv6 host routes in the host forwarding table.

```
switch> show ipv6 route host
R - receive F - FIB, A - attached

 F  ::1 to cpu
 A  fee7:48a2:0c11:1900:400::1 on Vlan102
 R  fee7:48a2:0c11:1900:400::2 to cpu
 F  fee7:48a2:0c11:1a00::b via fe80::21c:73ff:fe0b:a80e on Vlan3902
 R  fee7:48a2:0c11:1a00::17 to cpu
 F  fee7:48a2:0c11:1a00::20 via fe80::21c:73ff:fe0b:33e on Vlan3913
 F  fee7:48a2:0c11:1a00::22 via fe80::21c:73ff:fe01:5fe1 on Vlan3908
```

```
                                    via fe80::21c:73ff:fe01:5fe1 on Vlan3902 switch>
```

13.2.4.45 show ipv6 route interface

The **show ipv6 route interface** command displays routing table entries on a specified routed port.

Command Mode

EXEC

Command Syntax

show ipv6 route [**ADDRESS**] **interface PORT_NAME** [**INFO_LEVEL**]

Parameters

ADDRESS, when present, is always listed first. All other parameters can be placed in any order.

- **ADDRESS** filters routes by IPv6 address or prefix.
  - ***no parameter*** all routing table entries.
  - ***ipv6_address*** routing table entries matching specified IPv6 address.
  - ***ipv6_prefix*** routing table entries matching specified IPv6 prefix (CIDR notation).
- **PORT_NAME** interfaces for which command displays status.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- **INFO_LEVEL** Filters entries by next hop connection.
  - ***no parameter*** filters routes whose next hops are directly connected.
  - **detail** displays all routes.

Example

This command displays the IPv6 routes in ***interface ethernet 8***.

```
switch> show ipv6 route interface ethernet 8
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:629f:63af:1232::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:4118::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:4119::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:411a::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:fe78::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
C   fd7a:629f:63af:fe88::/64 [0/1]
     via ::, Ethernet12
O   fd7a:629f:63af:fe8c::/64 [10/20]
     via fe80::21c:73ff:fe00:3640, Ethernet8
C   fe80:0:40::/64 [0/1]
     via ::, Ethernet8
```

13.2.4.46 show ipv6 route match tag

The **show ipv6 route match tag** command displays the route tag assigned to the specified IPv6 address or prefix. Route tags are added to static routes for use by route maps.

Command Mode

EXEC

Command Syntax

show ipv6 route ADDRESS **match tag**

Parameters

ADDRESS filters routes by IPv6 address or prefix.

- ***ipv6_address*** routing table entries matching specified address (A:B:C:D:E:F:G:H).
- ***ipv6_prefix*** routing table entries matching specified IPv6 prefix (A:B:C:D:E:F:G:H/PL).

Example

This command displays the route tag for the specified prefix.

```
switch> show ipv6 route 2001:0DB8::/64 match tag
IPv6 Routing Table - 74 entries
Codes: C - connected, S - static, K - kernel, O3 - OSPFv3, B - BGP, R - RIP, A
 B
- BGP Aggregate, I L1 - IS-IS level 1, I L2 - IS-IS level 2, DH - DHCP, NG -
Nexthop Group Static Route, M - Martian, DP - Dynamic Policy Route, L - VRF
 Leaked

C   2001:0DB8::/64 tag 0 switch>
```

13.2.4.47 show ipv6 route summary

The **show ipv6 route summary** command displays the information about the IPv6 routing table.

Command Mode

EXEC

Command Syntax

show ipv6 route summary

Example

This command displays the route source and the corresponding number of routes in the IPv6 routing table.

```
switch> show ipv6 route summary
Route Source    Number Of Routes
----------------- ---------------
   connected                    2
   static                       0
   ospf                         5
   bgp                          7
   isis                         0
   internal                     1
   attached                     0
   aggregate                    2

   Total Routes                 17 switch>
```

13.2.4.48 show platform fap mroute ipv6

The **show platform fap mroute ipv6** command enables PIM Sparse Mode (PIM-SM) and IGMP (router mode) on the configuration mode interface.

Command Mode

EXEC

Command Syntax

show platform

Example

This command enables PIM sparse mode on ***VLAN 4*** interface.

```
switch# show platform fap mroute ipv6
Jericho0 Multicast Routes:
--------------------------
Location  GroupId Group               Source          IIF      McId    OIF
FLP/TT    FLP/TT  TT                  FLP             FLP      FLP     FLP
-------------------------------------------------------------------------------------------------
4096/2048 1/1     ff33::1:0:0:23/128  101:1::2/128    Vlan1357 21504   Vlan1044(Et7/1) Vlan1123(Et9/1)
                                                                      Vlan1200(Et8/1) Vlan1223(Et2/1)
                                                                      Vlan1226(Et5/1) Vlan1232(Et3/1)
                                                                      Vlan1307(Et6/1) Vlan1337(Et4/1)
```

13.2.4.49 show rib route ipv6

The **show rib route ipv6** command displays a list of IPv6 Routing Information Base (RIB) routes.

Command Mode

EXEC

Command Syntax

show rib route ipv6 [vrf ***vrf_name***] [**PREFIX**][**ROUTE TYPE**]

Parameters

- **vrf *vrf_name*** displays RIB routes from the specified VRF.
- **PREFIX** displays routes filtered by the specified IPv6 information. Options include:
  - ***ipv6_address*** displays RIB routes filtered by the specified IPv6 address.
  - ***ipv6_subnet_mask*** displays RIB routes filtered by the specified IPv6 address and subnet mask.
  - ***ipv6_prefix*** displays RIB routes filtered by the specified IPv6 prefix.
- **ROUTE TYPE** displays routes filtered by the specified route type. Options include:
  - **bgp** displays RIB routes filtered by BGP.
  - **connected** displays RIB routes filtered by connected routes.
  - **dynamicPolicy** displays RIB routes filtered by dynamic policy routes.
  - **host** displays RIB routes filtered by host routes.
  - **isis** displays RIB routes filtered by ISIS routes.
  - **ospf** displays RIB routes filtered by OSPF routes.
  - **ospf3** displays RIB routes filtered by OSPF3 routes.
  - **reserved** displays RIB routes filtered by reserved routes.
  - **route-input** displays RIB routes filtered by route-input routes.
  - **static** displays RIB routes filtered by static routes.

Examples

- This command displays IPv6 RIB BGP routes.

```
switch# show rib route ipv6 bgp
```

```
VRF name: default, VRF ID: 0xfe, Protocol: bgp
Codes: C - Connected, S - Static, P - Route Input
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis
       > - Best Route, * - Unresolved Nexthop
       L - Part of a recursive route resolution loop
 B    2001:10:1::/64 [200/42]
         via 2001:10:1::100 [0/1]
            via Ethernet1, directly connected
>B    2001:10:100::/64 [200/200]
         via 2001:10:1::100 [0/1]
            via Ethernet1, directly connected
>B    2001:10:100:1::/64 [200/0]
         via 2001:10:1::100 [0/1]
            via Ethernet1, directly connected
>B    2001:10:100:2::/64 [200/42]
         via 2001:10:1::100 [0/1]
            via Ethernet1, directly connected switch#
```

- This command displays IPv6 RIB connected routes.

```
switch# show rib route ipv6 connected
VRF name: default, VRF ID: 0xfe, Protocol: connected
Codes: C - Connected, S - Static, P - Route Input
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis
       > - Best Route, * - Unresolved Nexthop
       L - Part of a recursive route resolution loop
>C    2001:10:1::/64 [0/1]
         via 2001:10:1::102, Ethernet1
>C    2001:10:2::/64 [0/1]
         via 2001:10:2::102, Ethernet2
>C    2001:10:3::/64 [0/1]
         via 2001:10:3::102, Ethernet3 switch#
```

13.3 Ingress and Egress Per-Port for IPv4 and IPv6 Counters

This feature provides support for per-interface ingress and egress packet and byte counters for both IPv4 and IPv6.

This section describes Ingress and Egress per-port for IPv4 and IPv6 counters, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Configuration
- Show Commands
- Dedicated ARP Entry for TX IPv4 and IPv6 Counters
- Limitations

13.3.1 Configuration

IPv4 and IPv6 ingress counters (count **bridged and routed** traffic, supported only on front-panel ports) can be enabled and disabled using the **`hardware counter feature ip in`** command:

```
[no] hardware counter feature ip in
```

For IPv4 and IPv6 ingress and egress counters that include only **routed** traffic (supported on Layer3 interfaces such as routed ports, L3 subinterfaces only):

On the DCS-7300X, DCS-7250X, DCS-7050X, and DCS-7060X platforms, IPv4 and IPv6 **packet** counters for only **routed** traffic do not require any configuration. They are collected by default. Other platforms (DCS-7280SR, DCS-7280CR and DCS-7500-R) need the feature enabled.

```
[no]hardware counter feature ip in layer3
```

```
[no]hardware counter feature ip out layer3
```

13.3.2 Show Commands

Use the **show interfaces counters ip** command to display IPv4, IPv6 packets, and octets.

```
switch# show interfaces counters ip
Interface   IPv4InOctets    IPv4InPkts     IPv6InOctets    IPv6InPkts
Et1/1               0               0                0             0
Et1/2               0               0                0             0
Et1/3               0               0                0             0
Et1/4               0               0                0             0
...
Interface  IPv4OutOctets  IPv4OutPkts    IPv6OutOctets   IPv6OutPkts

Et1/1               0               0                0             0
Et1/2               0               0                0             0
Et1/3               0               0                0             0
Et1/4               0               0                0             0
...
```

The output from the **show interfaces counters ip** can also be queried through SNMP via the ARISTA-IP-MIB.

To clear the the IPv4 or IPv6 counters, use the **clear counters** command

```
switch# clear counters
```

13.3.3 Dedicated ARP Entry for TX IPv4 and IPv6 Counters

IPv4/IPv6 egress Layer 3 (**hardware counter feature ip out layer3**) counting on DCS-7280SR, DCS-7280CR and DCS-7500-R platforms work, based on ARP entry of the nexthop. By default, IPv4 next hop and IPv6 next hop both resolve to the same MAC address and interface that have shared ARP entry. To differentiate the counters between IPv4 and IPv6, disable arp entry sharing with following command:

```
ip hardware fib next-hop arp dedicated
```

On the DCS-7280SR, DCS-7280CR and DCS-7500-R platforms, this command is required for IPv4 and IPv6 egress counters to operate.

13.3.4 Limitations

- Packet sizes greater than 9236 bytes are not counted by per-port IPv4 and IPv6 counters.
- Only the DCS-7260X3, DCS-7368, DCS-7300, DCS-7050SX3, DCS-7050CX3, DCS-7280SR, DCS-7280CR and DCS-7500-R platforms support the **`hardware counter feature ip in`** command.
- Only the DCS-7280SR, DCS-7280CR and DCS-7500-R platforms support the **`hardware counter feature ip [in|out] layer3`** command.

13.4 ACLs and Route Maps

The switch uses rule-based lists to control packet access to ports and to select routes for redistribution to routing domains defined by dynamic routing protocols. This section describes the construction of Access Control Lists (ACLs), prefix lists, and route maps.

This section includes the following topics:

- ACL, Service ACL, Route Map, Prefix List, and RACL Divergence Introduction
- Access Control Lists
- Service ACLs
- Sub-interface ACLs
- Egress ACL Counters
- RACL Sharing on SVIs
- Route Maps
- Prefix Lists
- Port ACLs with User-Defined Fields
- ACL, Route Map, and Prefix List Commands

13.4.1 ACL, Service ACL, Route Map, Prefix List, and RACL Divergence Introduction

Access Control Lists (ACLs), Service ACLs, route maps, and prefix lists are all processed in order, beginning with the first rule and proceeding until a match is encountered.

An Access Control List (ACL) is a list of rules that control the inbound flow of packets into Ethernet interfaces, subinterfaces, and port channel interfaces or the switch control plane. The switch supports the implementation of a wide variety of filtering criteria including IP and MAC addresses, TCP/UDP ports with include/exclude options without compromising its performance or feature set. Filtering syntax is industry standard.

A Service ACL is an ACL applied by a control-plane process to control connections to, or packets processed by, the agent process.

A route map is a list of rules that control the redistribution of IP routes into a protocol domain on the basis of such criteria as route metrics, access control lists, next hop addresses, and route tags. Route maps can also alter parameters of routes as they are redistributed.

A prefix list is a list of rules that defines route redistribution access for a specified IP address space. Route maps often use prefix lists to filter routes.

The RACL divergence optimizes the usage of hardware resources occupied on each forwarding ASIC by installing ACLs only on the hardware components corresponding to the member interfaces belonging to the SVIs on which ACL is applied. Hence, saving the hardware resources used and enables RACLs to scale-up to a larger configuration. The show commands are used to display the interface mapping, TCAM entries, and TCAM utilization information.

13.4.2 Access Control Lists

These sections describe access control lists:

- ACL Types
- ACL Configuration
- Applying ACLs

13.4.2.1 ACL Types

The switch supports the following ACL types:

- **IPv4** can match on IPv4 source or destination addresses, with L4 modifiers including protocol, port number, and DSCP value.
- **IPv6** can match on IPv6 source or destination addresses, with L4 modifiers including protocol, port number, etc.
- **Standard IPv4** can match only on source IPv4 address.
- **Standard IPv6** can match only on source IPv6 address.
- **MAC** can match on L2 source and destination addresses..

ACLs can also be made dynamic (not persisting in the EOS), and the **payload** keyword can be used to turn an ACL into a User-Defined Field (UDF) alias for use in other ACLs.

13.4.2.1.1 ACL Structure

An ACL is an ordered list of rules that defines access restrictions for the entities (the control plane, or an interface) to which it is applied. ACLs are also used by route maps to select routes for redistribution into specified routing domains.

ACL rules specify the data to which packet contents are compared when filtering data.

- The interface forwards packets that match all commands in a permit rule.
- The interface drops packets that match all commands in a deny rule.
- The interface drops packets that do not match at least one rule.

Upon its arrival at an interface, a packet's fields are compared to the first rule of the ACL applied to the interface. Packets that match the rule are forwarded (permit rule) or dropped (deny rule). Packets that do not match the rule are compared to the next rule in the list. This process continues until the packet either matches a rule or the rule list is exhausted. The interface drops packets not matching a rule.

The sequence number designates the rule's placement in the ACL.

13.4.2.1.2 ACL Rules

ACL rules consist of a command list that is compared to inbound packet fields. When all of a rule's criteria match a packet's contents, the interface performs the action specified by the rule.

The set of available commands depend on the ACL type and the specified protocol within the rule. The following is a list of commands available for supported ACL types

IPv4 ACL Rule Parameters

All rules in IPv4 ACLs include the following criteria:

- **Protocol**: The packet's IP protocol. Valid rule inputs include:
  - Protocol name for a limited set of common protocols.
  - Assigned protocol number for all IP protocols.
- **Source Address**: The packet's source IPv4 address. Valid rule inputs include:
  - A subnet address (CIDR or address-mask). Discontiguous masks are supported.
  - A host IP address (dotted decimal notation).
  - *any* to denote that the rule matches all source addresses.
- **Destination Address**: The packet's destination IP address. Valid rule inputs include:

- A subnet address (CIDR or address-mask). Discontiguous masks are supported.
- A host IP address (dotted decimal notation).
- *any* to denote that the rule matches all destination addresses.

All rules in IPv4 ACLs ***may*** include the following criteria:

- **Fragment**: Rules filter on the fragment bit.
- **Time-to-live**: Compares the TTL (time-to-live) value in the packet to a specified value. Valid in ACLs applied to the control plane. Validity in ACLs applied to the data plane varies by switch platform. Comparison options include:
  - **Equal:** Packets match if packet value equals statement value.
  - **Greater than:** Packets match if packet value is greater than statement value.
  - **Less than:** Packets match if packet value is less than statement value.
  - **Not equal:** Packets match if packet value does not equal statement value.

The availability of the following optional criteria depends on the specified protocol:

- **Source Ports / Destination Ports**: A rule filters on ports when the specified protocol supports IP address-port combinations. Rules provide one of these port filtering values:
  - ***any*** denotes that the rule matches all ports.
  - A list of ports that matches the packet port. Maximum list size is 10 ports.
  - Negative port list. The rule matches any port not in the list. Maximum list size is 10 ports.
  - Integer (lower bound): The rule matches any port with a number larger than the integer.
  - Integer (upper bound): The rule matches any port with a number smaller than the integer.
  - Range integers: The rule matches any port whose number is between the integers.
- **Flag bits**: Rules filter TCP packets on flag bits.
- **Message type**: Rules filter ICMP type or code.
- **Tracked**: Matches packets in existing ICMP, UDP, or TCP connections. Valid in ACLs applied to the control plane. Validity in ACLs applied to the data plane varies by switch platform.

IPv6 ACL Rule Parameters



Note: When calculating the size of ACLs, be aware that Arista switches install four rules in every IPv6 ACL so that ICMPv6 neighbor discovery packets bypass the default drop rule.

All rules in IPv6 ACLs include the following criteria:

- **Protocol**: All rules filter on the packet's IP protocol field. Rule input options include:
  - Protocol name for a limited set of common protocols.
  - Assigned protocol number for all IP protocols.
- **Source Address**: The packet's source IPv6 address. Valid rule inputs include:
  - An IPv6 prefix (CIDR). Discontiguous masks are supported.
  - A host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all addresses.
- **Destination Address**: The packet's destination IP address. Valid rule inputs include:
  - A subnet address (CIDR or address-mask). Discontiguous masks are supported.
  - A host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all addresses.

All rules in IPv6 ACLs ***may*** include the following criteria:

- **Fragment**: Rules filter on the fragment bit.
- ***HOP*** Compares the packet's hop-limit value to a specified value. Comparison options include:
  - **eq**: Packets match if hop-limit value equals statement value.

- **gt**: Packets match if hop-limit value is greater than statement value.
- **lt**: Packets match if hop-limit value is less than statement value.
- **neq**: Packets match if hop-limit value is not equal to statement value.

The availability of the following optional criteria depends on the specified protocol:

- **Source Ports / Destination Ports**: A rule filters on ports when the specified protocol supports IP address-port combinations. Rules provide one of these port filtering values:
  - ***any*** denotes that the rule matches all ports.
  - A list of ports that matches the packet port. Maximum list size is 10 ports.
  - Negative port list. The rule matches any port not in the list. Maximum list size is 10 ports.
  - Integer (lower bound): The rule matches any port with a number larger than the integer.
  - Integer (upper bound): The rule matches any port with a number smaller than the integer.
  - Range integers: The rule matches any port whose number is between the integers.
- **Flag bits**: Rules filter TCP packets on flag bits.
- **Message type**: Rules filter ICMP type or code.
- **Tracked**: Matches packets in existing ICMP, UDP, or TCP connections. Valid in ACLs applied to the control plane. Validity in ACLs applied to the data plane varies by switch platform.

Standard IPv4 and IPv6 ACL Rule Parameters



Note: When calculating the size of ACLs, be aware that Arista switches install four rules in every IPv6 ACL so that ICMPv6 neighbor discovery packets bypass the default drop rule.

Standard ACLs filter only on the source address.

MAC ACL Rule Parameters

MAC ACLs filter traffic on a packet's layer 2 header. Criteria that MAC ACLs use to filter packets include:

- **Source Address** and **Mask**: The packet's source MAC address. Valid rule inputs include:
  - MAC address range (address-mask in 3x4 dotted hexadecimal notation).
  - ***any*** to denote that the rule matches all source addresses.
- • **Destination Address** and **Mask**: The packet's destination MAC address. Valid rule inputs include:
  - MAC address range (address-mask in 3x4 dotted hexadecimal notation).
  - ***any*** to denote that the rule matches all destination addresses.
  - **Protocol**: The packet's protocol as specified by its EtherType field contents. Valid inputs include:
    - Protocol name for a limited set of common protocols.
    - Assigned protocol number for all protocols.

13.4.2.1.3 Creating and Modifying Lists

The switch provides configuration modes for creating and modifying ACLs. The command that enters an ACL configuration mode specifies the name of the list that the mode modifies. The switch saves the list to the running configuration when the configuration mode is exited.

- ACLs are created and modified in ACL configuration mode.
- Standard ACLs are created and modified in Standard-ACL-configuration mode.
- MAC ACLs are created and modified in MAC-ACL-configuration mode.

Lists that are created in one mode cannot be modified in any other mode.

A sequence number designates the rule's placement in a list. New rules are inserted into a list according to their sequence numbers. A rule's sequence number can be referenced when deleting it from a list.

ACL Configuration describes procedures for configuring ACLs.

13.4.2.1.4 Implementing Access Control Lists

An Access Control List (ACL) is implemented by assigning the list to an Ethernet interface or subinterface, to a port channel interface, or to the control plane. The switch assigns a default ACL to the control plane unless the configuration contains a valid control-plane ACL assignment statement. Ethernet and port channel interfaces are not assigned an ACL by default. Standard ACLs are applied to interfaces in the same manner as other ACLs.

IPv4 and MAC ACLs are separately applied for inbound and outbound packets. An interface or subinterface can be assigned multiple ACLs, with a limit of one ACL per packet direction per ACL type. Egress ACLs are supported on a subset of all available switches. The control-plane does not support egress ACLs.

Applying ACLs describes procedures for applying ACLs to interfaces or the control plane.

13.4.2.1.5 ACL Rule Tracking

ACL rule tracking determines the impact of ACL rules on the traffic accessing interfaces upon which they are applied. ACLs provide two tracking mechanisms:

- **ACL logging**: A syslog entry is logged when a packet matches specified ACL rules.
- **ACL counters**: ACL counters increment when a packet matches a rule in specified ACLs.

ACL Logging

ACL rules provide a **log** option that produces a log message when a packet matches the rule. ACL logging creates a syslog entry when a packet matches an ACL rule where logging is enabled. Packets that match a logging-enabled ACL rule are copied to the CPU by the hardware. These packets trigger the creation of a syslog entry. The information provided in the entry depends on the ACL type or the protocol specified by the ACL. Hardware rate limiting is applied to packets written to the CPU, avoiding potential DoS attacks. The rate of logging is also software limited to avoid the creation of syslog lists that are too large for practical use by human operators.

ACL Rule Tracking Configuration describes procedures for configuring and enabling ACL logging.

ACL Counters

An ACL counter is assigned to each ACL rule. The activity of the ACL counters for rules within a list depend on the list's counter state. When the list is in counting state, the ACL counter of a rule increments when the rule matches a packet. When the list is in a non-counting state, the counter does not increment. A list's counter state applies to all rules in the ACL. The default state for new ACLs is non-counting.

When an ACL changes from counting state to non-counting state, or when the ACL is no longer applied to any interfaces that increment counters, counters for all rules in the list maintain their values and do not reset. When the ACL returns to counting mode or is applied to an interface that increments counters, the counter operation resumes from its most recent value.

Counters never decrement and are reset only through CLI commands.

ACL Rule Tracking Configuration describes procedures for configuring and enabling ACL counters.

13.4.2.1.6 Egress ACL Counters

Egress ACL counters count the number of packets matching rules associated with egress ACLs applied to various interfaces in a switch. For 7050 and 7060 series switches, these counters are maintained for every TCAM rule; on these platforms, packet counters greater than zero are always shown by commands such as **`show platform trident tcam`**, **`show platform trident counters`**, and **`show ip access-list`**. For other switches, counters are not enabled by default and must be configured for each ACL, and the counters can be shown with the **`show hardware counter`** and **`show ip access-list`** commands.

13.4.2.1.6.1Configuring Egress ACL Counters

For 7050 and 7060 series switches, egress ACL counters are always enabled, and no configuration is required.

For other platforms, to enable egress ACL counters for a specific ACL, use the **counter per-entry** command in the configuration mode for the ACL.

Example

In the following example, configure the **counters per-entry** command in the ACL configuration mode.

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# counters per-entry
```

Enabling Egress Counters Globally

For 7050 and 7060 series switches, egress counters are always enabled.

For other switches, both IPv4 and IPv6 egress ACL counters are enabled in the global configuration mode by using the **hardware counter feature acl out** command.

Example

The following example shows how to enable IPv4 egress ACL counters.

```
switch(config)# hardware counter feature acl out ipv4
switch(config)#
```

The following example shows how to enable IPv6 egress ACL counters.

```
switch(config)# hardware counter feature acl out ipv6
switch(config)#
```

Disabling Egress Counters Globally

For 7050 and 7060 series switches, egress counters cannot be disabled.

For other switches, both IPv4 and IPv6 egress ACL counters are also disabled in the global configuration mode by using the **hardware counter feature acl out** command.

The following example shows how to disable IPv4 egress ACL counters.

```
switch(config)# no hardware counter feature acl out ipv4
switch(config)#
```

The following example shows how to disable IPv6 egress ACL counters.

```
switch(config)# no hardware counter feature acl out ipv6
switch(config)#
```

Egress Counter Roll Over in the Global Mode

The counters roll over when the counter value for an ACL rule exceeds ***2^64***.

Example

In the following example, the **hardware counter feature acl ipv6 out** command is configured using units and packets.

```
switch(config)# hardware counter feature acl ipv6 out units packets
```

```
switch(config)#
```

The **clear ip access-lists counters** command clears the counters for all of the IPv4 ACLs or a specific IPv4 ACL, either globally or per-CLI session.

Example

In the following example the ACL list named ***red*** is selected.

```
switch(config)# clear ip access-list counters red session
switch(config)#
```

The IPv6 egress ACL counters do not work in unshared mode.

Example

Use the **hardware access-lists resource sharing vlan ipv6 out** command to enable egress IPv6 ACL sharing.

```
switch(config)#hardware access-list resource sharing vlan ipv6 out
switch(config)#
```

The **clear ipv6 access-list counters** command clears the counters for all of the IPv6 ACLs or a specific IPv6 ACL, either globally or per-CLI session.

Example

In the following example the ACL list named ***green*** is selected.

```
switch(config)# clear ipv6 access-list counters green session
switch(config)#
```

13.4.2.1.6.2Displaying Egress ACL Counters

Use the following show commands to display Egress ACL Counters information.

Use the **show ip access-lists** command to display all the IPv4 ACLs, or a specific IPv4 ACL configured in a switch. The output contains details such as rules in an ACL and respective counter values with each rule.

```
switch(config)# show ip access-list acl1
IP Access List acl1
        counter per-entry
        10 deny ip 11.1.1.0/24 any dscp af11
        20 deny ip any any [match 39080716, 0:00:00 ago]
```

Use the **show ipv6 access-lists** command to display all the IPv6 ACLs or a specific IPv6 ACL configured in a switch. The output contains details such as rules in an ACL and respective counter values with each rule.

```
switch(config)# show ipv6 access-list acl1
IPV6 Access List acl1
        counter per-entry
        10 permit ipv6 any any [match 3450000, 0:00:10 ago]
        20 deny ipv6 any any
```

The counter name **EgressAclDropCounter** in the output of this show command signifies the aggregate counter value for the remaining egress IPv4 ACL. In this example the deny rules, whose per rule counters, are not allocated. The per rule counters is not allocated when the user does not configure the ***counter per-entry*** parameter for the respective ACL.

```
switch(config)# show hardware counter drop
```

```
Summary:
Total Adverse (A) Drops: 0
Total Congestion (C) Drops: 0
Total Packet Processor (P) Drops: 250
Type Chip CounterName : Count : First Occurrence : Last Occurrence
------------------------------------------------------------------------------
P Fap0 EgressAclDropCounter : 250 : 2015-11-11 22:39:02 : 2015-11-11 22:51:44
```

13.4.2.2 ACL Configuration

Access Control Lists (ACLs) are created and modified in an ACL-configuration mode. A list can be edited only in the mode where it was created. The switch provides five configuration modes for creating and modifying access control lists:

- **ACL configuration mode** for IPv4 access control lists.
- **IPv6-ACL configuration mode** for IPv6 access control lists.
- **Std-ACL configuration mode** for Standard IPv4 access control lists.
- **Std-IPv6-ACL configuration mode** for Standard IPv6 access control lists.
- **MAC-ACL configuration mode** for MAC access control lists.

These sections describe the creation and modification of ACLs:

- Managing ACLs
- Modifying an ACL
- ACL Rule Tracking Configuration
- Displaying ACLs
- Configuring Per-Port Per-VLAN QoS
- Displaying Per-Port Per-VLAN QoS
- Configuring Mirror Access Control Lists

13.4.2.2.1 Managing ACLs

Creating and Opening a List

To create an ACL, enter one of the following commands, followed by the name of the list:

- ip access-list for IPv4 ACLs.
- ipv6 access-list for IPv6 ACLs.
- ip access-list standard for standard IPv4 ACLs.
- ipv6 access-list standard for standard IPv6 ACLs.
- mac access-list for MAC ACLs.

The switch enters the appropriate ACL configuration mode for the list. If the command is followed by the name of an existing ACL, subsequent commands edit that list (see Modifying an ACL for additional information).

Examples

- This command places the switch in ***ACL*** configuration mode to create an ACL named ***test1***.

```
switch(config)# ip access-list test1
switch(config-acl-test1)#
```

- This command places the switch in Standard-ACL-configuration mode to create a Standard ACL named ***stest1***.

```
switch(config)# ip access-list standard stest1
```

```
switch(config-std-acl-stest1)#
```

- This command places the switch in ***MAC-ACL*** configuration mode to create an MAC ACL named ***mtest1***.

```
switch(config)# mac access-list mtest1
switch(config-mac-acl-mtest1)#
```

Saving List Modifications

ACL configuration modes are group-change modes. Changes made in a group-change mode are saved by exiting the mode. To exit the group-change mode, changes can also be discarded using the `abort` command instead of exit.

Examples

- These commands enter the first three rules into a new ACL.

```
switch(config-acl-test1)# permit ip 10.10.10.0/24 any
switch(config-acl-test1)# permit ip any host 10.20.10.1
switch(config-acl-test1)# deny ip host 10.10.10.1 host 10.20.10.1
```

- To view the edited list, type **`show`**.

```
switch(config-acl-test1)# show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Because the changes were not yet saved, the ACL remains empty, as shown by show ip access-lists.

```
switch(config-acl-test1)# show ip access-lists test1
switch(config-acl-test1)#
```

To save all current changes to the ACL and exit ACL configuration mode, type **`exit`**.

```
switch(config-acl-test1)# exit
switch(config)# show ip access-lists test1
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Discarding List Changes

The **`abort`** command exits ACL configuration mode without saving pending changes.

Examples

- These commands enter the first three rules into a new ACL.

```
switch(config-acl-test1)# permit ip 10.10.10.0/24 any
switch(config-acl-test1)# permit ip any host 10.20.10.1
switch(config-acl-test1)# deny ip host 10.10.10.1 host 10.20.10.1
```

- To view the edited list, type **`show`**.

```
switch(config-acl-test1)# show
```

```
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

To discard the changes, enter **abort**. If the ACL existed before entering ACL-configuration mode, **abort** restores the version that existed before entering ACL-configuration mode. Otherwise, show ip access-lists shows the ACL was not created.

```
switch(config-acl-test1)# abort
switch(config)#
```

13.4.2.2.2 Modifying an ACL

An existing ACL, including those currently applied to interfaces, can be modified by entering the appropriate configuration mode for the ACL as described in Creating and Opening a List. By default, while an ACL is being modified all traffic is blocked on any interface to which the ACL has been applied.

Permit All Traffic During ACL Update

Because blocking ports during ACL modifications can result in packet loss and can interfere with features such as routing and dynamic NAT, 7050X, 7060X, 7150, 7250X, 7280, 7280R, 7300X, 7320X, and 7500 series switches can be configured instead to permit all traffic on Ethernet and VLAN interfaces while ACLs applied to those interfaces are being modified. This is done with the hardware access-list update default-result permit command.

These commands add deny rules to the appropriate ACL:

- deny (IPv4 ACL) adds a deny rule to an IPv4 ACL.
- deny (IPv6 ACL) adds a deny rule to an IPv6 ACL.
- deny (Standard IPv4 ACL) adds a deny rule to an IPv4 standard ACL.
- deny (Standard IPv6 ACL) adds a deny rule to an IPv6 standard ACL.
- deny (MAC ACL) adds a deny rule to a MAC ACL.

These commands add permit rules to the appropriate ACL:

- permit (IPv4 ACL) adds a permit rule to an IPv4 ACL.
- permit (IPv6 ACL) adds a permit rule to an IPv6 ACL.
- permit (Standard IPv4 ACL) adds a permit rule to an IPv4 standard ACL.
- permit (Standard IPv6 ACL) adds a permit rule to an IPv6 standard ACL.
- permit (MAC ACL) adds a permit rule to a MAC ACL.

Adding a Rule

To append a rule to the end of a list, enter the rule without a sequence number while in ACL configuration mode for the list. The new rule's sequence number is derived by adding ***10*** to the last rule's sequence number.

Examples

- This command configures the switch to permit all traffic during ACL modifications on interfaces to which the ACL has been applied. The rules in modified ACLs are applied after exiting ACL configuration mode, and after the ACL rules have been populated in hardware.

```
switch(config)# hardware access-list update default-result permit
```

- These commands enter the first three rules into a new ACL.

```
switch(config-acl-test1)# permit ip 10.10.10.0/24 any
switch(config-acl-test1)# permit ip any host 10.20.10.1
switch(config-acl-test1)# deny ip host 10.10.10.1 host 10.20.10.1
```

- To view the edited list, type **show**.

```
switch(config-acl-test1)# show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
```

- This command appends a rule to the ACL. The new rule's sequence number is ***40***.

```
switch(config-acl-test1)# permit ip any any
switch(config-acl-test1)# show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Inserting a Rule

To insert a rule into a ACL, enter the rule with a sequence number between the existing rules' numbers.

Example

This command inserts a rule between the first two rules by assigning it the sequence number ***15***.

```
Switch(config-acl-test1)# 15 permit ip 10.30.10.0/24 host 10.20.10.1
Switch(config-acl-test1)# show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        15 permit ip 10.30.10.0/24 host 10.20.10.1
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Deleting a Rule

To remove a rule from the current ACL, perform one of these commands:

- Enter **no**, followed by the sequence number of the rule to be deleted.
- Enter **no**, followed by the rule be deleted.
- Enter **default**, followed by the rule to be deleted.

Examples

- These equivalent commands remove rule ***20*** from the list.

```
switch(config-acl-test1)# no 20
switch(config-acl-test1)# no permit ip any host 10.20.10.1
switch(config-acl-test1)# default permit ip any host 10.20.10.1
```

- This ACL results from entering one of the preceding commands.

```
switch(config-acl-test1)# show
```

```
ip access list test1
        10 permit ip 10.10.10.0/24 any
        15 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Resequencing Rule Numbers

Sequence numbers determine the order of the rules in an access control list. After a list editing session where existing rules are deleted and new rules are inserted between existing rules, the sequence number distribution may not be uniform. Resequencing rule numbers changes the sequence number of rules to provide a constant difference between adjacent rules. The resequence (ACLs) command adjusts the sequence numbers of ACL rules.

Example

The resequence (ACLs) command renumbers rules in the test1 ACL. The sequence number of the first rule is ***100***; subsequent rules numbers are incremented by ***20***.

```
switch(config-acl-test1)# show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        25 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        50 permit ip any any
        90 remark end of list
switch(config-acl-test1)# resequence 100 20
switch(config-acl-test1)# show
IP Access List test1
        100 permit ip 10.10.10.0/24 any
        120 permit ip any host 10.20.10.1
        140 deny ip host 10.10.10.1 host 10.20.10.1
        160 permit ip any any
        180 remark end of list
```

13.4.2.2.3 ACL Rule Tracking Configuration

ACL rules provide a **log** option that produces a syslog message about the packets matching packet. ACL logging creates a syslog entry when a packet matches an ACL rule with logging enabled.

This feature is currently available on Arad switches and on 7100 series switches. On 7100 series switches, matches are logged only on ingress, not on egress.

Example

This command creates an ACL rule with logging enabled.

```
switch(config-acl-test1)# 15 permit ip 10.30.10.0/24 host 10.20.10.1 log
switch(config-acl-test1)#
```

The format of the generated Syslog message depends on the ACL type and the specified protocol:

- Messages generated by a TCP or UDP packet matching an IP ACL use this format:

  IPACCESS: list *acl intf filter protocol src-ip*(*src_port*) **->** *dst-ip*(*dst_port*)
- Messages generated by ICMP packets matching an IP ACL use this format:

  IPACCESS: list *acl intf filter* **icmp** *src-ip*(*src-port*) **->** *dst-ip*(*dst-port*) **type=** *n* **code=** *m*
- Messages generated by all other IP packets matching an IP ACL use this format:

  IPACCESS: list *acl intf filter protocol src-ip* **->** *dst-ip*

- Messages generated by packets matching a MAC ACL use this format:

  MACACCESS: list *acl intf filter vlan ether src_mac* **->** *dst_mac*
- Messages generated by a TCP or UDP packet matching a MAC ACL use this format:

  MACACCESS: list *acl intf filter vlan ether ip-prt src-mac src-ip* **:** *src-prt* **->** *dst-mac dst-ip* **:** *dst-prt*
- Messages generated by any other IP packet matching a MAC ACL use this format:

  MACACCESS: list *acl intf filtervlan ether src_mac src_ip* **->** *dst_mac dst_ip*

Variables in the Syslog messages display the following values:

- **acl** Name of ACL.
- **intf** Name of interface that received the packet.
- **filter** Action triggered by ACL (**denied** or **permitted**).
- **protocol** IP protocol specified by packet.
- **vlan** Number of VLAN receiving packet.
- **ether** EtherType protocol specified by packet.
- **src-ip** and **dst-ip** source and destination IP addresses.
- **src-prt** and **dst-prt** source and destination ports.
- **src-mac** and **dst-mac** source and destination MAC addresses.

ACLs provide a command that configures its counter state (counting or non-counting). The counter state applies to all rules in the ACL. The initial state for new ACLs is non-counting.

The counters per-entry (ACL configuration modes) command places the ACL in counting mode.

This command places the configuration mode ACL in counting mode.

```
switch(config-acl-test1)# counters per-entry
switch(config-acl-test1)#exit
switch(config-acl-test1)#show ip access-list test1
IP Access List test1
        counters per-entry
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

The clear ip access-lists counters and clear ipv6 access-lists counters commands set the IP access list counters to zero for the specified IP access list.

This command clears the ACL counter for the test1 ACL.

```
switch(config)# clear ip access-lists counters test1
switch(config)#
```

13.4.2.2.4 Displaying ACLs

ACLs can be displayed by a **`show running-config`** command. The show ip access-lists also displays ACL rosters and contents, as specified by command parameters.

When editing an ACL, the show (ACL configuration modes) command displays the current or pending list, as specified by command parameters.

Displaying a List of ACLs

To display the roster of ACLs on the switch, enter show ip access-lists with the **summary** option.

Example

This command lists the available access control lists.

```
switch(config)# show ip access-list summary
IPV4 ACL default-control-plane-acl
        Total rules configured: 12
        Configured on: control-plane
        Active on    : control-plane

IPV4 ACL list2
        Total rules configured: 3

IPV4 ACL test1
        Total rules configured: 6

IPV4 ACL test_1
        Total rules configured: 1

IPV4 ACL test_3
        Total rules configured: 0
switch(config)#
```

Displaying Contents of an ACL

These commands display ACL contents.

- show ip access-lists
- show ipv6 access-lists
- show mac access-lists

Each command can display the contents of one ACL or of all ACLs of the type specified by the command:

- To display the contents of one ACL, enter show ip access-lists followed by the name of the ACL.
- To display the contents of all ACLs on the switch, enter the command without any options.

ACLs that are in counting mode display the number of inbound packets each rule in the list matched and the elapsed time since the last match.

Examples

- This command displays the rules in the **default-control-plane-acl** ACL.

```
switch# show ip access-lists default-control-plane-acl
IP Access List default-control-plane-acl [readonly]
        counters per-entry
        10 permit icmp any any
        20 permit ip any any tracked [match 1725, 0:00:00 ago]
        30 permit ospf any any
        40 permit tcp any any eq ssh telnet www snmp bgp https
        50 permit udp any any eq bootps bootpc snmp [match 993, 0:00:29 ago]
        60 permit tcp any any eq mlag ttl eq 255
        70 permit udp any any eq mlag ttl eq 255
        80 permit vrrp any any
        90 permit ahp any any
        100 permit pim any any
        110 permit igmp any any [match 1316, 0:00:23 ago]
        120 permit tcp any any range 5900 5910
```

- This command displays the rules in all ACLs on the switch.

```
switch# show ip access-lists
IP Access List default-control-plane-acl [readonly]
```

```
        counters per-entry
        10 permit icmp any any
        20 permit ip any any tracked [match 1371, 0:00:00 ago]
        30 permit ospf any any
        40 permit tcp any any eq ssh telnet www snmp bgp https
        50 permit udp any any eq bootps bootpc snmp
        60 permit tcp any any eq mlag ttl eq 255
        70 permit udp any any eq mlag ttl eq 255
        80 permit vrrp any any
        90 permit ahp any any
        100 permit pim any any
        110 permit igmp any any [match 1316, 0:00:23 ago]
        120 permit tcp any any range 5900 5910

IP Access List list2
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 permit ip any host 10.20.10.1
        40 deny ip host 10.10.10.1 host 10.20.10.1
        50 permit ip any any

IP Access List test1
Switch(config)#
```

Displaying ACL Modifications

While editing an ACL in ACL-configuration mode, the show (ACL configuration modes) command provides options for displaying ACL contents.

- To display the list, as modified in ACL configuration mode, enter **`show`** or **`show pending`**.
- To display the list, as stored in ***running-config***, enter **`show active`**.
- To display differences between the pending list and the stored list, enter **`show diff`**.

Examples

The examples in this section assume these ACL commands were previously entered.

These commands are stored in the configuration:

```
10 permit ip 10.10.10.0/24 any
20 permit ip any host 10.21.10.1
30 deny ip host 10.10.10.1 host 10.20.10.1
40 permit ip any any
50 remark end of list
```

The current edit session removed this command. This change is not yet stored to ***running-config***:

```
20 permit ip any host 10.21.10.1
```

The current edit session added these commands ACL. They are not yet stored to ***running-config***:

```
20 permit ip 10.10.0.0/16 any
25 permit tcp 10.10.20.0/24 any
45 deny pim 239.24.124.0/24 10.5.8.4/30
```

This command displays the pending ACL, as modified in ***ACL*** configuration mode.

```
switch(config-acl-test_1)# show pending
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.10.0.0/16 any
```

```
        25 permit tcp 10.10.20.0/24 any
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        45 deny pim 239.24.124.0/24 10.5.8.4/30
        50 remark end of list
```

This command displays the ACL, as stored in the configuration.

```
switch(config-acl-test_1)#show active
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.21.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

This command displays the difference between the saved and modified ACLs.

- Rules added to the pending list are denoted with a plus sign (+).
- Rules removed from the saved list are denoted with a minus sign (-).

```
switch(config-acl-test_1)#show diff
---
+++
@@ -1,7 +1,9 @@
 IP Access List test_1
         10 permit ip 10.10.10.0/24 any
-        20 permit ip any host 10.21.10.1
+        20 permit ip 10.10.0.0/16 any
+        25 permit tcp 10.10.20.0/24 any
         30 deny ip host 10.10.10.1 host 10.20.10.1
         40 permit ip any any
+        45 deny pim 239.24.124.0/24 10.5.8.4/30
```

13.4.2.2.5 Configuring Per-Port Per-VLAN QoS

To configure per-port per-VLAN QoS, first, configure the ACL policing for QoS, and then apply the policy-map on a single Ethernet or port-channel interfaces on a per-port per-VLAN basis. The per port per VLAN QoS allows a class-map to match traffic for a single VLAN or for a range of VLANs separated by commas. Per-port per-VLAN works with QoS-based class-maps only.

To configure per-port per-VLAN QoS on DCS-7280(R) and DCS-7500(R), change the TCAM profile to QoS as shown below.

1. Change the TCAM profile to QoS.

```
switch# config
switch(config)# hardware tcam profile qos
```

2. Create an ACL and then match the traffic packets based on the VLAN value and the VLAN mask configured in the ACL.

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# permit vlan 100 0xfff ip any any
switch(config-acl-acl1)# exit
```

3. Similarly, create a class-map and then match the traffic packets based on the range of VLAN values configured in the class-map.

```
switch(config)# class-map match-any class1
switch(config-cmap-qos-class1)# match vlan 20-40, 1000-1250, 2000
```

```
switch(config-cmap-qos-class1)# exit
```

13.4.2.2.6 Displaying Per-Port Per-VLAN QoS

The following show commands display the status, traffic hit counts, tcam profile information, and policy-maps configured on an interface.

The **show policy-map** command displays the policy-map information of the configured policy-map.

Examples

- ```
  switch# show policy-map policy1
  Service-policy policy1
  Class-map: class1 (match-any)
  Match: ip access-group name acl1
  Police cir 512000 bps bc 96000 bytes
  Class-map: class-default (match-any)
  ```
- The **show policy-map interface** command displays the policy-map configured on an interface.

  ```
  switch# show policy-map interface ethernet 1
  Service-policy input: p1
  Hardware programming status: Successful
  Class-map: c2001 (match-any)
  Match: vlan 2001 0xfff
  set dscp 4
  Class-map: c2002 (match-any)
  Match: vlan 2002 0xfff
  set dscp 8
  Class-map: c2003 (match-any)
  Match: vlan 2003 0xfff
  set dscp 12
  ```

13.4.2.2.7 Configuring Mirror Access Control Lists

Access Control Lists (ACLs) are configured to permit or deny traffic between source and destination ports on Strata-based platforms. Mirror ACLs are used in mirroring traffic by matching VLAN ID of the configured ACLs. Mirror ACLs are applied for IPv4, IPv6, and MAC ACLs.

Note: Mirror ACLs work in receiving direction only.

Examples

- These commands configure ACL to permit VLAN traffic between any source and destination host.

  ```
  switch(config)# ip access-list acl1
  switch(config-acl-acl1)# permit vlan 1234 0x0 ip any any
  ```
- These commands configure monitor session ***sess1*** with ***Ethernet 1*** as source port and ***Ethernet 2*** as destination port for an ingress ip ***acl_1***.

  ```
  switch(config)# monitor session sess1 source ethernet 1 rx ip access-group
   acl1
  switch(config)# monitor session sess1 destination ethernet 2
  ```

13.4.2.3 Applying ACLs

Access Control Lists become active when they are assigned to an interface or subinterface or to the control plane. This section describes the process of adding and removing ACL interface assignments.
  ```

Applying an ACL to an Interface

The switch must be in interface configuration mode to assign an ACL to an interface or subinterface.

- The ip access-group command applies the specified IP or standard IP ACL to the configuration mode interface or subinterface.
- The ip access-group command applies the specified IP or standard IP ACL to the conntrol-plane traffic.
- The mac access-group command applies the specified MAC ACL to the configuration mode interface.

IPv4, IPv6, and MAC ACLs are separately applied for inbound and outbound packets. An interface or subinterface can be assigned with multiple ACLs, with a limit of one ACL per packet direction per ACL type. Egress ACLs are supported on a subset of all available switches. IPv6 egress ACLs have limited availability, and IPv6 egress ACLs applied to routed interfaces or subinterfaces across the same chip on the DCS-7500E and the DCS-7280E series can be shared. In addition to that, the DSCP value can match on IPv6 egress ACLs. This result in a more efficient utilization of system resources, and is particularly useful for environments with few, potentially large, IPv6 egress ACLs applied across multiple routed interfaces.

Examples

- These commands assign ***test1*** ACL to ***interface ethernet 3***, then verify the assignment.

```
switch(config)# interface ethernet 3
switch(config-if-Et3)# ip access-group test1 in
switch(config-if-Et3)# show running-config interfaces ethernet 3
interface Ethernet3
   ip access-group test1 in
switch(config-if-Et3)#
```

- These commands place the switch in control plane configuration mode and applies the ACL assignment to the control-plane traffic.

```
switch(config)# control-plane
switch(config-cp)# ip access-group test_cp in
```

- This command enables shared ACLs.

```
switch(config)# hardware access-list resource sharing vlan ipv6 out
switch(config)#
```

- This command disables shared ACLs.

```
switch(config)# no hardware access-list resource sharing vlan ipv6 out
switch(config)#
```

- These commands apply an IPv4 ACL named ***test_ACL*** to ingress traffic on ***interface ethernet 5.1***.

```
switch(config)# interface ethernet 5.1
switch(config-if-Et5.1)# ipv4 access-group test_ACL in
switch(config-if-Et5.1)#
```

Removing an ACL from an Interface

The **`no ip access-group`** command removes an IP ACL assignment statement from ***running-config*** for the configuration mode interface. After an ACL is removed, the interface is not associated with an IP ACL.

The **`no mac ip access-group`** command removes a MAC ACL assignment statement from ***running-config*** for the configuration mode interface. After a MAC ACL is removed, the interface is not associated with an MAC ACL.

To remove an ACL from the control plane, enter the **no ip access-group** command in control plane configuration mode. Removing the control plane ACL command from ***running-config*** reinstates **default-control-plane-acl** as the control plane ACL.

Examples

- These commands remove the assigned IPv4 ACL from ***interface ethernet 3***.

```
switch(config)# interface ethernet 3
switch(config-if-Et3)# no ip access-group test in
switch(config-if-Et3)#
```

- These commands place the switch in control plane configuration mode and remove the ACL assignment from ***running-config***, restoring **default-control-plane-acl** as the control plane ACL.

```
switch(config)# control-plane
switch(config-cp)# no ip access-group test_cp in
switch(config-cp)#
```

13.4.3 Service ACLs

These sections describe Service ACLs:

- Service Access Control List Description
- Configuring Service ACLs and Displaying Status and Counters

13.4.3.1 Service Access Control List Description

Service ACL enforcement is a feature added to a control plane service (the SSH server, the SNMP server, routing protocols, etc) that allows the switch administrator to restrict the processing of packets and connections by the control plane processes that implement that service. The control plane program run by the control plane process checks already received packets and connections against a user configurable Access Control List (ACL), a Service ACL. The Service ACL contains permit and deny rules matching any of the source address, destination address, and TCP or UDP ports of received packets or connections. After receiving a packet or connection, the control plane process evaluates the packet or connection against the rules of the Service ACL configured for the control plane process, and if the received packet or connection matches a deny rule the control plane process drops or closes it without further processing.

Control Plane Process Enforced Access Control enables the system administrator to restrict which systems on the network can access the services provided by the switch. Each service has its own access control list, giving the system administrator fine grained control over access to the switch's control plane services. The CLI for this uses the familiar pattern of access control lists assigned for a specific purpose, in this case for each control plane service.

13.4.3.2 Configuring Service ACLs and Displaying Status and Counters

13.4.3.2.1 SSH Server

To apply the SSH server Service ACLs for IPv4 and IPv6 traffic, use the ip access-group (Service ACLs) and ipv6 access-group (Service ACLs) commands in ***mgt-ssh*** configuration mode as shown below.

```
switch(config)# management ssh
switch(config-mgmt-ssh)# ip access-group <acl_name> [vrf
 <vrf_name>] inswitch(config-mgmt-ssh)#ipv6 access-group <acl_name> [vrf
 <vrf_name>] in
```

In ***Release EOS-4.19.0***, all VRFs are required to use the same SSH server Service ACL. The Service ACL assigned without the **vrf** keyword is applied to all VRFs where the SSH server is enabled.

To display the status and counters of the SSH server Service ACLs, use the following commands.

```
switch> show management ssh ip access-list
switch> show management ssh ipv6 access-list
```

13.4.3.2.2 SNMP Server

To apply the SNMP server Service ACLs to restrict which hosts can access SNMP services on the switch, use the snmp-server community command as shown below.

snmp-server community *community-name* [view ***viewname***] [ro | rw] ***acl_name***

snmp-server community *community-name* [view ***viewname***] [ro | rw] ipv6 ***ipv6_acl_name***

13.4.3.2.3 EAPI

To apply Service ACLs to the EOS application programming interface (EAPI) server, use the ip access-group (Service ACLs) and ipv6 access-group (Service ACLs) commands as shown below.

```
switch(config)# management api http-commands
switch(config-mgmt-api-http-cmds)# vrf <vrf_name>
switch(config-mgmt-api-http-cmds-vrf-<vrf>)# ip access-group <acl_name>
switch(config-mgmt-api-http-cmds-vrf-<vrf>)# ipv6 access-group <ipv6_acl_name>
```

To display the status and counters of the EAPI server Service ACLs, use the following commands.

```
switch> show management api http-commands ip access-list
switch> show management api http-commands ipv6 access-list
```

13.4.3.2.4 BGP

To apply Service ACLs for controlling connections to the BGP routing protocol agent, use the ip access-group (Service ACLs) and ipv6 access-group (Service ACLs) commands as shown below.

```
switch(config)# router bgp <asn>
switch(config-router-bgp)# ip access-group <acl_name>
switch(config-router-bgp)# ipv6 access-group <ipv6_acl_name>
switch(config-router-bgp)# vrf <vrf_name>
switch(config-router-bgp-vrf-<vrf>)# ip access-group <acl_name>
switch(config-router-bgp-vrf-<vrf>)# ipv6 access-group <ipv6_acl_name>
```

To display the status and counters of the BGP routing protocol Service ACLs, use the following commands.

```
switch> show bgp ipv4 access-list
switch> show bgp ipv6 access-list
```

13.4.3.2.5 UCMP auto adjust for BGP

UnequalCcost Multi-Path (UCMP) for BGP forwards traffic based on weight assignments for next hops of routes of ECMP traffic. The weights are programmed in the FIB. By disseminating BGP link-bandwidth extended community attribute information with BGP routes, the receiver device of all routes, programs the next hops in the FIB using the received link-bandwidth values. The percentage of interface speed is appended to the received link bandwidth extended community value of the route. The weight ratio of the traffic sent over egress ports is adjusted to forward more traffic towards the peer with higher interface speed.

Configuring UCMP auto adjust for BGP

The following command enables the weight adjustment.

This command configures the adjust auto to ***62.3*** percent.

```
switch(config-router-bgp)# neighbor group1 link-bandwidth adjust auto
 percent 62.3
```

PERCENT is a float value between ***0.0*** to ***100.0*** and is optional.

13.4.3.2.6 OSPF

To apply Service ACLs for controlling packets processed by the OSPF routing protocol agent, use the ip access-group (Service ACLs) and ipv6 access-group (Service ACLs) commands as shown below.

```
switch(config)# router ospf <id>
switch(config-router-ospf)# ip access-group <acl_name>
switch(config-router-ospf)# ipv6 access-group <ipv6_acl_name>
```

When using VRFs, each per-VRF OSPF instance must be assigned its Service ACL explicitly.

To display the status and counters of the OSPF routing protocol Service ACLs, use the following commands.

```
switch> show ospf ipv4 access-list
switch> show ospf ipv6 access-list
```

13.4.3.2.7 PIM

To apply Service ACLs for controlling packets processed by the PIM routing protocol agent, use the **access-group** command as shown below.

```
switch(config)# router pim
switch(config-router-pim)# ipv4
switch(config-router-pim-ipv4)# access-group <acl_name>
switch(config-router-pim-ipv4)# vrf <vrf_name>
switch(config-router-pim-vrf-<vrf>)# ipv4
switch(config-router-pim-vrf-<vrf>-ipv4)# access-group <acl_name>
```

To display the status and counters of the PIM routing protocol Service ACLs, use the following commands.

```
switch> show ip pim access-list
```

13.4.3.2.8 IGMP

To apply Service ACLs for controlling packets processed by the IGMP management protocol agent, use the **ip igmp access-group** command as shown below.

```
switch(config)# router igmp
switch(config-router-igmp)# ip igmp access-group <acl_name>
switch(config-router-igmp)# vrf <vrf_name>
switch(config-router-igmp-vrf-<vrf>)# ip igmp access-group <acl_name>
```

To display the status and counters of the IGMP management protocol Service ACLs, use the following commands.

```
switch> show ip igmp access-list
```

13.4.3.2.9 DHCP Relay

To apply Service ACLs for controlling packets processed by the DHCP relay agent, use the **ip dhcp relay access-group** and **ipv6 dhcp relay access-group** commands as shown below.

```
switch(config)# ip dhcp relay access-group <acl_name> [vrf <vrf_name>]
switch(config)# ipv6 dhcp relay access-group <acl_name> [vrf <vrf_name>]
```

To display the status and counters of the DHCP relay agent Service ACLs, use the following commands.

```
switch> show ip dhcp relay access-list
switch> show ipv6 dhcp relay access-list
```

13.4.3.2.10 LDP

To apply Service ACLs for controlling packets and connections processed by the LDP MPLS label distribution protocol, use the command as shown below.

ip access-group (Service ACLs)

```
switch(config)# mpls ldp
switch(config-mpls-ldp)# ip access-group <acl_name>
```

To display the status and counters of the LDP Service ACLs, use the following command.

```
switch> show mpls ldp access-list
```

13.4.3.2.11 LANZ

To apply Service ACLs for controlling connections accepted by the LANZ agent, use the ip access-group (Service ACLs) and ipv6 access-group (Service ACLs) commands as shown below.

```
switch(config)# queue-monitor streaming
switch(config-qm-streaming)# ip access-group <acl_name>
switch(config-qm-streaming)# ipv6 access-group <ipv6_acl_name>
```

To display the status and counters of the LDP Service ACLs, use the following command.

```
switch> show queue-monitor streaming access-lists
```

13.4.3.2.12 MPLS Ping and Traceroute

To apply Service ACLs for controlling connections accepted by the MPLS Ping agent, use the ip access-group (Service ACLs) and ipv6 access-group (Service ACLs) commands as shown below.

```
switch(config)# mpls ping
switch(config-mpls-ping)# ip access-group <acl_name> [vrf <vrf_name>]
switch(config-mpls-ping)# ipv6 access-group <ipv6_acl_name> [vrf <vrf_name>]
```

13.4.3.2.13 Telnet Server

To apply Service ACLs to the Telnet server, use the ip access-group (Service ACLs) and ipv6 access-group (Service ACLs) commands as shown below.

```
switch(config)# management telnet
switch(config-mgmt-telnet)# ip access-group <acl_name> [vrf <vrf_name>] in
switch(config-mgmt-telnet)# ipv6 access-group <ipv6_acl_name> [vrf <vrf_name>]
 in
```

In ***EOS 4.19.0***, all VRFs are required to use the same Telnet server Service ACL. The Service ACL assigned without the **vrf** keyword is applied to all VRFs where the Telnet server is enabled.

To display the status and counters of the LDP Service ACLs, use the following commands.

```
switch> show management telnet ip access-list
switch> show management telnet ipv6 access-list
```

13.4.4 Sub-interface ACLs

This Sub-interface ACLs feature enables ACL functionality on subinterfaces.

13.4.4.1 Configuring Sub-interface ACLs

Configure the ACLs on subinterfaces, use the following command:

ip|ipv6 access-group ***acl-name*** in | out

To unconfigure the ACLs on subinterfaces, use the following command:

no ip|ipv6 access-group in | out

13.4.4.2 Sub-interface ACLs Limitations

The sub-interface ACLs feature contains the following limitations:

- Egress IPv4 ACLs on subinterfaces are not supported when sharing mode is disabled for Egress IPv4 RACLs.
- Egress IPv6 ACL deny logging is not supported on subinterfaces.
- Blocking traffic while modifying ACLs is not supported on Egress IPv4 ACLs on subinterfaces.

13.4.4.3 Sub-interface ACLs Show Commands

The **show ip|ipv6 access-lists** displays the summary of a configured ACL including the subinterface on which the ACL is configured and active.

show ip|ipv6 access-lists ***acl-name*** summary

Examples

```
switch(config)# show ip access-lists acl1 summary
IPV4 ACL acl1
 Total rules configured: 1
 Configured on Ingress: Et5.1
 Active on Ingress: Et5.1
```

```
switch(config)# show ipv6 access-lists acl1 summary
IPV6 ACL acl1
 Total rules configured: 1
 Configured on Egress: Et5.1
```

```
Active on Egress: Et5.1
```

13.4.5 RACL Sharing on SVIs

13.4.5.1 IPv4 Ingress Sharing

The IPv4 ingress sharing optimizes the utilization of hardware resources by sharing the hardware resources between different VLAN interfaces when they have same ACL attached.

Larger deployments are benefited with this function, where IPv4 ingress sharing is applied on multiple SVIs with member interfaces on same forwarding ASIC. For example, a trunk port carrying multiple VLANs and an ingress sharing is applied on all VLANs, it occupies lesser hardware resources irrespective of number of VLANs. By default, IPv4 ingress sharing is disabled on the switches.

To enable IPv4 Ingress Sharing use **`no hardware access-list resource sharing vlan in`** command. Note, enabling or disabling the IPv4 ingress sharing requires the restart of software agents on the switches which is a disruptive process and will impact the traffic forwarding. The **no** form of the command disables the IPv4 ingress sharing on the switch. To display the IPv4 ingress sharing information use **`show platform trident`** command on the switch.

13.4.5.2 IPv4 Egress Sharing

The IPv4 Egress Sharing optimizes the utilization of hardware resources by sharing TCAM entries for a group of SVIs on which IPv4 ACLs shared. The TCAM entries are shared for all the SVIs per chip, hence, saving a lot of hardware resources and enabling ACLs to scale to a larger configurations.

Larger deployments are benefited, where IPv4 Egress Sharing is applied on multiple SVIs with member interfaces on same forwarding ASIC. For example, a trunk port carrying multiple VLANs, and when Egress Sharing is applied on all VLANs it occupies lesser hardware resources irrespective of number of VLANs. By default, IPv4 Egress Sharing is enabled on the switches. However, both IPv4 Egress Sharing and uRPF cannot be enabled at the same time. Disabling IPv4 RACL sharing will allow uRPF configuration and make sure RACL configuration, non-shared mode, is configured at the same time.

To enable unicast Reverse Path Forwarding (uRPF) on the switche, the IPv4 Egress Sharing must me disabled using the **`no hardware access-list resource sharing vlan ipv4 out`** command.

To enable IPv4 Egress Sharing if previously disabled from the default configuration, use hardware access-list resource sharing vlan ipv4 out command. Note, enabling or disabling the IPv4 Egress Sharing requires the restart of software agents on the switches which is a disruptive process and will impact the traffic forwarding.

The following **`show`** commands can be used to verify the IPv4 Egress Sharing information on the switch.

- show ip access-lists
- show vlan
- show platform arad acl tcam
- show ip route
- show platform arad ip route

13.4.5.3 Configuring IPv4 Egress Sharing

Use **`hardware access-list resource sharing vlan ipv4 out`** command to enable the IPv4 Egress Sharing on the switch. By default, IPv4 Egress Sharing is enabled on the switche.The **no** form of the command disables the IPv4 Egress Sharing on the switch and user is allowed to configure the uRPF on the switch.

13.4.5.4 Displaying IPv4 Egress Sharing Information

Examples

- The show ip access-lists command displays the list of all the configured IPv4 ACLs.

```
switch# show ip access-lists summary
IPV4 ACL default-control-plane-acl [readonly]
 Total rules configured: 17
 Configured on Ingress: control-plane(default VRF)
 Active on Ingress: control-plane(default VRF)

IPV4 ACL ipAclLimitTest
 Total rules configured: 0
 Configured on Egress: Vl2148,2700
 Active on Egress: Vl2148,2700
```

- The **show vlan** command displays the list of all the member interfaces under each SVI.

```
switch# show vlan
VLAN  Name            Status    Ports
----- --------------- --------- -----------------
1     default         active
2148  VLAN2148        active    Cpu, Et1, Et26
2700  VLAN2700        active    Cpu, Et18
```

- The show platform arad acl tcam command displays the number of TCAM entries (hardware resources) occupied by the ACL on each forwarding ASIC and the percentage of TCAM utilization per forwarding ASIC.

```
switch# show platform arad acl tcam detail
ip access-list ipAclLimitTest (Shared RACL, 0 rules, 1 entries, direction
 out,
state success, Acl Label 2)
Fap: Arad0, Shared: true, Interfaces: Vl2148, Vl2700
Bank Offset Entries
0         0       1
Fap: Arad1, Shared: true, Interfaces: Vl2148
Bank Offset Entries
0         0       1

switch# show platform arad acl tcam summary
The total number of TCAM lines per bank is 1024.
============================================================
Arad0:
============================================================
 Bank   Used                  Used %             Used By
    0      1                       0   IP Egress PACLs/RACLs
Total Number of TCAM lines used is: 1
============================================================
Arad1:
============================================================
 Bank   Used                   Used %             Used By
    0      1                       0   IP Egress PACLs/RACLs
Total Number of TCAM lines used is: 1
```

- The **show ip route** command displays the unicast ip routes installed in the system.

```
switch# show ip route
VRF name: default
Codes: C - connected, S - static, K - kernel,
 O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
 E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
 N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
 R - RIP, I - ISIS, A B - BGP Aggregate, A O - OSPF Summary,
 NG - Nexthop Group Static Route
```

```
Gateway of last resort is not set
 C 10.1.0.0/16 is directly connected, Vlan2659
 C 10.2.0.0/16 is directly connected, Vlan2148
 C 10.3.0.0/16 is directly connected, Vlan2700
 S 172.17.0.0/16 [1/0] via 172.24.0.1, Management1
 S 172.18.0.0/16 [1/0] via 172.24.0.1, Management1
 S 172.19.0.0/16 [1/0] via 172.24.0.1, Management1
 S 172.20.0.0/16 [1/0] via 172.24.0.1, Management1
 S 172.22.0.0/16 [1/0] via 172.24.0.1, Management1
 C 172.24.0.0/18 is directly connected, Management1
```

- The **show platform arad ip route** command displays the platform unicast forwarding routes.

```
switch# show platform arad ip route
Tunnel Type: M(mpls), G(gre)
 ---------------------------------------------------------------------------
|                              Routing Table                                |
     |
|-------------------------------------------------------------------------
|VRF|   Destination    |      |                    |     | Acl   |                |
ECMP| FEC | Tunnel
| ID|   Subnet         | Cmd  |       Destination  | VID | Label |  MAC / CPU Code
|Index|Index|T Value

---------------------------------------------------------------------------
|0  |0.0.0.0/8          |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1031 | -
|0  |10.1.0.0/16        |TRAP | CoppSystemL3DstMiss|2659 | - | ArpTrap | - |1030 | -
|0  |10.2.0.0/16        |TRAP | CoppSystemL3DstMiss|2148 | - | ArpTrap | - |1026 | -
|0  |10.3.0.0/16        |TRAP | CoppSystemL3DstMiss|2700 | - | ArpTrap | - |1034 | -
|0  |127.0.0.0/8        |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1031 | -
|0  |172.17.0.0/16      |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1025 | -
|0  |172.18.0.0/16      |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1025 | -
|0  |172.19.0.0/16      |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1025 | -
|0  |172.20.0.0/16      |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1025 | -
|0  |172.22.0.0/16      |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1025 | -
|0  |172.24.0.0/18      |TRAP | CoppSystemL3DstMiss|0    | - | ArpTrap | - |1032 | -
|0  |0.0.0.0/0          |TRAP | CoppSystemL3LpmOver|0    | - | SlowReceive | -
|1024 | -
|0  |10.1.0.0/32*       |TRAP | CoppSystemIpBcast  |0    | - | BcastReceive | -
|1027 | -
|0  |10.1.0.1/32*       |TRAP | CoppSystemIpUcast  |0    | - | Receive | - |32766| -
|0  |10.1.255.1/32*     |ROUTE| Po1                |2659 |4094 | 00:1f:5d:6b:ce:45
| - |1035 | -
|0  |10.1.255.255/32*   |TRAP | CoppSystemIpBcast  |0    | - | BcastReceive | -
|1027 | -
|0  |10.2.0.0/32*       |TRAP | CoppSystemIpBcast  |0    | - | BcastReceive | -
|1027 | -
|0  |10.2.0.1/32*       |TRAP | CoppSystemIpUcast  |0    | - | Receive | - |32766| -
|0  |10.2.255.1/32*     |ROUTE| Et1                |2148 |2 | 00:1f:5d:6d:54:dc |
- |1036 | -
|0  |10.2.255.255/32*   |TRAP | CoppSystemIpBcast  |0    | - | BcastReceive | -
|1027 | -
|0  |10.3.0.0/32*       |TRAP | CoppSystemIpBcast  |0    | - | BcastReceive | -
|1027 | -
|0  |10.3.0.1/32*       |TRAP | CoppSystemIpUcast  |0    | - | Receive | - |32766| -
|0  |10.3.255.1/32*     |ROUTE| Et18               |2700 |2 | 00:1f:5d:6b:00:01 |
- |1038 | -
```

13.4.6 Route Maps

A route map is an ordered set of rules that control the redistribution of IP routes into a protocol domain on the basis of such criteria as route metrics, access control lists, next hop addresses, and route tags. Route maps can also alter parameters of routes as they are redistributed.

13.4.6.1 Route Map Description

Route maps are composed of route map statements, each of which consists of a list of match and set commands.

Route Map Statements

Route map statements are categorized by the resolution of routes that the statement filters.

- Permit statements facilitate the redistribution of matched routes.
- Deny statements prevent the redistribution of matched routes.

Route map statement elements include name, sequence number, filter type, match commands, set commands, and continue commands.

- ***name*** identifies the route map to which the statement belongs.
- ***sequence number*** designates the statement's placement within the route map.
- **filter type** specifies the route resolution. Valid types are **permit** and **deny**.
- **match commands** specify criteria that select routes that the statement is evaluating for redistribution.
- **set commands** modify route parameters for redistributed routes.
- **continue commands** prolong the route map evaluation of routes that match a statement.

Statements filter routes for redistribution. Routes that statements pass are redistributed (permit statements) or rejected (deny statements). Routes that statements fail are filtered by the next statement in the route map.

- When a statement does not contain a **`match`** command, the statement passes all routes.
- When a statement contains a single **`match`** command that lists a single object, the statement passes routes whose parameters match the object.
- When a statement contains a single **`match`** command that lists multiple objects, the statements passes routes whose parameters match at least one object.
- When a statement contains multiple **`match`** commands, the statement passes routes whose parameters match all match commands.

`Set` commands modify parameters for redistributed routes. **`Set`** commands are valid in permit statements.

Example

The following route map statement is named ***MAP_1*** with sequence number ***10***. The statement matches all routes from BGP Autonomous System 10 and redistributes them with a local preference set to ***100***. Routes that do not match the statement are evaluated against the next statement in the route map.

```
switch# route-map MAP_1 permit 10
   match as 10
   set local-preference 100
```

Route Maps with Multiple Statements

A route map consists of statements with the same name and different sequence numbers. Statements filter routes in ascending order of their sequence numbers. When a statements passes a route, the redistribution action is performed as specified by the filter type and all subsequent statements are ignored. When the statement fails the route, the statement with the smallest sequence number that is larger than the current one filters the route.

All route maps have an implied final statement that contains a single deny statement with no match command. This denies redistribution to routes that are not passed by any statement.

Example

The following route map is named ***MAP_1*** with two permit statements. Routes that do not match either statement are denied redistribution into the target protocol domain.

```
switch# route-map MAP_1 permit 10
   match as 10
   set local-preference 100
!
```

```
switch# route-map MAP_1 permit 20
   match metric-type type-1
   match as 100
```

Route Map Configuration describes route map configuration procedures.

Route Maps with Multiple Statements and Continue Commands

Route map statements that contain a **continue (route map)** command support additional route map evaluation of routes whose parameters meet the statement's match commands. Routes that match a statement containing a **continue** command are evaluated against the statement specified by the **continue** command.

When a route matches multiple route map statements, the filter action (deny or permit) is determined by the last statement that the route matches. The **set** commands in all statements matching the route are applied to the route after the route map evaluation is complete. Multiple set commands are applied in the same order by which the route was evaluated against the statements containing them.

Example

The following route map is named ***MAP_1*** with a permit statement and a deny statement. The permit statement contains a continue command. Routes that match statement 10 are evaluated against statement 20.

```
route-map MAP_2 permit 10
   match as 10
   continue 20
   set local-preference 100
!
route-map MAP_2 deny 20
   match metric-type type-1
   match as 100
```

The route is redistributed if it passes statement 10 and is rejected by statement 20. The route is denied redistribution in all other instances. The **continue** command guarantees the evaluation of all routes against both statements.

13.4.6.2 Route Map Configuration

Route maps are created and modified in route map configuration mode. These sections describe the configuration mode and its commands.

- Route Map Creation and Editing
- Modifying Route Map Components

13.4.6.2.1 Route Map Creation and Editing

Creating a Route Map Statement

To create a route map, enter route-map followed by the map name and filter type (**deny** or **permit**). The default sequence number is assigned to the statement if the command does not include a number.

Example

This command places the switch in ***route map*** configuration mode to create a route map statement named *map1* with a sequence number of ***50***.

```
switch(config)# route-map map1 permit 50
```

```
switch(config-route-map-map1)#
```

Editing a Route Map Statement

To edit an existing route map statement, enter route-map with the map's name and statement's number. The switch enters route map configuration mode for the statement. Subsequent match (route-map) and set (route-map) commands add the corresponding commands to the statement.

The **show** command displays contents of the existing route map.

Example

This command places the switch in route map configuration mode to edit an existing route map statement. The **show** command displays contents of all statements in the route map.

```
switch(config)# route-map MAP2
switch(config-route-map-MAP2)#show
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
switch(config-route-map-MAP2)#
```

Saving Route Map Modifications

Route map configuration mode is a group-change mode. Changes are saved by exiting the mode, either with an explicit **exit** command or by switching directly to another configuration mode. This includes switching to the configuration mode for a different route map.

Example

The first command creates the ***map1*** statement with sequence number of 10. The second command is not yet saved to the route map, as displayed by the **show** command.

```
switch(config)# route-map map1 permit
switch(config-route-map-map1)# match as 100
switch(config-route-map-map1)# show

switch(config-route-map-map1)#
```

The **exit** command saves the **match** command.

```
switch(config-route-map-map1)# exit
switch(config)# show route-map map1
route-map map1 permit 10
  Match clauses:
    match as 100
  Set clauses:
switch(config)#
```

Discarding Route Map Modifications

The **abort** command discards all pending changes and exits route map configuration mode.

Example

The **abort** command discards the pending **match** command and restores the original route map.

```
switch(config)# route-map map1 permit
switch(config-route-map-map1)# match as 100
switch(config-route-map-map1)# abort
switch(config)# show route-map map1
switch(config)#
```

13.4.6.2.2 Modifying Route Map Components

These commands add rules to the configuration mode route map:

- match (route-map) adds a match rule to a route map.
- set (route-map) adds a set rule to a route map.

Inserting a Statement

To insert a new statement into an existing route map, create a new statement with a sequence number that differs from any existing statement in the map.

Example

This command adds statement 50 to the ***Map1*** route map, then displays the new route map.

```
switch(config)# route-map Map1 permit 50
switch(config-route-map-Map1)# match as 150
switch(config-route-map-Map1)#exit
switch(config)#show route-map Map1
route-map Map1 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
route-map Map1 permit 50
  Match clauses:
    match as 150
  Set clauses:
switch(config)#
```

Deleting Route Map Components

To remove a component from a route map, perform one of the following:

- To remove a command from a statement, enter **no**, followed by the command to be removed.
- To remove a statement, enter no, followed by the route map with the filter type and the sequence number of the statement to be removed.
- To remove a route map, enter **no** followed by the route map without a sequence number.

13.4.6.3 Using Route Maps

Protocol redistribution commands include a route map parameter that determines the routes to be redistributed into the specified protocol domain.

Example

This command uses ***Map1*** route map to select OSPFv2 routes for redistribution into BGP AS1.

```
switch(config)# router bgp 1
switch(config-router-bgp)# redistribute ospf route-map Map1
switch(config-router-bgp)# exit
switch(config)#
```

13.4.7 Prefix Lists

A prefix list is an ordered set of rules that defines route redistribution access for a specified IP address space. A prefix list rules consists of a filter action (deny or permit), an address space identifier (IPv4 subnet address or IPv6 prefix), and a sequence number.

Prefix lists are referenced by route map match commands when filtering routes for redistribution.

- Prefix List Configuration describes the prefix list configuration process.
- Using Prefix Lists describes the use of prefix lists.
- Static Routes Redistribution into IGPs describes redistribution of routes whose configured next-hops satisfy the route-map policy.

13.4.7.1 Prefix List Configuration

A prefix list is an ordered set of rules that defines route redistribution access for a specified IP address space. A prefix list rule consists of a filter action (deny or permit), a network address (IPv4 subnet or IPv6 prefix), and a sequence number. A rule may also include a alternate mask size.

The switch supports IPv4 and IPv6 prefix lists. The switch is placed in a Prefix-list configuration mode to create and edit IPv4 or IPv6 prefix lists.

13.4.7.1.1 IPv4 Prefix Lists

IPv4 prefix lists are created or modified by adding an IPv4 prefix list rule in the Prefix-list configuration mode. Each rule includes the name of a prefix list, in addition to the sequence number, network address, and filter action. A list consists of all rules that have the same prefix list name.

The ip prefix-list command creates a prefix list or adds a rule to an existing list. Route map match commands use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains.

Creating an IPv4 Prefix List

To create an IPv4 prefix list, enter the ip prefix-list command, followed by the name of the list. The switch enters ***IPv4 prefix-list*** configuration mode for the list. If the command is followed by the name of an existing ACL, subsequent commands edit that list.

Examples

- This command places the switch in ***IPv4 prefix list*** configuration mode to create an IPv4 prefix list named route-one.

```
switch(config)# ip prefix-list route-one
switch(config-ip-pfx)#
```

- These commands create four different rules for the prefix-list named ***route-one***.

```
switch(config)# ip prefix-list route-one
switch(config-ip-pfx)# seq 10 deny 10.1.1.0/24
switch(config-ip-pfx)# seq 20 deny 10.1.0.0/16
switch(config-ip-pfx)# seq 30 permit 12.15.4.9/32
```

```
switch(config-ip-pfx)# seq 40 deny 1.1.1.0/24
```

To view the list, save the rules by exiting the ***Prefix-list*** command mode, then re-enter the configuration mode and type **`show active`**.

```
switch(config-ip-pfx)# exit
switch(config)# ip prefix-list route-one
switch(config-ip-pfx)# show active
ip prefix-list route-one
   seq 10 deny 10.1.1.0/24
   seq 20 deny 10.1.0.0/16
   seq 30 permit 12.15.4.9/32
   seq 40 deny 1.1.1.0/24
switch(config-ip-pfx)# ip prefix-list route-one
```

IPv4 prefix lists are referenced in match (route-map) command.

13.4.7.1.2 IPv6 Prefix Lists

Creating an IPv6 Prefix List

The switch provides ***IPv6 prefix-list*** configuration mode for creating and modifying IPv6 prefix lists. A list can be edited only in the mode where it was created.

To create an IP ACL, enter the ipv6 prefix-list command, followed by the name of the list. The switch enters ***IPv6 prefix-list*** configuration mode for the list. If the command is followed by the name of an existing ACL, subsequent commands edit that list.

Example

This command places the switch in ***IPv6 prefix list*** configuration mode to create an IPv6 prefix list named ***map1***.

```
switch(config)# ipv6 prefix-list map1
switch(config-ipv6-pfx)#
```

Adding a Rule

To append a rule to the end of a list, enter the rule without a sequence number while in ***Prefix-List*** configuration mode for the list. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Example

These commands enter the first two rules into a new prefix list.

```
switch(config-ipv6-pfx)# permit 3:4e96:8ca1:33cf::/64
switch(config-ipv6-pfx)# permit 3:11b1:8fe4:1aac::/64
```

To view the list, save the rules by exiting the ***prefix-list*** command mode, then re-enter the configuration mode and type **`show active`**.

```
switch(config-ipv6-pfx)# exit
switch(config)# ipv6 prefix-list map1
switch(config-ipv6-pfx)# show active
ipv6 prefix-list map1
   seq 10 permit 3:4e96:8ca1:33cf::/64
   seq 20 permit 3:11b1:8fe4:1aac::/64
```

```
switch(config-ipv6-pfx)#
```

This command appends a rule to the end of the prefix list. The new rule's sequence number is ***30***.

```
switch(config-ipv6-pfx)# permit 3:1bca:1141:ab34::/64
switch(config-ipv6-pfx)# exit
switch(config)# ipv6 prefix-list map1
switch(config-ipv6-pfx)# show active
ipv6 prefix-list map1
   seq 10 permit 3:4e96:8ca1:33cf::/64
   seq 20 permit 3:11b1:8fe4:1aac::/64
   seq 30 permit 3:1bca:1141:ab34::/64
switch(config-ipv6-pfx)#
```

Inserting a Rule

To insert a rule into a prefix list, use the **seq (IPv6 Prefix Lists)** command to enter a rule with a sequence number that is between numbers of two existing rules.

Example

This command inserts a rule between the first two rules by assigning it the sequence number ***15***.

```
switch(config-ipv6-pfx)# seq 15 deny 3:4400::/64
switch(config-ipv6-pfx)# exit
switch(config)# show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
switch(config)#
```

Deleting a Rule

To remove a rule from the configuration mode prefix list, enter **`no seq`** (see **seq (IPv6 Prefix Lists)**), followed by the sequence number of the rule to be removed.

Example

These commands remove rule 20 from the prefix list, then displays the resultant prefix list.

```
switch(config-ipv6-pfx)# no seq 20
switch(config-ipv6-pfx)# exit
switch(config)# show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 30 permit 3:1bca:3ff2:634a::/64
switch(config)#
```

13.4.7.2 Using Prefix Lists

Route map match commands include an option that matches a specified prefix list.

Example

The ***MAP_1*** route map uses a match command that references the ***PL_1*** prefix list.

```
switch(config)# route-map MAP_1 permit
switch(config-route-map-MAP_1)# match ip address prefix-list PL_1
switch(config-route-map-MAP_1)# set community 500
switch(config-route-map-MAP_1)# exit
```

13.4.7.3 Static Routes Redistribution into IGPs

Use **match ip next-hop** route-map, while redistributing static routes into IGPs to redistribute the static routes whose **configured** next-hops satisfies the route-map policy.

The following example applies match ip next-hop clause for static routes redistributed into IGPs for multi-agent mode as well. The following configures a static route.

```
switch(config)# ip route 10.20.30.0/24 1.2.3.4
```

The following configures a prefix-list.

```
switch (config)# ip prefix-list prefixListName
switch(config-ip-pfx)# permit 1.2.3.4/32
```

1.2.3.4 is a **configured** next-hop for static route ***10.20.30.0/24***.

The following configures a route map.

```
switch(config)# route-map routeMapName
switch(config-route-map-routeMapName)# match ip next-hop prefix-list
 prefixListName
```

To redistribute static routes with ‘match ip next-hop’ route-map clause in IS-IS.

```
switch(config-router-isis)# redistribute static route-map routeMapName
```

Redistributed routes can be seen using the following show commands. If routes are redistributed into IS-IS then **show isis database detail**If routes are redistributed into OSPFv2 then **show ip ospf database detail**.

```
switch# show ip route

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

Gateway of last resort is not set

 ...
 I L2     10.20.30.0/24 [115/10] via 1.2.3.4, Ethernet1


switch# show isis database detail

IS-IS Instance: B VRF: default
```

```
  IS-IS Level 1 Link State Database
    LSPID                 Seq Num   Cksum  Life   IS Flags
    ...
  IS-IS Level 2 Link State Database
    LSPID                 Seq Num   Cksum  Life   IS Flags
    0000.0000.0001.00-00  6         10364  840    L2 <>
      ...
      Reachability         : 10.20.30.0/24 Metric: 0 Type: 1 Up
      ...
```

13.4.8 Port ACLs with User-Defined Fields

Describes the support for specifying User-Defined Fields (UDF) in Port ACLs including IPv4, IPv6, and MAC ACLs. The purpose of the User-Defined Fields feature is to permit or deny packets based on custom offset pattern matching.

User-Defined Fields, or UDFs, are defined as part of an access-list filter and are comprised of an offset, length, pattern match and mask. This describes a single portion of any incoming packet to match the provided value upon.

UDFs may also be defined via aliases. Aliases are a way to save a UDF configuration for reuse in multiple access-lists and or access-list rules. An alias may substitute for a fully defined UDF including the offset, pattern and mask. The pattern or mask may be overridden when the alias is used in an access-list rule.

The behavior, CLI syntax and configuration of UDFs are identical to Traffic Steering UDF and Mirroring ACL UDF.

This section describes port ACLs with user-defined fields, including configuration instructions. Topics covered by this section include:

- Configuring Port ACLs with User-Defined Fields
- Port ACLs with User-Defined Fields Limitations

13.4.8.1 Configuring Port ACLs with User-Defined Fields

User-Defined Fields are specified as part of an access-list. The type of access-list however, dictates the base position of the UDF and the options available. In addition, a TCAM profile must be configured to include UDFs as part of the Port ACL feature's key.

13.4.8.1.1 TCAM Profile

User-Defined Fields are defined as additional fields in the Port ACL feature's key. By default, UDFs are not included in the keys for the Port ACL features. Adding a UDF to the key requires removal of different key fields to fit within the TCAM width restrictions.



Note: Each UDF is either 16 bits wide or 32 bits wide.

Below are example configurations of the TCAM profile.

13.4.8.1.1.1IPv4 Port ACL

The following configurations create a new profile based on the default profile. This new profile replaces the Layer 4 port key fields with one 16-bit UDF and one 32-bit UDF.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile ipv4Udf copy default
switch(config-hw-tcam-profile-ipv4Udf)# feature acl port ip
switch(config-hw-tcam-profile-ipv4Udf-feature-acl-port-ip)# no key field l4-ops
switch(config-hw-tcam-profile-ipv4Udf-feature-acl-port-ip)# no key field l4-
src-port
```

```
switch(config-hw-tcam-profile-ipv4Udf-feature-acl-port-ip)# no key field l4-
dst-port
switch(config-hw-tcam-profile-ipv4Udf-feature-acl-port-ip)# key field udf-16b-1
switch(config-hw-tcam-profile-ipv4Udf-feature-acl-port-ip)# key field udf-32b-1
switch(config-hw-tcam-profile-ipv4Udf-feature-acl-port-ip)# exit
switch(config-hw-tcam-profile-ipv4Udf)# exit
switch(config-hw-tcam)# system profile ipv4Udf
```

Example-16-bit IPv4 Header Match

The following configurations match IPv4 packets based on the Identification(ID) field. Packets ingressing into ***interface ethernet 7*** with an ID equal to ***1000*** is forwarded. Packets with an ID different than ***1000*** is dropped.

```
(config)# ip access-list udfAcl
(config-acl-udfAcl)# permit ip any any payload header start offset 1 pattern
 0x03E80000 mask 0x0000FFFF
(config-acl-udfAcl)# deny ip any any
(config-acl-udfAcl)# exit
(config)# interface ethernet 7
(config-if-Et7)#
```

13.4.8.1.1.2IPv6 Port ACL

The following configurations create a new profile based on the default profile. This new profile replaces the destination IPv6 address key field with two 32-bit UDFs.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile ipv6Udf copy default
switch(config-hw-tcam-profile-ipv6Udf)# feature acl port ipv6
switch(config-hw-tcam-profile-ipv6Udf-feature-acl-port-ipv6)# no key field dst-
ipv6
switch(config-hw-tcam-profile-ipv6Udf-feature-acl-port-ipv6)# key field
 udf-32b-1
switch(config-hw-tcam-profile-ipv6Udf-feature-acl-port-ipv6)# key field
 udf-32b-2
switch(config-hw-tcam-profile-ipv6Udf-feature-acl-port-ipv6)# exit
switch(config-hw-tcam-profile-ipv6Udf)# exit
switch(config-hw-tcam)# system profile ipv6Udf
```

Example-32-bit IPv6 Payload Match

The following configurations match IPv6 UDP packets based on the first 32 bits of the packet payload. UDP packets ingressing into ***interface ethernet 7*** that starts with ***0x1234567X*** (where X can be any valid hexadecimal) in the payload are forwarded. Any other packets are dropped. The offset is set to 2 (2 x 4-byte words) to skip the UDP header.

```
(config)# ipv6 access-list udfAcl
(config-ipv6-acl-udfAcl)# permit udp any any payload offset 2 pattern
 0x12345670 mask 0x0000000f
(config-ipv6-acl-udfAcl)# deny ipv6 any any
(config-ipv6-acl-udfAcl)# exit
(config)# interface ethernet 7
(config-if-Et7)# ipv6 access-group udfAcl in
```

13.4.8.2 Port ACLs with User-Defined Fields Limitations

User-defined fields consume a limited set of copy resources. For each unique offset, if a pattern is specified that is masked to be > 16 bits wide, then a 32-bit resource is used. If no 32-bit resource is available, then two 16-bit resources are used, if available. Copy resources are dependent upon the number of UDF key fields added to the feature key. Each UDF key field maps to one copy resource. Using the above TCAM profile configurations:

- IPv4: 1 x 16-bit pattern + 1 x 32-bit pattern.
- IPv6: 2 x 32-bit pattern.
- MAC: 1 x 16-bit pattern + 1 x 32-bit pattern.

Other limitations include:

- Maximum offset value is ***31***, which is 31 4-byte words, or 124 bytes.
- UDFs only work on ingress Port ACLs.

13.4.9 ACL, Route Map, and Prefix List Commands

This section describes CLI commands that this chapter references.

ACL Creation and Access Commands

- hardware access-list resource sharing vlan in
- hardware access-list resource sharing vlan ipv4 out
- ip access-list
- ip access-list standard
- ipv6 access-list
- ipv6 access-list standard
- mac access-list
- system profile

ACL Implementation Commands

- ip access-group
- ipv6 access-group
- mac access-group

Service ACL Implementation Commands

- ip access-group (Service ACLs)
- ipv6 access-group (Service ACLs)

ACL Edit Commands

- counters per-entry (ACL configuration modes)
- hardware access-list update default-result permit
- no <sequence number> (ACLs)
- resequence (ACLs)
- show (ACL configuration modes)

ACL Rule Commands

- deny (IPv4 ACL)
- deny (IPv6 ACL)
- deny (MAC ACL)
- deny (Standard IPv4 ACL)

- deny (Standard IPv6 ACL)
- permit (IPv4 ACL)
- permit (IPv6 ACL)
- permit (MAC ACL)
- permit (Standard IPv4 ACL)
- permit (Standard IPv6 ACL)
- remark

ACL List Counter Commands

- clear ip access-lists counters
- clear ipv6 access-lists counters
- hardware counter feature acl out

ACL Display Commands

- show ip access-lists
- show ipv6 access-lists
- show mac access-lists

Prefix List Creation and Access Commands

- ip prefix-list
- ipv6 prefix-list

Prefix List Edit Commands

- deny (IPv6 Prefix List)
- permit (IPv6 Prefix List)
- seq (IPv6 Prefix Lists)

Prefix List Display Commands

- show hardware tcam profile
- show ip prefix-list
- show ipv6 prefix-list
- show platform arad acl tcam
- show platform arad acl tcam summary
- show platform arad mapping
- show platform fap acl
- show platform fap acl tcam
- show platform fap acl tcam hw
- show platform fap acl tcam summary
- show platform trident tcam

Route Map Creation and Access Command

- route-map

Route Map Edit Commands

- continue (route map)
- description (route map)
- match (route-map)
- set (route-map)

- set as-path prepend
- set as-path match
- set community (route-map)
- set extcommunity (route-map)

Route Map Display Commands

- show route-map

13.4.9.1 clear ip access-lists counters

The **clear ip access-lists counters** command sets ACL counters to zero for the specified IPv4 Access Control List (ACL). The **session** parameter limits ACL counter clearing to the current CLI session.

Command Mode

Privileged EXEC

Command Syntax

clear ip access-lists counters [**ACL_NAME**][**SCOPE**]

Parameters

- **ACL_NAME** Name of ACL. Options include:
  - ***no parameter*** all ACLs.
  - ***access_list*** name of ACL.
- **SCOPE** Session affected by command. Options include:
  - ***no parameter*** command affects counters on all CLI sessions.
  - **session** affects only current CLI session.

Example

This command resets all IPv4 ACL counters.

```
switch(config)# clear ip access-lists counters
switch(config)#
```

13.4.9.2 clear ipv6 access-lists counters

The **clear ipv6 access-lists counters** command sets ACL counters to zero for the specified IPv6 Access Control List (ACL). The **session** parameter limits ACL counter clearing to the current CLI session.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 access-lists counters [**ACL_NAME**][**SCOPE**]

Parameters

- **ACL_NAME** name of ACL. Options include:
  - ***no parameter*** all IPv6 ACLs.
  - ***access_list*** name of IPv6 ACL.
- **SCOPE** Session affected by command. Options include:
  - ***no parameter*** command affects counters on all CLI sessions.
  - **session** affects only current CLI session.

Example

This command resets all IPv6 ACL counters.

```
switch(config)# clear ipv6 access-lists counters
switch(config)#
```

13.4.9.3 continue (route map)

The **continue** command creates a route map statement entry that enables additional route map evaluation of routes whose parameters meet the statement's matching criteria.

A statement typically contains a match (route-map) and a set (route-map) command. The evaluation of routes whose settings are the same as **match** command parameters normally ends and the statement's **set** commands are applied to the route. Routes that match a statement containing a **continue** command are evaluated against the statement specified by the continue command.

When a route matches multiple route map commands, the filter action (**deny** or **permit**) is determined by the last statement that the route matches. The **set** commands in all statements matching the route are applied to the route after the route map evaluation is complete. Multiple set commands are applied in the same order by which the route was evaluated against the statement containing them.

The **no continue** and **default continue** commands remove the corresponding **continue** command from the configuration mode *route map* statement by deleting the corresponding command from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

continue NEXT_SEQ

no continue NEXT_SEQ

default continue NEXT_SEQ

Parameters

NEXT_SEQ Specifies next statement for evaluating matching routes. Options include:

- ***no parameter*** Next statement in the route map, as determined by sequence number.
- ***seq_number*** Specifies the number of the next statement. Values range from ***1*** to ***16777215***.

Restrictions

A **continue** command cannot specify a sequence number smaller than the sequence number of its route map statement.

Related Command

route-map command enters ***route map*** configuration mode.

Example

This command creates route map ***map1***, statement ***40*** with a match command, a set command, and a continue command. Routes that match the statement are subsequently evaluated against statement ***100***. The **set local-preference** command is applied to matching routes regardless of subsequent matching operations.

```
switch(config)# route-map map1 deny 40
switch(config-route-map-map1)# match as 15
switch(config-route-map-map1)# continue 100
switch(config-route-map-map1)# set local-preference 50
```

```
switch(config-route-map-map1)#
```

13.4.9.4 counters per-entry (ACL configuration modes)

The **counters per-entry** command places the ACL in counting mode. In counting mode, the feature generally displays the number of instances each rule in the list matches an inbound packet and the elapsed time since the last match. However, for certain select platforms, in addition to the packet counter, ACL counters can also be enabled for byte counts when applied to data plane ACLs. A complete list of platforms that support byte count for data plan ACLs are listed below:



Note: Byte counting is supported only for data plane ACLs.

Only the below platforms support ACL byte counting

- CCS-710/720/722/755/758 series
- DCS-7010TX
- DCS-7050SX3/CX3/TX3/CX4/DX4/PX4
- DCS-7060 Series
- DCS-7300X3/7304X3/7308X3/7316/7320X/7324/7328/7358X4/7368/7388

On the FM6000 platform, this command has no effect when used in an ACL that is part of a PBR class map.

The **no counters per-entry** and **default counters per-entry** commands place the ACL in non-counting mode.

Command Mode

ACL Configuration

IPv6-ACL Configuration

Std-ACL Configuration

Std-IPv6-ACL Configuration

MAC-ACL Configuration

Command Syntax

counters per-entry

no counters per-entry

default counters per-entry

Examples

- This command places the ***test1*** ACL in counting mode.

```
switch(config)# ip access-list test1
switch(config-acl-test1)# counters per-entry
switch(config-acl-test1)#
```

- This command displays the ACL, with counter information, for an ACL in counting mode.

```
switch# show ip access-lists
IP Access List default-control-plane-acl [readonly]
  counters per-entry
  10 permit icmp any any
  20 permit ip any any tracked [match 12041 packets, 0:00:00 ago]
  30 permit ospf any any
  40 permit tcp any any eq ssh telnet www snmp bgp https [match 11 packets,
 1:41:07 ago]
  50 permit udp any any eq bootps bootpc snmp rip [match 78 packets, 0:00:27
 ago]
```

```
  60 permit tcp any any eq mlag ttl eq 255
  70 permit udp any any eq mlag ttl eq 255
  80 permit vrrp any any
  90 permit ahp any any
  100 permit pim any any
  110 permit igmp any any [match 14 packets, 0:23:27 ago]
  120 permit tcp any any range 5900 5910
  130 permit tcp any any range 50000 50100
  140 permit udp any any range 51000 51100
```

- On the platforms that support byte counting, counter information is displayed as shown below:

```
switch#show ip access-lists
IP Access List default-control-plane-acl [readonly]
        counters per-entry
        10 permit icmp any any [match 30 packets, 0:02:08 ago]
        20 permit ip any any tracked [match 97777 packets, 0:00:00 ago]
        30 permit udp any any eq bfd ttl eq 255
        40 permit udp any any eq bfd-echo ttl eq 254
        50 permit udp any any eq multihop-bfd micro-bfd sbfd
        60 permit udp any eq sbfd any eq sbfd-initiator
        70 permit ospf any any
        80 permit tcp any any eq ssh telnet www snmp bgp https msdp ldp
 netconf-ssh gnmi [match 72 packets, 0:00:00 ago]
        90 permit udp any any eq bootps bootpc snmp rip ntp ldp ptp-event
 ptp-general
        100 permit tcp any any eq mlag ttl eq 255
        110 permit udp any any eq mlag ttl eq 255
        120 permit vrrp any any
        130 permit ahp any any
        140 permit pim any any


IP Access List ipCountersTest:The ipCountersTest ACL is applied to the data
 plane. Hence, it displays the byte count information as shown below:
        counters per-entry
        10 permit tcp host 10.1.1.1 range 2000 4000 host 10.2.1.1 [match 486
 bytes in 3 packets, 0:00:26 ago]
        20 permit tcp host 10.1.1.1 range 14000 16000 host 10.2.1.1 [match
 486 bytes in 3 packets, 0:00:18 ago]
        30 permit udp host 10.1.1.1 range 62000 64000 host 10.2.1.1 [match
 450 bytes in 3 packets, 0:00:00 ago]
        40 permit tcp host 10.1.1.1 range 50000 52000 host 10.2.1.1 [match
 486 bytes in 3 packets, 0:00:02 ago]
        50 permit tcp host 10.1.1.1 range 38000 40000 host 10.2.1.1 [match
 486 bytes in 3 packets, 0:00:10 ago]
        60 permit tcp host 10.1.1.1 range 26000 28000 host 10.2.1.1 [match
 486 bytes in 3 packets, 0:00:18 ago]
```

13.4.9.5 deny (IPv4 ACL)

The **deny** command adds a deny rule to the configuration mode IPv4 Access Control List (ACL). Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding ***10*** to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes the specified rule from the ACL.

Command Mode

ACL Configuration

Command Syntax

[**SEQ_NUM**] **deny PROTOCOL SOURCE_ADDR** [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**][**FLAGS**] [**MESSAGE**][**fragments**][**tracked**][**DSCP_FILTER**][**TTL_FILTER**][log]

no deny PROTOCOL SOURCE_ADDR [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**][**FLAGS**] [**MESSAGE**][**fragments**][**tracked**][**DSCP_FILTER**][**TTL_FILTER**][log]

default deny PROTOCOL SOURCE_ADDR [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**][**FLAGS**] [**MESSAGE**][**fragments**][**tracked**][**DSCP_FILTER**][**TTL_FILTER**][log]

Note: Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a deny rule.

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding 10 to the number of the ACL's last rule.
  - ***14294967295*** Number assigned to entry.
- **PROTOCOL** protocol field filter. Values include:
  - **ahp** Authentication Header Protocol (51).
  - **icmp** Internet Control Message Protocol (1).
  - **igmp** Internet Group Management Protocol (2).
  - **ip** Internet Protocol v4 (4).
  - **ospf** Open Shortest Path First (89).
  - **pim** Protocol Independent Multicast (103).
  - **tcp** Transmission Control Protocol (6).
  - **udp** User datagram protocol (17).
  - **vrrp** Virtual Router Redundancy Protocol (112).
  - ***protocol_num*** Integer corresponding to an IP protocol. Values range from ***0*** to ***255***.
- **SOURCE_ADDR** and **DEST_ADDR** Source and destination address filters. Options include:
  - ***network_addr*** Subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host *ip_addr*** IP address (dotted decimal notation).

  Subnet addresses support discontiguous masks.
- **SOURCE_PORT** and **DEST_PORT** Source and destination port filters. Options include:
  - **any** All ports.
  - **eq *port-1 port-2*** ... ***port-n*** A list of ports. Maximum list size is 10 ports.
  - **neq *port-1 port-2*** ... ***port-n*** The set of all ports not listed. Maximum list size is 10 ports.
  - **gt *port*** The set of ports with larger numbers than the listed port.
  - **lt *port*** The set of ports with smaller numbers than the listed port.
  - **range *port_1 port_2*** The set of ports whose numbers are between the range.
- **fragments** Filters packets with FO bit set (indicates a non-initial fragment packet).
- **FLAGS** Flag bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- **MESSAGE** Message type filters (ICMP packets). Use CLI syntax assistance (?) to display options.
- **tracked** Rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- **DSCP_FILTER** Rule filters packet by its DSCP value. Values include:
  - ***no parameter*** Rule does not use DSCP to filter packets.
  - **dscp *dscp_value*** Packets match if DSCP field in packet is equal to ***dscp_value***.
- **TTL_FILTER** Rule filters packet by its TTL (time-to-live) value. Values include:

- **ttl eq *ttl_value*** Packets match if **ttl** in packet is equal to ***ttl_value***.
- **ttl gt *ttl_value*** Packets match if **ttl** in packet is greater than ***ttl_value***.
- **ttl lt *ttl_value*** Packets match if **ttl** in packet is less than ***ttl_value***.
- **ttl neq *ttl_value*** Packets match if **ttl** in packet is not equal to ***ttl_value***.
    - Valid in ACLs applied to the control plane.
    - Validity in ACLs applied to data plane varies by switch platform.
- **log** Triggers an informational log message to the console about the matching packet.
    - Valid in ACLs applied to the control plane.
    - Validity in ACLs applied to data plane varies by switch platform.

Examples

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops OSPF packets from ***10.10.1.1/24*** to any host.

```
switch(config)# ip access-list text1
switch(config-acl-text1)# deny ospf 10.1.1.0/24 any
switch(config-acl-text1)#
```

- This command inserts a **deny** statement with the sequence number 65. The **deny** statement drops all PIM packets.

```
switch(config-acl-text1)# 65 deny pim any any
switch(config-acl-text1)#
```

13.4.9.6 deny (IPv6 ACL)

The **deny** command adds a deny rule to the configuration mode IPv6 Access Control List (ACL). Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding ***10*** to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes the specified rule from the ACL.

Command Mode

IPv6-ACL Configuration

Command Syntax

[**SEQ_NUM**] **deny PROT SRC_ADDR** [**SRC_PT**] **DEST_ADDR** [**DEST_PT**][**FLAG**] [**MSG**][**hop**][**tracked**] [**DSCP_FILTER**] [log]

no deny PROT SRC_ADDR [**SOURCE_PT**] **DEST_ADDR** [**DEST_PT**][**FLAG**][**MSG**][**hop**][**tracked**] [**DSCP_FILTER**][log]

default deny PROT SRC_ADDR [**SOURCE_PT**] **DEST_ADDR** [**DEST_PT**][**FLAG**][**MSG**] [**hop**][**tracked**] [**DSCP_FILTER**][log]

Note: Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a deny rule.

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
    - ***no parameter*** Number is derived by adding ***10*** to the number of the ACL's last rule.
    - ***1 - 4294967295*** Number assigned to entry.
- **PROT** Protocol field filter. Values include:

  - **icmpv6** Internet Control Message Protocol for version 6 (58).
  - **ipv6** Internet Protocol IPv6 (41).
  - **ospf** Open Shortest Path First (89).
  - **tcp** Transmission Control Protocol (6).
  - **udp** User Datagram Protocol (17).
  - ***protocol_num*** Integer corresponding to an IP protocol. Values range from ***0*** to ***255***.
- **SRC_ADDR** and **DEST_ADDR** Source and destination address filters. Options include:
  - ***ipv6_prefix*** IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host *ipv6_addr*** IPv6 host address.
- **SRC_PT** and **DEST_PT** Source and destination port filters. Options include:
  - **any** All ports.
  - **eq *port-1 port-2*** ... ***port-n*** A list of ports. Maximum list size is 10 ports.
  - **neq *port-1 port-2*** ... ***port-n*** The set of all ports not listed. Maximum list size is 10 ports.
  - **gt *port*** The set of ports with larger numbers than the listed port.
  - **lt *port*** The set of ports with smaller numbers than the listed port.
  - **range *port_1 port_2*** The set of ports whose numbers are between the range.
- **HOP** Filters by packet's hop-limit value. Options include:
  - ***no parameter*** Rule does not use hop limit to filter packets.
  - **hop-limit eq *hop_value*** Packets match if **hop-limit** value in packet equals ***hop_value***.
  - **hop-limit gt *hop_value*** Packets match if **hop-limit** in packet is greater than ***hop_value***.
  - **hop-limit lt *hop_value*** Packets match if **hop-limit** in packet is less than ***hop_value***.
  - **hop-limit neq *hop_value*** Packets match if **hop-limit** in packet is not equal to ***hop_value***.
- **FLAG** Flag bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- **MSG** Message type filters (ICMPv6 packets). Use CLI syntax assistance (?) to display options.
- **tracked** Rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- **DSCP_FILTER** Rule filters packet by its DSCP value. Values include:
  - ***no parameter*** Rule does not use DSCP to filter packets.
  - **dscp *dscp_value*** Packets match if DSCP field in packet is equal to ***dscp_value***.
- **log** Triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Example

This command appends a **deny** statement at the end of the ACL. The **deny** statement drops IPv6 packets from ***3710:249a:c643:ef11::/64*** to any host.

```
switch(config)# ipv6 access-list text1
switch(config-acl-text1)# deny ipv6 3710:249a:c643:ef11::/64 any
switch(config-acl-text1)#
```

13.4.9.7 deny (IPv6 Prefix List)

The **deny** command adds a rule to the configuration mode IPv6 prefix list. Route map match commands use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains. Routes are denied access when they match the prefix that a **deny** statement specifies.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode prefix list. The **no** seq (IPv6 Prefix Lists) command also removes the specified rule from the prefix list.

Command Mode

IPv6-pfx Configuration

Command Syntax

[**SEQUENCE**] **deny** ***ipv6_prefix*** [**MASK**]

Parameters

- **SEQUENCE** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding ***10*** to the number of the list's last rule.
  - **seq** ***seq_num*** Number is specified by ***seq_num***. Value ranges from ***0 to 65535***.
- ***ipv6_prefix*** IPv6 prefix upon which command filters routes (CIDR notation).
- **MASK** Range of the prefix to be matched.
  - ***no parameter*** Exact match with the subnet mask is required.
  - **eq** ***mask_e*** Prefix length is equal to ***mask_e***.
  - **ge** ***mask_g*** Range is from ***mask_g*** to ***128***.
  - **le** ***mask_l*** Range is from ***subnet*** mask length to ***mask_l***.
  - **ge** ***mask_l*** **le** ***mask_g*** Range is from ***mask_g*** to ***mask_l***.
  - ***mask_e***, ***mask_land***, and ***mask_g*** range from ***1 to 128***.
  - When **le** and **ge** are specified, ***subnet*** mask ***mask_g mask_l***.

Example

This command appends a **deny** statement at the end of the ***text1*** prefix list. The **deny** statement denies redistribution of routes with the specified prefix.

```
switch(config)# ipv6 prefix-list route-five
switch(config-ipv6-pfx)# deny 3100::/64
switch(config-ipv6-pfx)#
```

13.4.9.8 deny (MAC ACL)

The **deny** command adds a deny rule to the configuration mode MAC Access Control List (ACL). Packets filtered by a deny rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes the specified rule from the ACL.

Command Mode

MAC-ACL Configuration

Command Syntax

[**SEQ_NUM**] **deny SOURCE_ADDR DEST_ADDR** [**PROTOCOL**][log]

no deny SOURCE_ADDR DEST_ADDR [**PROTOCOL**][log]

default deny SOURCE_ADDR DEST_ADDR [**PROTOCOL**][log]

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding ***10*** to the number of the ACL's last rule.
  - ***1 - 4294967295*** Number assigned to entry.

- **SOURCE_ADDR** and **DEST_ADDR** Source and destination address filters. Options include:
  - ***mac_address mac_mask*** MAC address and mask.
  - **any** Packets from all addresses are filtered.
  - ***mac_address*** specifies a MAC address in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh).
  - ***mac_mask*** specifies a MAC address mask in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh).
  - ***0*** bits require an exact match to filter.
  - ***1*** bits filter on any value.
- **PROTOCOL** Protocol field filter. Values include:
  - **aarp** Appletalk Address Resolution Protocol (0x80f3).
  - **appletalk** Appletalk (0x809b).
  - **arp** Address Resolution Protocol (0x806).
  - **ip** Internet Protocol Version 4 (0x800).
  - **ipx** Internet Packet Exchange (0x8137).
  - **lldp** LLDP (0x88cc).
  - **novell** Novell (0x8138).
  - **rarp** Reverse Address Resolution Protocol (0x8035).
  - ***protocol_num*** Integer corresponding to a MAC protocol. Values range from ***0 to 65535***.
- **log** Triggers an informational log message to the console about the matching packet.

Examples

- This command appends a permit statement at the end of the ACL. The deny statement drops all aarp packets from ***10.1000.0000*** through ***10.1000.FFFF*** to any host.

```
switch(config)# mac access-list text1
switch(config-mac-acl-text1)# deny 10.1000.0000 0.0.FFFF any aarp
```

- This command inserts a permit statement with the sequence number ***25***. The deny statement drops all packets through the interface.

```
switch(config-mac-acl-text1)# 25 deny any any
```

13.4.9.9 deny (Standard IPv4 ACL)

The **deny** command adds a deny rule to the configuration mode standard IPv4 Access Control List (ACL). Standard ACL rules filter on the source field.

Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding ***10*** to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes the specified rule from the ACL.

Command Mode

Std-ACL Configuration

Command Syntax

[**SEQ_NUM**] **deny SOURCE_ADDR** [log]

no deny SOURCE_ADDR [log]

default deny SOURCE_ADDR [log]

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:

- ***no parameter*** Number is derived by adding ***10*** to the number of the ACL's last rule.
- ***1 - 4294967295*** Number assigned to entry.

- **SOURCE_ADDR** Source address filter. Options include:
  - **network_addr** Subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host *ip_addr*** IP address (dotted decimal notation).

    Subnet addresses support discontiguous masks.
- **log** Triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Example

This command appends a **deny** statement at the end of the ACL. The **deny** statement drops packets from ***10.10.1.1/24***.

```
switch(config)# ip access-list standard text1
switch(config-std-acl-text1)# deny 10.1.1.1/24
switch(config-std-acl-text1)#
```

13.4.9.10 deny (Standard IPv6 ACL)

The **deny** command adds a deny rule to the configuration mode standard IPv6 Access Control List (ACL). Standard ACL rules filter on the source field.

Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding ***10*** to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes the specified rule from the ACL.

Command Mode

Std-IPv6-ACL Configuration

Command Syntax

[**SEQ_NUM**] **deny SOURCE_ADDR**

no deny SOURCE_ADDR

default deny SOURCE_ADDR

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding ***10*** to the number of the ACL's last rule.
  - ***1 - 4294967295*** Number assigned to entry.
- **SOURCE_ADDR** Source address filter. Options include:
  - ***ipv6_prefix*** IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host *ipv6_addr*** IPv6 host address.

Example

This command appends a **deny** statement at the end of the ACL. The **deny** statement drops packets from ***2103::/64***.

```
switch(config)# ipv6 access-list standard text1
switch(config-std-acl-ipv6-text1)# deny 2103::/64
switch(config-std-acl-ipv6-text1)#
```

13.4.9.11 description (route map)

The **description** command adds a text string to the configuration mode route map. The string has no functional impact on the route map.

The **no description** and **default description** commands remove the text string from the configuration mode route map by deleting the corresponding **description** command from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

description ***label_text***

no description

default description

Parameters

label_text Character string assigned to the route map configuration.

Related Command

route-map

Example

These commands add description text to the ***XYZ-1*** route map.

```
switch(config)# route-map XYZ-1
switch(config-route-map-XYZ-1)# description This is the first map.
switch(config-route-map-XYZ-1)# exit
switch(config)# show route-map XYZ-1
route-map XYZ-1 permit 10
  Description:
    description This is the first map.
  Match clauses:
  Set clauses:
switch(config)#
```

13.4.9.12 hardware access-list resource sharing vlan in

The **hardware access-list resource sharing vlan in** command enables the IPv4 Ingress Sharing of hardware resources on the switch same ACL is applied on different VLANs.

The **no hardware access-list resource sharing vlan in** command disables the IPv4 Ingress Sharing of hardware resources on the switch.

Command Mode

Global Configuration

Command Syntax

hardware access-list resource sharing vlan in

no hardware access-list resource sharing vlan in

Guidelines

- This command is compatible only with the DCS-7010 and DCS-7050x series switches.
- Enabling IPv4 Ingress Sharing requires the restart of software agents on the platform. This is a disruptive process and impacts traffic forwarding.

Use the **show platform trident** command to verify the Ingress IPv4 Sharing information.

13.4.9.13 hardware access-list resource sharing vlan ipv4 out

The **hardware access-list resource sharing vlan ipv4 out** command enables the IPv4 Egress RACL TCAM sharing on the switch.

The **no hardware access-list resource sharing vlan ipv4 out** command disables the IPv4 Egress RACL TCAM sharing on the switch. By default, the IPv4 Egress RACL sharing is enabled on the switch.

Command Mode

Global Configuration

Command Syntax

hardware access-list resource sharing vlan ipv4 out

no hardware access-list resource sharing vlan ipv4 out

Guidelines

- This command is compatible only with the DCS-7280E and DCS-7500E series switches.
- Disabling IPv4 RACL sharing requires the restart of software agents on the platform. This is a disruptive process and impacts traffic forwarding.
- Enabling IPv4 RACL sharing, if previously disabled from the default configuration, requires the restart of software agents on the platform. This is a disruptive process and impacts traffic forwarding. Enabling IPv4 RACL sharing if uRPF is configured will disable uRPF.
- Use the **show running-config all | include sharing** command to verify whether or not sharing for egress IPv4 RACLs is enabled.

Example

This command verifies whether IPv4 RACL sharing is enabled or disabled.

```
switch# show running-config all | include sharing

hardware access-list resource sharing vlan ipv4 out
                         ---->It returns the following output if IPv4 RACL
 sharing is enabled.
```

13.4.9.14 hardware access-list update default-result permit

The **hardware access-list update default-result permit** command configures the switch to permit all traffic on Ethernet and VLAN interfaces with ACLs applied to them while those ACLs are being modified. Traffic is permitted when the ACL is available for modification using one of the **ip access-list** commands, and ends when the ACL configuration mode is exited and rules are populated in hardware. This command is disabled by default.

The **no hardware access-list update default-result permit** and **default hardware access-list update default-result permit** commands restore the switch to its default state (blocking traffic during ACL modifications) by removing the corresponding **hardware access-list update default-result permit** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

hardware access-list update default-result permit

no hardware access-list update default-result permit

default hardware access-list update default-result permit

Restrictions

This command is available on the Arista 7050X, 7060X, 7150, 7250X, 7280, 7280R, 7300X, 7320X, and 7500 series switches.

This command does not support egress ACLs.

While this command is enabled, static NAT, and ACL-based mirroring are affected during ACL updates.

Example

This command configures a 7150 series switch to permit all traffic on Ethernet and VLAN interfaces with ACLs applied to them while those ACLs are being modified.

```
switch(config)# hardware access-list update default-result permit
switch(config)#
```

13.4.9.15 hardware counter feature acl out

The **hardware counter feature acl out** command enables egress ACL hardware counters for IPv4 or IPv6, which count the number of packets hitting rules associated with egress ACLs applied to various interfaces on a switch.

The **no hardware counter feature acl out** and **default hardware counter feature acl out** commands disable or return the egress ACL hardware counters to the default state.

Command Mode

Global Configuration

Command Syntax

hardware counter feature acl out [**OPTIONS**]

no hardware counter feature acl out [**OPTIONS**]

default hardware counter feature acl out [**OPTIONS**]

Parameters

- **OPTIONS** ACL hardware counter options include:
  - **ipv4** Address family IPv4.
  - **ipv6** Address family IPv6.

Examples

- This command enables IPv4 egress ACL hardware counters.

```
switch(config)# hardware counter feature acl out ipv4
switch(config)#
```

- This command disables IPv4 egress ACL hardware counters.

```
switch(config)# no hardware counter feature acl out ipv4
```

```
switch(config)#
```

13.4.9.16 ip access-group (Service ACLs)

The **ip access-group** (Service ACLs) command configures a Service ACL to be applied by a control-plane service. The service is specified by the command mod (Service ACLs)e in which the Service ACL is applied.

The **no ip access-group** (Service ACLs) and **default ip access-group** (Service ACLs) commands remove the corresponding **ip access-group** (Service ACLs) command from ***running-config***.

Command Mode

Mgmt-SSH Configuration

Mgmt-API Configuration

Router-BGP Configuration

Router-OSPF Configuration

Router-IGMP Configuration

MPLS-LDP Configuration

Queue-Monitor-Streaming Configuration

MPLS-Ping Configuration

Mgmt-Telnet Configuration

Command Syntax

ip access-group ***acl_name*** [vrf***vrf_name***][in]

no ip access-group ***acl_name*** [vrf***vrf_name***][in]

default ip access-group ***acl_name*** [vrf***vrf_name***][in]

Parameters

Parameters vary by process.

- ***acl_name*** Name of the Service ACL assigned to control-plane service.
- **vrf** ***vrf_name*** Specifies the VRF in which the Service ACL is to be applied.
- **in** Specifies inbound connections or packets only (keyword required for SSH and Telnet services).

Example

These commands apply the Service ACL ***bgpacl*** to the BGP routing protocol in VRF ***purple***.

```
(config)# router bgp 5
(config-router-bgp)# vrf purple
(config-router-bgp-vrf-purple)# ip access-group bgpacl
```

For additional configuration examples, see **Configuring Service ACLs and Displaying Status and Counters**.

13.4.9.17 ip access-group

The **ip access-group** command applies an IPv4 or standard IPv4 Access Control List (ACL) to the configuration mode interface or subinterface.

The **no ip access-group** and **default ip access-group** commands remove the corresponding **ip access-group** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip access-group` ***list_name*** **DIRECTION**

`no ip access-group` ***list_name*** **DIRECTION**

`default ip access-group` ***list_name*** **DIRECTION**

Parameters

- ***list_name*** Name of ACL assigned to interface.
- **DIRECTION** Transmission direction of packets, relative to interface. Valid options include:
  - **in** Inbound packets.
  - **out** Outbound packets.

Restrictions

Filtering of outbound packets by ACLs is not supported on Petra platform switches.

Filtering of outbound packets by ACLs on FM6000 switches is supported on physical interfaces only (Ethernet and port channels).

ACLs on sub-interfaces are supported on DCS-7280E, DCS-7500E, DCS-7280R, and DCS-7500R.

Example

These commands apply the IPv4 ACL named ***test2*** to ***interface ethernet 3***.

```
switch(config)# interface ethernet 3
switch(config-if-Et3)# ip access-group test2 in
switch(config-if-Et3)#
```

13.4.9.18 ip access-list

The **`ip access-list`** command places the switch in ACL configuration mode, which is a group change mode that modifies an IPv4 access control list. The command specifies the name of the IPv4 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **`exit`** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **`abort`** command discards pending ACL changes, returning the switch to global configuration mode.

The **`no ip access-list`** and **`default ip access-list`** commands delete the specified IPv4 ACL.

Command Mode

Global Configuration

Command Syntax

`ip access-list` ***list_name***

`no ip access-list` ***list_name***

`default ip access-list` ***list_name***

Parameters

list_name Name of ACL. Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in ACL configuration mode:

- deny (IPv4 ACL)
- no <sequence number> (ACLs)
- permit (IPv4 ACL)
- remark
- resequence (ACLs)
- show (ACL configuration modes)

Related Commands:

- ip access-list standard Enters ***std-acl*** configuration mode for editing standard IP ACLs.
- show ip access-lists Displays IP and standard ACLs.

Examples

- This command places the switch in ACL configuration mode to modify the ***filter1*** IPv4 ACL.

```
switch(config)# ip access-list filter1
switch(config-acl-filter1)#
```

- This command saves changes to ***filter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-acl-filter1)# exit
switch(config)#
```

- This command discards changes to ***filter1***, then returns the switch to global configuration mode.

```
switch(config-acl-filter1)# abort
switch(config)#
```

13.4.9.19 ip access-list standard

The **`ip access-list standard`** command places the switch in std-ACL configuration mode, which is a group change mode that modifies a standard IPv4 access control list. The command specifies the name of the standard IPv4 ACL that subsequent commands modify, and creates an ACL if it references a nonexistent list. All group change mode edit session changes are pending until the session ends.

The **`exit`** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. Pending changes are also saved by entering a different configuration mode.

The **`abort`** command discards pending ACL changes, returning the switch to global configuration mode.

The **`no ip access-list standard`** and **`default ip access-list standard`** commands delete the specified ACL.

Command Mode

Global Configuration

Command Syntax

`ip access-list standard` ***list_name***

`no ip access-list standard` ***list_name***

`default ip access-list standard` ***list_name***

Parameters

list_name Name of standard ACL. Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in std-ACL configuration mode:

- deny (Standard IPv4 ACL)
- no <sequence number> (ACLs)
- permit (Standard IPv4 ACL)
- remark
- resequence (ACLs)
- show (ACL configuration modes)

Related Commands

- ip access-list Enters ACL configuration mode for editing IPv4 ACLs.
- show ip access-lists Displays IPv4 and standard IPv4 ACLs.

Examples

- This command places the switch in std-ACL configuration mode to modify the ***filter2*** IPv4 ACL.

```
switch(config)# ip access-list standard filter2
switch(config-std-acl-filter2)#
```

- This command saves changes to ***filter2*** ACL, then returns the switch to global configuration mode.

```
switch(config-std-acl-filter2)# exit
switch(config)#
```

- This command discards changes to ***filter2***, then returns the switch to global configuration mode.

```
switch(config-std-acl-filter2)# abort
switch(config)#
```

13.4.9.20 ip prefix-list

The **`ip prefix-list`** command creates a prefix list or adds an entry to an existing list. Route map match commands use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains.

A prefix list comprises all prefix list entries with the same label. The sequence numbers of the rules in a prefix list specify the order that the rules are applied to a route that the match command is evaluating.

The **`no ip prefix-list`** and **`default ip prefix-list`** commands delete the specified prefix list entry by removing the corresponding **`ip prefix-list`** statement from ***running-config***. If the **`no`** or **`default ip prefix-list`** command does not list a sequence number, the command deletes all entries of the prefix list.

Command Mode

Global Configuration

Command Syntax

`ip prefix-list` ***list_name*** [**SEQUENCE**] **FILTER_TYPE** ***network_addr*** [**MASK**]

`no ip prefix-list` ***list_name*** [**SEQUENCE**]

`default ip prefix-list` ***list_name*** [**SEQUENCE**]

Parameters

- ***list_name*** The label that identifies the prefix list.
- **SEQUENCE** Sequence number of the prefix list entry. Options include:
  - ***no parameter*** Entry's number is ten plus highest sequence number in current list.
  - **seq** ***seq_num*** Number assigned to entry. Value ranges from ***0*** to ***65535***.
- **FILTER_TYPE** Specifies route access when it matches IP prefix list. Options include:
  - **permit** Routes are permitted access when they match the specified subnet.
  - **deny** Routes are denied access when they match the specified subnet.

- ***network_addr*** Subnet upon which command filters routes. Format is CIDR or address-mask.
- **MASK** Range of the prefix to be matched.
  - ***no parameter*** Exact match with the subnet mask is required.
  - **eq *mask_e*** Prefix length is equal to ***mask_e***.
  - **ge *mask_g*** Range is from ***1*** to ***32***.
  - **le *mask_l*** Range is from ***subnet*** mask length to ***mask_l***.
  - **ge *mask_l* le *mask_g*** Range is from ***mask_g*** to ***mask_l***.
  - ***mask_e***, ***mask_l***, and ***mask_g*** range from ***1*** to ***32***. When **le** and **ge** are specified, ***subnet* mask *mask_g*>*mask_l***.

Example

- This command places the switch in IPv4 prefix list configuration mode to create an IPv4 prefix list named ***route-one***.

```
switch(config)# ip prefix-list route-one
switch(config-ip-pfx)#
```

- These commands create four different rules for the prefix-list named ***route-one***.

```
switch(config)# ip prefix-list route-one
switch(config-ip-pfx)# seq 10 deny 10.1.1.0/24
switch(config-ip-pfx)# seq 20 deny 10.1.0.0/16
switch(config-ip-pfx)# seq 30 permit 12.15.4.9/32
switch(config-ip-pfx)# seq 40 deny 1.1.1.0/24
```

13.4.9.21 ipv6 access-group

The **ipv6 access-group** command applies an IPv6 or standard IPv6 Access Control List (ACL) to the configuration mode interface.

The **no ipv6 access-group** and **default ipv6 access-group** commands remove the corresponding **ipv6 access-group** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ipv6 access-group *list_name* DIRECTION

no ipv6 access-group *list_name* DIRECTION

default ipv6 access-group *list_name* DIRECTION

Parameters

- ***list_name*** Name of ACL assigned to interface.
- **DIRECTION** Transmission direction of packets, relative to interface. Valid options include:
  - **in** Inbound packets.
  - **out** Outbound packets.

Examples

These commands assign the IPv6 ACL named ***test2*** to the ***interface ethernet 3***.

```
switch(config)# interface ethernet 3
```

```
switch(config-if-Et3)# ipv6 access-group test2 in
switch(config-if-Et3)#
```

13.4.9.22 ipv6 access-group (Service ACLs)

The **ipv6 access-group** (Service ACLs) command configures an IPv6 or standard IPv6 Service ACL to be applied by a control-plane service. The service is specified by the command mode in which the Service ACL is applied.

The **no ipv6 access-group** (Service ACLs) and **default ipv6 access-group** (Service ACLs) commands remove the corresponding **ipv6 access-group** (Service ACLs) command from ***running-config***.

Command Mode

Mgmt-SSH Configuration

Mgmt-API Configuration

Router-BGP Configuration

Router-OSPF Configuration

MPLS-LDP Configuration

Queue-Monitor-Streaming Configuration

MPLS-Ping Configuration

Mgmt-Telnet Configuration

Command Syntax

ipv6 access-group ***ipv6_acl_name*** [vrf***vrf_name***][in]

no ipv6 access-group [***ipv6_acl_name***][vrf***vrf_name***][in]

default ipv6 access-group ***ipv6_acl_name*** [vrf ***vrf_name***][in]

Parameters

Parameters vary by process.

- ***ipv6_acl_name*** Name of the IPv6 Service ACL assigned to control-plane service.
- **vrf** ***vrf_name*** Specifies the VRF in which the Service ACL is to be applied.
- **in** Specifies inbound connections or packets only (keyword required for SSH and Telnet services).

Example

These commands apply the IPv6 Service ACL ***bgpacl*** to the BGP routing protocol in VRF ***purple***.

```
(config)# router bgp 5
(config-router-bgp)# vrf purple
(config-router-bgp-vrf-purple)# ipv6 access-group bgpacl
```

For additional configuration examples, see **Configuring Service ACLs and Displaying Status and Counters**.

13.4.9.23 ipv6 access-list

The **ipv6 access-list** command places the switch in ***IPv6-ACL*** configuration mode, which is a group change mode that modifies an IPv6 access control list. The command specifies the name of the IPv6 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ipv6 access-list** and **default ipv6 access-list** commands delete the specified IPv6 ACL.

Command Mode

Global Configuration

Command Syntax

ipv6 access-list ***list_name***

no ipv6 access-list ***list_name***

default ipv6 access-list ***list_name***

Parameters

list_name Name of ACL. Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in IPv6-ACL configuration mode:

- deny (IPv6 ACL)
- no <sequence number> (ACLs)
- permit (IPv6 ACL)
- remark
- resequence (ACLs)
- show (ACL configuration modes)

Related Commands

- ipv6 access-list standard Enters ***std-ipv6-acl*** configuration mode for editing standard IPv6 ACLs.
- show ipv6 access-lists Displays IPv6 and standard IPv6 ACLs.

Examples

- This command places the switch in IPv6-ACL configuration mode to modify the ***filter1*** IPv6 ACL.

```
switch(config)# ipv6 access-list filter1
switch(config-ipv6-acl-filter1)#
```

- This command saves changes to ***filter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-ipv6-acl-filter1)# exit
switch(config)#
```

- This command discards changes to ***filter1***, then returns the switch to global configuration mode.

```
switch(config-ipv6-acl-filter1)# abort
switch(config)#
```

13.4.9.24 ipv6 access-list standard

The **ipv6 access-list standard** command places the switch in std-IPv6-ACL-configuration mode, which is a group change mode that modifies a standard IPv6 access control list. The command specifies the name of the standard IPv6 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All group change mode edit session changes are pending until the session ends.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. Pending changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ipv6 access-list standard** and **default ipv6 access-list standard** commands delete the specified ACL.

Command Mode

Global Configuration

Command Syntax

ipv6 access-list standard ***list_name***

no ipv6 access-list standard ***list_name***

default ipv6 access-list standard ***list_name***

Parameters

list_name Name of ACL. Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in std-IPv6-ACL configuration mode:

- deny (Standard IPv6 ACL)
- no <sequence number> (ACLs)
- permit (Standard IPv6 ACL)
- remark
- resequence (ACLs)
- show (ACL configuration modes)

Related Commands

- ipv6 access-list Enters IPv6-ACL configuration mode for editing IPv6 ACLs.
- show ipv6 access-lists Displays IPv6 and standard IPv6 ACLs.

Examples

- This command places the switch in Std-IPv6 ACL configuration mode to modify the ***filter2*** ACL.

```
switch(config)# ipv6 access-list standard filter2
switch(config-std-ipv6-acl-filter2)#
```

- This command saves changes to ***filter2*** ACL, then returns the switch to global configuration mode.

```
switch(config-std-ipv6-acl-filter2)# exit
switch(config)#
```

- This command discards changes to ***filter2***, then returns the switch to global configuration mode.

```
switch(config-std-ipv6-acl-filter2)# abort
switch(config)#
```

13.4.9.25 ipv6 prefix-list

The **ip prefix-list** command places the switch in ***IPv6 prefix-list*** configuration mode, which is a group change mode that modifies an IPv6 prefix list. The command specifies the name of the IPv6 prefix list that subsequent commands modify and creates a prefix list if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending prefix list changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no ipv6 prefix-list** and **default ipv6 prefix-list** commands delete the specified IPv6 prefix list.

Command Mode

Global Configuration

Command Syntax

ipv6 prefix-list ***list_name***

no ipv6 prefix-list ***list_name***

default ipv6 prefix-list ***list_name***

Parameter

list_name Name of prefix list. Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in IPv6-pfx configuration mode:

- deny (IPv6 Prefix List)
- permit (IPv6 Prefix List)
- seq (IPv6 Prefix Lists)

Examples

- This command places the switch in ***IPv6 prefix-list*** configuration mode to modify the ***route-five*** prefix list.

```
switch(config)# ipv6 prefix-list route-five
switch(config-ipv6-pfx)#
```

- This command saves changes to the prefix list, then returns the switch to global configuration mode.

```
switch(config-ipv6-pfx)# exit
switch(config)#
```

- This command saves changes to the prefix list, then places the switch in ***interface-ethernet*** mode.

```
switch(config-ipv6-pfx)# interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to the prefix list, then returns the switch to global configuration mode.

```
switch(config-ipv6-pfx)# abort
switch(config)#
```

13.4.9.26 mac access-group

The **mac access-group** command applies a MAC Access Control List (MAC ACL) to the configuration mode interface.

The **no mac access-group** and **default mac access-group** commands remove the specified **mac access-group** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

mac access-group ***list_name*** **DIRECTION**

no mac access-group ***list_name*** **DIRECTION**

`default mac access-group` *list_name* DIRECTION

Parameters

- ***list_name*** Name of MAC ACL.
- **DIRECTION** Transmission direction of packets, relative to interface. Valid options include:
  - **in** Inbound packets.
  - **out** Outbound packets.

Restrictions

Filtering of outbound packets by MAC ACLs is supported only on Helix, Trident, and Trident II platform switches.

Example

These commands assign the MAC ACL named ***mtest2*** to ***interface ethernet 3*** to filter inbound packets.

```
switch(config)# interface ethernet 3
switch(config-if-Et3)# mac access-group mtest2 in
switch(config-if-Et3)#
```

13.4.9.27 mac access-list

The **`mac access-list`** command places the switch in ***MAC-ACL*** configuration mode, which is a group change mode that modifies a MAC access control list. The command specifies the name of the MAC ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **`exit`** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **`abort`** command discards pending ACL changes, returning the switch to global configuration mode.

The **`no mac access-list`** and **`default mac access-list`** commands delete the specified list.

Command Mode

Global Configuration

Command Syntax

`mac access-list` *list_name*

`no mac access-list` *list_name*

`default mac access-list` *list_name*

Parameters

list_name Name of MAC ACL. Names must begin with an alphabetic character and cannot contain a space or quotation mark.

Commands Available in MAC-ACL configuration mode:

- deny (MAC ACL)
- no <sequence number> (ACLs)
- permit (MAC ACL)
- remark
- resequence (ACLs)
- show (ACL configuration modes)

Examples

- This command places the switch in ***MAC-ACL*** configuration mode to modify the ***mfilter1*** MAC ACL.

```
switch(config)# mac access-list mfilter1
switch(config-mac-acl-mfilter1)#
```

- This command saves changes to ***mfilter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-mac-acl-mfilter1)# exit
switch(config)#
```

- This command saves changes to ***mfilter1*** ACL, then places the switch in ***interface-ethernet*** mode.

```
switch(config-mac-acl-mfilter1)# interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to ***mfilter1***, then returns the switch to global configuration mode.

```
switch(config-mac-acl-mfilter1)# abort
switch(config)#
```

13.4.9.28 match (route-map)

The **match** command creates a route map statement entry that specifies one route filtering command. When a statement contains multiple match commands, the permit or deny filter applies to a route only if its properties are equal to corresponding parameters in each **match** command. When a route properties do not equal the command parameters, the route is evaluated against the next statement in the route map, as determined by sequence number. If all statements fail to permit or deny the route, the route is denied.

The **no match** and **default match** commands remove the **match** command from the configuration mode route map statement by deleting the corresponding command from ***running-config***.

Note: The route map configuration supports only standard ACL.

Command Mode

Route-Map Configuration

Command Syntax

match CONDITION

no match CONDITION

default match CONDITION

Parameters

- **CONDITION** Specifies criteria for evaluating a route. Options include:
  - **aggregate-role** Role in BGP contributor-aggregate relation. Options include:
    - **contributor** BGP aggregate's contributor.
    - **aggregate-attributes** Route map to apply against the aggregate route.
  - **as *1*** to ***4294967295*** BGP Autonomous System number.
  - **as-path *path_name*** BGP Autonomous System path access list.
  - **community NAME** BGP community. Options for **NAME** include:
    - **listname** BGP community.
    - ***listname* exact-match** BGP community; list must match set that is present.
  - **extcommunity *listname*** BGP extended community. Options for **NAME** include:
    - ***listname*** BGP community.
    - ***listname* exact-match** BGP community; list must match set that is present.

- ***interface* INTF_NAME** Specifies an interface. Options for **INTF_NAME** include:
  - **ethernet *e_num*** Ethernet interface.
  - **loopback *l_num*** Loopback interface.
  - **port-channel *p_num*** Port channel interface.
  - **vlan *v_num*** VLAN interface.

  invert-result Invert sub route map result.
- **ip address LIST** IPv4 address filtered by an ACL or prefix list. **LIST** options include:
  - ***access-list acl_name*** IPv4 address filtered by access control list (ACL).
  - **prefix-list *plv4_name*** IPv4 address filtered by IP prefix list.
- **ip next-hop prefix-list *plv4_name*** IPv4 next-hop filtered by IP prefix list.
- **ip resolved-next-hop prefix-list *plv4_name*** IPv4 resolved nexthop filtered by IP prefix list.
- **ipv6 address prefix-list *plv6_name*** IPv6 address filtered by IPv6 prefix list.
- **ipv6 next-hop prefix-list *plv6_name*** IPv6 next-hop filtered by IPv6 prefix list.
- **ipv6 resolved-next-hop prefix-list *plv6_name*** IPv6 resolved nexthop filtered by IPv6 prefix list.
- **local-preference *1*** to ***4294967295*** BGP local preference metric.
- **metric *1*** to ***4294967295*** Route metric.
- **metric-type OSPF_TYPE** OSPF metric type. Options include:
  - **type-1** OSPF type 1 metric.
  - **type-2** OSPF type 2 metric.
- **source-protocol *protocol_type*** Routing protocol of route's source. Options include:
  - **bgp**
  - **connected**
  - **ospf**
  - **rip**
  - **static**
- **tag *1*** to ***4294967295*** Route tag.

Related Command

route-map

Example

This command creates a ***route map*** match rule that filters routes from BGP ***as 15***.

```
switch(config)# route-map map1
switch(config-route-map-map1)# match as 15
switch(config-route-map-map1)#
```

13.4.9.29 no <sequence number> (ACLs)

The **no <sequence number>** command removes the rule with the specified sequence number from the ACL. The **default <sequence number>** command also removes the specified rule.

Command Mode

ACL Configuration

IPv6-ACL Configuration

Std-ACL Configuration

Std-IPv6-ACL Configuration

MAC-ACL Configuration

Command Syntax

no ***line_num***

default ***line_num***

Parameters

line_num Sequence number of rule to be deleted. Values range from ***1 - 4294967295***.

Example

This command removes statement ***30*** from the list.

```
switch(config-acl-test1)# show IP Access List test1
  10 permit ip 10.10.10.0/24 any
  20 permit ip any host 10.20.10.1
  30 deny ip host 10.10.10.1 host 10.20.10.1
  40 permit ip any any
  50 remark end of list
switch(config-acl-test1)# no 30
switch(config-acl-test1)# show IP Access List test1
  10 permit ip 10.10.10.0/24 any
  20 permit ip any host 10.20.10.1
  40 permit ip any any
  50 remark end of list
```

13.4.9.30 permit (IPv4 ACL)

The **permit** command adds a permit rule to the configuration mode IPv4 Access Control List (ACL). Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes a specified rule from the ACL.

Command Mode

ACL Configuration

Command Syntax

[**SEQ_NUM**] **permit PROTOCOL SOURCE_ADDR** [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**] [**FLAGS**][**MESSAGE**][fragments][tracked][**DSCP_FILTER**][**TTL_FILTER**][log]

no permit PROTOCOL SOURCE_ADDR [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**][**FLAGS** [**MESSAGE**] [fragments] [tracked][**DSCP_FILTER**][**TTL_FILTER**][log]

default permit PROTOCOL SOURCE_ADDR [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**][**FLAGS**] [**MESSAGE**][fragments][tracked][**DSCP_FILTER**][**TTL_FILTER**][log]

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a permit rule.

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding ***10*** to the number of the ACL's last rule.
  - ***1 - 4294967295*** Number assigned to entry.
- **PROTOCOL** Protocol field filter. Values include:
  - **ahp** Authentication Header Protocol (51).
  - **gre** Generic Routing Encapsulation.

    - **gtp** GPRS Tunneling Protocol.
    - **icmp** Internet Control Message Protocol (1).
    - **igmp** Internet Group Management Protocol (2).
    - **ip** Any Internet Protocol v4 (4).
    - **ospf** Open Shortest Path First (89).
    - **pim** Protocol Independent Multicast (103).
    - **tcp** Transmission Control Protocol (6).
    - **udp** User datagram protocol (17).
    - **vlan** Enter VLAN number and mask. VLAN value ranges from 1 to 4094; mask value ranges from 0x000-0xFFF .
    - **vrrp** Virtual Router Redundancy Protocol (112).
    - ***protocol_num*** Integer corresponding to an IP protocol. Values range from ***0 to 255***.
- **SOURCE_ADDR** and **DEST_ADDR** Source and destination address filters. Options include:
    - ***network_addr*** subnet address (CIDR or address-mask).
    - **any** Packets from all addresses are filtered.
    - **host *ip_addr*** IP address (dotted decimal notation).

    Source and destination subnet addresses support discontiguous masks.
- **SOURCE_PORT** and **DEST_PORT** Source and destination port filters. Options include:
    - **any** All ports.
    - **eq *port-1 port-2*** ... ***port-n*** A list of ports. Maximum list size is 10 ports.
    - **neq *port-1 port-2*** ... ***port-n*** The set of all ports not listed. Maximum list size is 10 ports.
    - **gt *port*** The set of ports with larger numbers than the listed port.
    - **lt *port*** The set of ports with smaller numbers than the listed port.
    - **range *port_1 port_2*** The set of ports whose numbers are between the range.
- **fragments** Filters packets with FO bit set (indicates a non-initial fragment packet).
- **FLAGS** Flags bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- **MESSAGE** Message type filters (ICMP packets). Use CLI syntax assistance (?) to display options.
- **tracked** Rule filters packets in existing ICMP, UDP, or TCP connections.
    - Valid in ACLs applied to the control plane.
    - Validity in ACLs applied to data plane varies by switch platform.
- **DSCP_FILTER** Rule filters packet by its DSCP value. Values include:
    - ***no parameter*** Rule does not use DSCP to filter packets.
    - ***dscp dscp_value*** Packets match if DSCP field in packet is equal to ***dscp_value***.
- **TTL_FILTER** Rule filters packet by its TTL (time-to-live) value. Values include:
    - **ttl eq *ttl_value*** Packets match if **ttl** in packet is equal to ***ttl_value***.
    - **ttl gt *ttl_value*** Packets match if **ttl** in packet is greater than ***ttl_value***.
    - **ttl lt *ttl_value*** Packets match if **ttl** in packet is less than ***ttl_value***.
    - **ttl neq *ttl_value*** Packets match if **ttl** in packet is not equal to ***ttl_value***.
        - Valid in ACLs applied to the control plane.
        - Validity in ACLs applied to data plane varies by switch platform.
- **log** Triggers an informational log message to the console about the matching packet.
    - Valid in ACLs applied to the control plane.
    - Validity in ACLs applied to data plane varies by switch platform.

Examples

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all OSPF packets from ***10.10.1.1/24*** to any host.

```
switch(config)# ip access-list text1
switch(config-acl-text1)# permit ospf 10.1.1.0/24 any
switch(config-acl-text1)#
```

- This command inserts a **permit** statement with the sequence number ***25***. The **permit** statement passes all PIM packets through the interface.

```
switch(config-acl-text1)# 25 permit pim any any
switch(config-acl-text1)#
```

- These commands configure ACL to permit VLAN traffic between any source and destination host.

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# permit vlan 1234 0x0 ip any any
```

13.4.9.31 permit (IPv6 ACL)

The **permit** command adds a permit rule to the configuration mode IPv6 Access Control List (ACL). Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes a specified rule from the ACL.

Command Mode

IPv6-ACL Configuration

Command Syntax

[**SEQ_NUM**] **permit PROT SRC_ADDR** [**SRC_PT**] **DEST_ADDR** [**DEST_PT**] [**FLAG**] [**MSG**] [**HOP**] [tracked] [**DSCP_FILTER**] [**FLOW_LABEL**] [log]

no permit PROT SRC_ADDR [**SRC_PT**] **DEST_ADDR** [**DEST_PT**] [**FLAG**] [**MSG**] [**HOP**] [tracked] [**DSCP_FILTER**] [**FLOW_LABEL**] [log]

default permit PROT SRC_ADDR [**SRC_PT**] **DEST_ADDR** [**DEST_PT**] [**FLAG**] [**MSG**] [**HOP**] [tracked] [**DSCP_FILTER**] [**FLOW_LABEL**] [log]

Note: Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a permit rule.

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - *no parameter* Number is derived by adding 10 to the number of the ACL's last rule.
  - *1 - 4294967295* Number assigned to entry.
- **PROT** Protocol field filter. Values include:
  - **icmpv6** Internet Control Message Protocol for v6 (58).
  - **ipv6** Internet Protocol IPv6 (41).
  - **ospf** Open Shortest Path First (89).
  - **tcp** Transmission Control Protocol (6).
  - **udp** User Datagram Protocol (17).
  - **vlan** Enter VLAN number. Value ranges from 1 to 4094.
  - *protocol_num* Integer corresponding to an IP protocol. Values range from 0 to 255.
- **SRC_ADDR** and **DEST_ADDR** Source and destination address filters. Options include:

  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.
- **SRC_PT** and **DEST_PT** Source and destination port filters. Options include:
  - **any** All ports.
  - **eq** *port-1* **port-2** ... *port-n* A list of ports. Maximum list size is 10 ports.
  - **neq** *port-1 port-2* ... *port-n* The set of all ports not listed. Maximum list size is 10 ports.
  - **gt** *port* The set of ports with larger numbers than the listed port.
  - **lt** *port* The set of ports with smaller numbers than the listed port.
  - **range** *port_1 port_2* The set of ports whose numbers are in the range.
- **HOP** The rule filters by packet's hop-limit value. Options include:
  - *no parameter* The rule does not use hop limit to filter packets.
  - **hop-limit eq** *hop_value* Packets match if **hop-limit** value in packet equals *hop_value*.
  - **hop-limit gt** *hop_value* Packets match if **hop-limit** in packet is greater than *hop_value*.
  - **hop-limit lt** *hop_value* Packets match if **hop-limit** in packet is less than *hop_value*.
  - **hop-limit neq** *hop_value* Packets match if **hop-limit** in packet is not equal to *hop_value*.
- **FLAG** Flag bit filters (TCP packets). Use CLI syntax assistance (**?**) to display options.
- **MSG** Message type filters (ICMPv6 packets). Use CLI syntax assistance (**?**) to display options.
- **tracked** The rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- **DSCP_FILTER** The rule filters packet by its DSCP value. Values include:
  - *no parameter* The rule does not use DSCP to filter packets.
  - **dscp** *dscp_value* Packets match if DSCP field in packet is equal to *dscp_value*.
- **FLOW_LABEL** The rule permits packets with IPv6 flow labels matching an exact value or a pattern based on a mask. Options include:
  - *no parameter* The rule does not use IPv6 flow labels to filter packets.
  - **flow-label eq** *ipv6_flow_label* The IPv6 flow label must exactly match *ipv6_flow_label*. Flow labels can range from 0 to 1048575.
  - **flow-label** *ipv6_flow_label flow_label_mask* The IPv6 flow label must match a pattern defined by *ipv6_flow_label* and *flow_label_mask*. The mask is an inverse mask. Where the mask has a 0 bit, the flow label must match the *ipv6_flow_label* value, and where the mask has a 1 bit, the corresponding bit in the flow label is ignored. For example, if *ipv6_flow_label* is 10 (0b01010 in binary) and *flow_label_mask* is 0x14 (0b10100 in binary), the rule will match flow labels described by 0b.1.10 (where "." is a wildcard and can be either 0 or 1); the flow labels that will match are 10 (0b01010), 14 (0b0110), 26 (0b11010), and 30 (0b1110). Flow labels can range from 0 to 1048575 and flow label masks can range from 0x00000 to 0xfffff.
- **log** Pass an informational log message to the console when a packet matches.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Examples

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all IPv6 packets with the source address 3710:249a:c643:ef11::/64 and with any destination address.

```
switch(config)#ipv6 access-list acl1
switch(config-acl-acl1)#permit ipv6 3710:249a:c643:ef11::/64 any
switch(config-acl-acl1)#exit
switch(config)#
```

- These commands configure ACL to permit VLAN traffic between any source and destination host.

```
switch(config)#ip access-list acl2
switch(config-acl-acl2)#permit ipv6 vlan 1234 0x0 ip any any
switch(config-acl-acl2)#exit
switch(config)#
```

- These commands add a rule to permit all IPv6 packets with flow label 23.

```
switch(config)#ipv6 access-list acl3
switch(config-acl-acl3)#permit ipv6 any any flow-label eq 23
switch(config-acl-acl3)#exit
switch(config)#
```

- These commands create a rule to permit all IPv6 packets matched by the flow label 23 and the mask 0x5678.

```
switch(config)#ipv6 access-list acl4
switch(config-acl-acl4)#permit ipv6 any any flow-label 23 0x5678
switch(config-acl-acl4)#exit
switch(config)#
```

13.4.9.32 permit (IPv6 Prefix List)

The **permit** command adds a rule to the configuration mode IPv6 prefix list. Route map match commands use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains. Routes are redistributed into the specified domain when they match the prefix that a **permit** statement specifies.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode prefix list. The **no** seq (IPv6 Prefix Lists) command also removes the specified rule from the prefix list.

Command Mode

IPv6-pfx Configuration

Command Syntax

[**SEQUENCE**] **permit** ***ipv6_prefix*** [**MASK**]

Parameters

- **SEQUENCE** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding 10 to the number of the list's last rule.
  - **seq** ***seq_num*** Number is specified by ***seq_num***. Value ranges from ***0 to 65535***.
- ***ipv6_prefix*** IPv6 prefix upon which command filters routes (CIDR notation).
- **MASK** Range of the prefix to be matched.
  - ***no parameter*** Exact match with the subnet mask is required.
  - **eq** ***mask_e*** Prefix length is equal to ***mask_e***.
  - **ge** ***mask_g*** Range is from ***mask_g*** to ***128***.
  - **le** ***mask_l*** Range is from ***subnet*** mask length to ***mask_l***.
  - **ge** ***mask_l*** **le** ***mask_g*** Range is from ***mask_g*** to ***mask_l***.
  - ***mask_e***, ***mask_l*** and ***mask_g*** range from ***1 to 128***.
  - When **le** and **ge** are specified, the prefix list size ***mask_g mask_l***.

Example

This command appends a **permit** statement at the end of the text1 prefix list. The **permit** statement allows redistribution of routes with the specified prefix.

```
switch(config)# ipv6 prefix-list route-five
```

```
switch(config-ipv6-pfx)# permit 3100::/64
switch(config-ipv6-pfx)#
```

13.4.9.33 permit (MAC ACL)

The **permit** command adds a permit rule to the configuration mode MAC access control list packets through the interface to which the list is applied. Rule filters include protocol, source, and destination.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes the specified rule from the ACL.

Command Mode

MAC-ACL Configuration

Command Syntax

[**SEQ_NUM**] **permit SOURCE_ADDR DEST_ADDR** [**PROTOCOL**][log]

no permit SOURCE_ADDR DEST_ADDR [**PROTOCOL**][log]

default permit SOURCE_ADDR DEST_ADDR [**PROTOCOL**][log]

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding ***10*** to the number of the ACL's last rule.
  - ***1 - 4294967295*** Number assigned to entry.
- **SOURCE_ADDR** and **DEST_ADDR** Source and destination address filters. Options include:
  - ***mac_address mac_mask*** MAC address and mask.
  - **any** Packets from all addresses are filtered.
  - ***mac_address*** Specifies a MAC address in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh).
  - **mac_mask** Specifies a MAC address mask in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh).
  - ***0*** bits require an exact match to filter.
  - ***1*** bits filter on any value.
- **PROTOCOL** Protocol field filter. Values include:
  - **aarp** Appletalk Address Resolution Protocol (0x80f3).
  - **appletalk** Appletalk (0x809b).
  - **arp** Address Resolution Protocol (0x806).
  - **ip** Internet Protocol Version 4 (0x800).
  - **ipx** Internet Packet Exchange (0x8137).
  - **lldp** LLDP (0x88cc).
  - **novell** Novell (0x8138).
  - **rarp** Reverse Address Resolution Protocol (0x8035).
  - ***protocol_num*** Integer corresponding to a MAC protocol. Values range from ***0 to 65535***.
- **log** Triggers an informational log message to the console about the matching packet.

Examples

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all aarp packets from ***10.1000.0000*** through ***10.1000.FFFF*** to any host.

```
switch(config)# mac access-list text1
switch(config-mac-acl-text1)# permit 10.1000.0000 0.0.FFFF any aarp
switch(config-mac-acl-text1)#
```

- This command inserts a **permit** statement with the sequence number ***25***. The **permit** statement passes all packets through the interface.

```
switch(config-mac-acl-text1)# 25 permit any any
switch(config-mac-acl-text1)#
```

13.4.9.34 permit (Standard IPv4 ACL)

The **permit** command adds a permit rule to the configuration mode standard IPv4 Access Control List (ACL). Standard ACL rules filter on the source field.

Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding ***10*** to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes the specified rule from the ACL.

Command Mode

Std-ACL Configuration

Command Syntax

[**SEQ_NUM**] **permit SOURCE_ADDR** [log]

no permit SOURCE_ADDR [log]

default permit SOURCE_ADDR [log]

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding ***10*** to the number of the ACL's last rule.
  - ***1 - 4294967295*** Number assigned to entry.
- **SOURCE_ADDR** Source address filter. Options include:
  - ***network_addr*** Subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host *ip_addr*** IP address (dotted decimal notation).

    Subnet addresses support discontiguous masks.
- **log** Triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Example

This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all packets with a source address of ***10.10.1.1/24***.

```
switch(config)# ip access-list standard text1
switch(config-std-acl-text1)# permit 10.1.1.1/24
switch(config-std-acl-text1)#
```

13.4.9.35 permit (Standard IPv6 ACL)

The **permit** command adds a permit rule to the configuration mode standard IPv6 access control list. Standard ACL rules filter on the source field.

Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The no <sequence number> (ACLs) command also removes the specified rule from the ACL.

Command Mode

Std-IPv6-ACL Configuration

Command Syntax

[**SEQ_NUM**] **permit SOURCE_ADDR**

no permit SOURCE_ADDR

default permit SOURCE_ADDR

Parameters

- **SEQ_NUM** Sequence number assigned to the rule. Options include:
  - ***no parameter*** Number is derived by adding ***10*** to the number of the ACL's last rule.
  - ***1 - 4294967295*** Number assigned to entry.
- **SOURCE_ADDR** Source address filter. Options include:
  - ***ipv6_prefix*** IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host *ipv6_addr*** IPv6 host address.

Example

This command appends a **permit** statement at the end of the ACL. The **permit** statement drops packets with a source address of ***2103::/64***.

```
switch(config)# ipv6 access-list standard text1
switch(config-std-acl-ipv6-text1)# permit 2103::/64
switch(config-std-acl-ipv6-text1)#
```

13.4.9.36 remark

The **remark** command adds a non-executable comment statement into the pending ACL. Remarks entered without a sequence number are appended to the end of the list. Remarks entered with a sequence number are inserted into the list as specified by the sequence number.

The **default remark** command removes the comment statement from the ACL.

The **no remark** command removes the comment statement from the ACL. The command can specify the remark by content or by sequence number.

Command Mode

ACL Configuration

IPv6-ACL Configuration

Std-ACL Configuration

Std-IPv6-ACL Configuration

MAC-ACL Configuration

Command Syntax

remark *text*

line_num **`remark`** [***text***]

`no remark` ***text***

`default remark` ***text***

Parameters

- ***text*** The comment text.
- ***line_num*** Sequence number assigned to the remark statement. Value ranges from ***1 - 4294967295***.

Example

This command appends a comment to the list.

```
switch(config-acl-test1)# remark end of list
switch(config-acl-test1)# show
IP Access List test1
  10 permit ip 10.10.10.0/24 any
  20 permit ip any host 10.20.10.1
  30 deny ip host 10.10.10.1 host 10.20.10.1
  40 permit ip any any
  50 remark end of list
```

13.4.9.37 resequence (ACLs)

The **`resequence`** command assigns sequence numbers to rules in the configuration mode ACL. Command parameters specify the number of the first rule and the numeric interval between consecutive rules.

Maximum rule sequence number is ***4294967295***.

Command Mode

ACL Configuration

IPv6-ACL Configuration

Std-ACL Configuration

Std-IPv6-ACL Configuration

MAC-ACL Configuration

Command Syntax

`resequence` [***start_num*** [***inc_num***]]

Parameters

- ***start_num*** Sequence number assigned to the first rule. Default is ***10***.
- ***inc_num*** Numeric interval between consecutive rules. Default is ***10***.

Example

The **`resequence`** command re-numbers the list, starting the first command at number ***100*** and incrementing subsequent lines by ***20***.

- ```
  switch(config-acl-test1)# show
  IP Access List test1
    10 permit ip 10.10.10.0/24 any
    20 permit ip any host 10.20.10.1
    30 deny ip host 10.10.10.1 host 10.20.10.1
    40 permit ip any any
    50 remark end of list
  switch(config-acl-test1)# resequence 100 20
  switch(config-acl-test1)# show
  IP Access List test1
  ```
  ```

```
100 permit ip 10.10.10.0/24 any
120 permit ip any host 10.20.10.1
140 deny ip host 10.10.10.1 host 10.20.10.1
160 permit ip any any
180 remark end of list
```

13.4.9.38 route-map

The **`route-map`** command places the switch in ***route map*** configuration mode, which is a group change mode that modifies a route map statement. The command specifies the name and number of the route map statement that subsequent commands modify and creates a route map statement if it references a nonexistent statement. All changes in a group change mode edit session are pending until the end of the session.

Route maps define commands for redistributing routes between routing protocols. A route map statement is identified by a name, filter type (**permit** or **deny**), and sequence number. Statements with the same name are components of a single route map; the sequence number determines the order in which the statements are compared to a route.

The **`exit`** command saves pending route map statement changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **`abort`** command discards pending changes, returning the switch to global configuration mode.

The **`no route-map`** and **`default route-map`** commands delete the specified route map statement from ***running-config***.

Note: The route map configuration supports only standard ACL.

Command Mode

Global Configuration

Command Syntax

`route-map` ***map_name*** [**FILTER_TYPE**] [***sequence_number***]

`no route-map` ***map_name*** [**FILTER_TYPE**] [***sequence_number***]

`default route-map` ***map_name*** [**FILTER_TYPE**][***sequence_number***]

Parameters

- ***map_name*** Label assigned to route map. Protocols reference this label to access the route map.
- **FILTER_TYPE** Disposition of routes matching commands specified by route map statement.
  - **permit** Routes are redistributed when they match route map statement.
  - **deny** Routes are not redistributed when they match route map statement.
  - ***no parameter*** Sssigns **`permit`** as the **FILTER_TYPE**.

When a route does not match the route map criteria, the next statement within the route map is evaluated to determine the redistribution action for the route.

- ***sequence_number*** The route map position relative to other statements with the same name.
  - ***no parameter*** Sequence number of 10 (default) is assigned to the route map.
  - ***1-16777215*** Specifies sequence number assigned to route map.

Commands Available in route map configuration mode:

- continue (route map)
- match (route-map)
- set (route-map)

Examples

- This command creates the route map named ***map1*** and places the switch in route map configuration mode. The route map is configured as a permit map.

```
switch(config)# route-map map1 permit 20
switch(config-route-map-map1)#
```

- This command saves changes to ***map1*** route map, then returns the switch to global configuration mode.

```
switch(config-route-map-map1)# exit
switch(config)#
```

- This command saves changes to ***map1*** route map, then places the switch in interface-Ethernet mode.

```
switch(config-route-map-map1)# interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to ***map1*** route map, then returns the switch to global configuration mode.

```
switch(config-route-map-map1)# abort
switch(config)#
```

13.4.9.39 seq (IPv6 Prefix Lists)

The **no seq** command removes the rule with the specified sequence number from the ACL. The **default seq** command also removes the specified rule.

The **seq** keyword is a command option used at the beginning of deny (IPv6 Prefix List) and permit (IPv6 Prefix List) commands that places a new rule between two existing rules.

Command Mode

IPv6-pfx Configuration

Command Syntax

no seq ***line_num***

default seq ***line_num***

Parameters

line_num Sequence number of rule to be deleted. Valid rule numbers range from ***0*** to ***65535***.

Example

These commands remove rule ***20*** from the ***map1*** prefix list, then displays the resultant list.

```
switch(config)# ipv6 prefix-list map1
switch(config-ipv6-pfx)# no seq 20
switch(config-ipv6-pfx)# exit
switch(config)# show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64
switch(config)#
```

13.4.9.40 set (route-map)

The **set** command specifies modifications to routes that are selected for redistribution by the configuration mode route map.

The **no set** and **default set** commands remove the specified **set** command from the configuration mode route map statement by deleting the corresponding **set** command from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

set CONDITION

no set CONDITION

default set CONDITION

Parameters

- **CONDITION** Specifies the route modification parameter and value. Options include:
  - **as-path prepend** BGP AS number that is prepended to as-path. For details, see the set as-path prepend command.
    - ***1 - 4294967295*** BGP AS number to prepend.
    - **auto** Use peer AS number for inbound and local AS for outbound to prepend.
  - **distance *1 - 255*** Protocol independent administrative distance.
  - **ip next-hop *ipv4_address*** Next hop IPv4 address.
  - **ip next-hop peer-address** Use BGP peering address as next hop IPv4 address.
  - **ipv6 next-hop *ipv6_address*** Next hop IPv6 address.
  - **ipv6 next-hop peer-address** Use BGP peering address as next hop IPv6 address.
  - **local-preference *1 - 4294967295*** BGP local preference metric.
  - **metric *1 - 4294967295*** Route metric.
  - **metric + *1 - 4294967295*** Add specified value to current route metric.
  - **metric - *1 - 4294967295*** Subtract specified value to current route metric.
  - **metric-type OSPF_TYPE** OSPF metric type. Options include:
    - **type-1** OSPF type 1 metric.
    - **type-2** OSPF type 2 metric.
  - **origin O_TYPE** BGP origin attribute. Options for **O_TYPE** include:
    - **egp** Exterior BGP route.
    - **igp** Interior BGP route.
    - **incomplete** BGP route of unknown origin.
  - **tag *1 - 4294967295*** Route tag.
  - **weight *1 - 65535*** BGP weight parameter.

Related Commands

- route-map enters ***route-map*** configuration mode.
- set (route-map) specifies community modifications for the redistributed routes.
- set community (route-map) specifies extended community modifications for the redistributed routes.

Example

This command creates a route map entry that sets the local preference metric to ***100*** on redistributed routes.

```
switch(config)# route-map map1
switch(config-route-map-map1)# set local-preference 100
switch(config-route-map-map1)#
```

13.4.9.41 set as-path match

The **set as-path match** command configures the **AS_PATH** attribute for prefixes that are either received from a BGP neighbor or advertised to a BGP neighbor in the route map configuration mode.

The **no set as-path match** command removes the AS path specified for the BGP prefix.

Command Mode

Route-Map Configuration

Command Syntax

set as-path match all replacement [[none | auto] ***as_path***]

no set as-path match all replacement [[none | auto] ***as_path***]

Parameters

- **none** Replaces the ***AS-Path*** of the matching routes with a null or an empty ***AS-Path***.
- **auto** iF the specific route map is applied as an inbound policy to a corresponding BGP neighbor statement, then replace the **AS_PATH** of the prefixes received from this neighbor with the neighbor's AS number. If this route map is applied as an outbound policy to a corresponding neighbor statement, then replace the **AS_PATH** of the prefixes advertised to this neighbor with the locally configured AS number.
- ***as_path*** Replaces the AS-Path of the matching routes with an arbitrary ***AS-Path***.

Examples

- This command replaces the AS-Path with the **none** option.

```
switch# show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not
 installed, E
- ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q -
 Queued
for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop
 -
Link Local Nexthop

Network Next Hop Metric LocPref Weight Path
* > 101.101.1.0/24 80.80.1.1 - - - 200 i
* > 102.102.1.0/24 80.80.1.1 - - - 200 i
* > 103.103.1.0/24 80.80.1.1 - - - 200 302 i
* > 202.202.1.0/24 80.80.1.1 - - - 200 i

switch# configure terminal
switch(config)# route-map foo permit 10
switch(config-route-map-foo)# set as-path match all replacement none
switch(config-route-map-foo)# exit
switch(config)# router bgp 200
switch(config-router-bgp)# neighbor 80.80.1.2 route-map foo out
switch(config-router-bgp)# end

switch# show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not
 installed, E
- ECMP head, e - ECMP
```

```
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q -
 Queued
for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop
 -
Link Local Nexthop

Network Next Hop Metric LocPref Weight Path
* > 101.101.1.0/24 80.80.1.1 - - - 200 i
* > 102.102.1.0/24 80.80.1.1 - - - 200 i
* > 103.103.1.0/24 80.80.1.1 - - - 200 i
* > 202.202.1.0/24 80.80.1.1 - - - 200 i
```

- The AS-Path of matching prefixes are replaced with an empty or a null AS-Path. AS ***302*** is removed from prefix ***103.103.1.0/24*** as shown in the above output.
  - This command replaces the AS-Path with the **auto** option.

```
switch(config)# route-map foo permit 10
switch(config-route-map-foo)# set as-path match all replacement auto
switch(config-route-map-foo)# end

switch# show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not
 installed, E
- ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q -
 Queued
for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL
 Nexthop -
Link Local Nexthop

Network Next Hop Metric LocPref Weight Path
* > 101.101.1.0/24 80.80.1.1 - - - 200 200 i
* > 102.102.1.0/24 80.80.1.1 - - - 200 200 i
* > 103.103.1.0/24 80.80.1.1 - - - 200 200 i
* > 202.202.1.0/24 80.80.1.1 - - - 200 200 i
```

    The AS-Path of matching prefixes are replaced with the locally configured AS ***200***.
  - This command replaces the AS-Path with another AS-Path.

```
switch(config)# route-map foo permit 10
switch(config-route-map-foo)# set as-path match all replacement 500 600
switch(config-route-map-foo)# end

switch# show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not
 installed, E
- ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q -
 Queued
for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL
 Nexthop -
Link Local Nexthop
```

```
Network Next Hop Metric LocPref Weight Path
* > 101.101.1.0/24 80.80.1.1 - - - 200 500 600 i
* > 102.102.1.0/24 80.80.1.1 - - - 200 500 600 i
* > 103.103.1.0/24 80.80.1.1 - - - 200 500 600 i
* > 202.202.1.0/24 80.80.1.1 - - - 200 500 600 i
```

The AS-Path of matching prefixes are replaced with ***500 600*** as configured.

- This command replaces the AS-Path with a combination of **auto** and an AS-Path.

```
switch(config)# route-map foo permit 10
switch(config-route-map-foo)# set as-path match all replacement auto 500 600
switch(config-route-map-foo)# end

switch# show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not
 installed, E
- ECMP head, e - ECMP
 S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q -
 Queued
for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop
 -
Link Local Nexthop

 Network Next Hop Metric LocPref Weight Path
 * > 101.101.1.0/24 80.80.1.1 - - - 200 200 500 600 i
 * > 102.102.1.0/24 80.80.1.1 - - - 200 200 500 600 i
 * > 103.103.1.0/24 80.80.1.1 - - - 200 200 500 600 i
 * > 202.202.1.0/24 80.80.1.1 - - - 200 200 500 600 i
```

The AS-Path of matching prefixes are replaced with the locally configured AS ***200*** and ***500 600***.

13.4.9.42 set as-path prepend

The **`set as-path prepend`** command adds a **`set`** statement to a route map to prepend one or more Autonomous System (AS) numbers to the **AS_PATH** attribute of a BGP route.

The **`no set as-path prepend`** and **`default set as-path prepend`** commands remove the specified set statements from the route map and update all corresponding routes.

Command Mode

Route-Map Configuration

Command Syntax

`set as-path prepend` {{auto | ***as_number...*** [auto | ***as_number***]} | last-as ***count***}

`no set as-path prepend` {{auto | ***as_number...*** [auto | ***as_number***]} | last-as ***count***}

`default set as-path prepend` {{auto | ***as_number...*** [auto | ***as_number***]} | last-as ***count***}

Parameters

- **auto** Prepends the peer AS number for peer inbound route maps and the local AS number for peer outbound route maps.
- ***as_number*** Prepends the specified AS number. Can be entered in plain notation (values range from ***1-4294967295***) or in asdot notation as described in RFC 5396. In asdot notation, AS numbers from ***1-65535*** are entered in plain notation, and AS numbers from ***65536 to 4294967295*** are entered as two values separated by a dot. The first value is high-order and represents a multiple of ***65536***; the second

value is low-order and represents a decimal integer. For example, AS number ***65552*** can be entered as either ***65552*** or 1.16 (i.e., 1*65536+16). However they are entered, AS numbers are stored internally in plain decimal notation and will appear that way in **`show`** outputs.

- **last-as *count*** Prepends the last AS number in the AS path *count* times. Values range from ***1 to 15***. This is mutually exclusive with the use of the **auto** keyword or the entry of one or more specified AS numbers, and is not supported in multi-agent mode.

Examples

- These commands create a route-map entry that prepends AS number ***64496*** and prepends either the peer or local AS number twice.

```
switch(config)# route-map map1
switch(config-route-map-map1)# set as-path prepend 64496 auto auto
switch(config-route-map-map1)# exit

switch(config)# show route-map map1
route-map map1 permit 10
  Description:
  Match clauses:
  SubRouteMap:
  Set clauses:
    set as-path prepend 64496 auto auto
switch(config)#
```

- The commands create a route-map entry that prepends AS numbers ***64496***, ***64498***, and ***65552***.

```
switch(config)# route-map map2
switch(config-route-map-map2)# set as-path prepend 64496 64498 1.16
switch(config-route-map-map2)# exit

switch(config)# show route-map map2
route-map map2 permit 10
  Description:
  Match clauses:
  SubRouteMap:
  Set clauses:
    set as-path prepend 64496 64498 65552
switch(config)#
```

- These commands create a route map entry that prepends the last AS number ***12*** times.

```
switch(config)# route-map map3
switch(config-route-map-map3)# set as-path prepend last-as 12
switch(config-route-map-map3)# exit

switch(config)# show route-map map3
route-map map3 permit 10
  Description:
  Match clauses:
  SubRouteMap:
  Set clauses:
    set as-path prepend last-as 12
switch(config)#
```

13.4.9.43 set community (route-map)

The **`set community`** command specifies community attribute modifications to routes that are selected for redistribution by the configuration mode route map. The **`set community none`** command removes community attributes from the route.

The **no set community** and **default set community** commands remove the specified community from the configuration mode route map statement by deleting the corresponding statement from the ***running config***.

Command Mode

Route-Map Configuration

Command Syntax

set community [**GSHUT** | ***aa:nn*** | ***community-list*** | internet | local-as | no-advertise | no-export | none | ***number***]

no set community [**GSHUT** | ***aa:nn*** | additive | ***community-list*** | delete | internet | local-as | no-advertise | no-export | none | ***number***]

default set community [**GSHUT** | ***aa:nn*** | additive | ***community-list*** | delete | internet | local-as | no-advertise | no-export | none | ***number***]

Parameters

- **GSHUT** Configures a graceful shutdown in BGP.
- ***aa:nn*** Configures the community AS and network number, separated by colon. Value ranges from ***0:0 to 65535:65535***.
- ***community-list*** A label for community list.
- **internet** Advertises route to the Internet community.
- **local-as** Advertises route only to local peers.
- **no-advertise** Does not advertise route to any peer.
- **no-export** Advertises route only within BGP AS boundary.
- **none** Does not provide any community attributes.
- ***number*** Configures the community number. Value ranges from ***1*** to ***4294967040***.
- **additive** Adds specified attributes to the current community.
- **delete** Removes specified attributes from the current community.

Related Commands

- ip community-list
- route-map
- set (route-map)
- set community (route-map)

Guideline

EOS does not support disabling the process of graceful shutdown community.

Example

This command advertises routes only to local peers.

```
switch(config-route-map-map1)# show active
route-map map1 permit 10
   match community instances <= 50
   set community 0:456 0:2345
switch(config-route-map-map1)# set community local-as
switch(config-route-map-map1)# ip community-list 345 permit 23
switch(config)# route-map map1
switch(config-route-map-map1)# show active
route-map map1 permit 10
   match community instances <= 50
   set community 0:456 0:2345 local-as
switch(config-route-map-map1)#
```

13.4.9.44 set extcommunity (route-map)

The **set extcommunity** command specifies extended community attribute modifications to routes that are selected for redistribution by the configuration mode route map. The **set extcommunity none** command removes extended community attributes from the route.

The **no set extcommunity** and **default set extcommunity** commands remove the specified **set extcommunity** command from the configuration mode route map statement by deleting the corresponding statement from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

set extcommunity **COND_X** [**COND_2**][**COND_N**][**MOD_TYPE**]

set extcommunity none

no set extcommunity**COND_X**[**COND_2**][**COND_N**][**MOD_TYPE**]

no set extcommunity none

default set extcommunity **COND_X** [**COND_2**][**COND_N**][**MOD_TYPE**]

default set extcommunity none

Parameters

- **COND_X** Specifies extended community route map modification. Command may contain multiple attributes. Options include:
  - **rt *ASN:nn*** Route target attribute (AS:network number).
  - **rt *IP-address:nn*** Route target attribute (IP address: network number).
  - **soo *ASN:nn*** Site of origin attribute (AS:network number).
  - **soo *IP-address:nn*** Site of origin attribute (IP address: network number).
- **MOD_TYPE** Specifies route map modification method. Options include:
  - ***no parameter*** Command replaces existing route map with specified parameters.
  - **additive** Command adds specified parameters to existing route map.
  - **delete** Command removes specified parameters from existing route map.

Related Commands

- route-map enters route map configuration mode.
- set (route-map) specifies attribute modifications for the redistributed routes.

Example

This command creates a route map entry in ***map1*** that sets the route target extended community attribute.

```
switch(config)# route-map map1
switch(config-route-map-map1)# set extcommunity rt 10.13.2.4:100
switch(config-route-map-map1)#
```

13.4.9.45 show (ACL configuration modes)

The **show** command displays the contents of an Access Control List (ACL).

- **show** or **show pending** displays the list as modified in ACL configuration mode.
- **show active** displays the list as stored in running-config.
- **show comment** displays the comment stored with the list.
- **show diff** displays the modified and stored lists, with flags denoting the modified rules.

Exiting the ACL configuration mode stores all pending ACL changes to ***running-config***.

Command Mode

ACL Configuration

IPv6-ACL Configuration

Std-ACL Configuration

Std-IPv6-ACL Configuration

MAC-ACL Configuration

Command Syntax

show

show active

show comment

show diff

show pending

Examples

The examples in this section assume these ACL commands are entered as specified.

These commands are stored in **none**:

```
10 permit ip 10.10.10.0/24 any
20 permit ip any host 10.21.10.1
30 deny ip host 10.10.10.1 host 10.20.10.1
40 permit ip any any
50 remark end of list
```

The current edit session removed this command. This change is not yet stored to **none**:

```
20 permit ip any host 10.21.10.1
```

The current edit session added these commands ACL. They are not yet stored to **none**:

```
20 permit ip 10.10.0.0/16 any
25 permit tcp 10.10.20.0/24 any
45 deny pim 239.24.124.0/24 10.5.8.4/30
```

- This command displays the ACL, as stored in the configuration.

```
switch(config-acl-test_1)# show active
IP Access List test_1
  10 permit ip 10.10.10.0/24 any
  20 permit ip any host 10.21.10.1
  30 deny ip host 10.10.10.1 host 10.20.10.1
  40 permit ip any any
  50 remark end of list
```

- This command displays the pending ACL, as modified in ACL configuration mode.

```
switch(config-acl-test_1)# show pending
IP Access List test_1
  10 permit ip 10.10.10.0/24 any
  20 permit ip 10.10.0.0/16 any
  25 permit tcp 10.10.20.0/24 any
  30 deny ip host 10.10.10.1 host 10.20.10.1
  40 permit ip any any
```

```
   45 deny pim 239.24.124.0/24 10.5.8.4/30
   50 remark end of list
```

- This command displays the difference between the saved and modified ACLs.
  - Rules added to the pending list are denoted with a plus sign (+).
  - Rules removed from the saved list are denoted with a minus sign (-)

```
switch(config-acl-test_1)# show diff
---
+++
@@ -1,7 +1,9 @@
 IP Access List test_1
  10 permit ip 10.10.10.0/24 any
  20 permit ip any host 10.21.10.1
  20 permit ip 10.10.0.0/16 any
  25 permit tcp 10.10.20.0/24 any
  30 deny ip host 10.10.10.1 host 10.20.10.1
  40 permit ip any any
  45 deny pim 239.24.124.0/24 10.5.8.4/30
```

13.4.9.46 show hardware tcam profile

The **show hardware tcam profile** command displays the hardware specific information for the current operational TCAM profile in the running configuration.

This command is applicable to DCS-7280(E/R) and DCS-7500(E/R) series switches only.

Command Mode

EXEC

Command Syntax

show hardware tcam profile

Parameters

- ***tcam*** Specifies the TCAM information.
- ***profile*** Specifies the TCAM profile information.

Example

This command displays the current operational TCAM profile details.

```
switch# show hardware tcam profile
Configuration        Status
FixedSystem          default              default
```

13.4.9.47 show ip access-lists

The **show ip access-list** command displays the contents of IPv4 and standard IPv4 Access Control List (ACLs) on the switch. Use the **summary** option to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show ip access-list [**LIST**][**SCOPE**]

Parameters

- **LIST** Name of lists to be displayed. Selection options include:
  - ***no parameter*** All IPv4 ACLs are displayed.
  - ***list_name*** Specified IPv4 ACL is displayed.
- **SCOPE** Information displayed. Selection options include:
  - ***no parameter*** All rules in the specified lists are displayed.
  - **summary** The number of rules in the specified lists are displayed.

Examples

- This command displays all rules in ***test1*** IPv4 ACL.

```
switch# show ip access-list list2
IP Access List list2
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1 switch#
```

- This command displays the name of, and number of rules in, each list on the switch.

```
switch# show ip access-list summary
IPV4 ACL default-control-plane-acl
        Total rules configured: 12
        Configured on: control-plane
        Active on    : control-plane

IPV4 ACL list2
        Total rules configured: 3

IPV4 ACL test1
        Total rules configured: 6

Standard IPV4 ACL test_1
        Total rules configured: 1

IPV4 ACL test_3
        Total rules configured: 0 switch#
```

- This command displays the summary and lists all the configured IPv4 ACLs.

```
switch # show ip access-lists summary
IPV4 ACL default-control-plane-acl [readonly]
 Total rules configured: 17
 Configured on Ingress: control-plane(default VRF)
 Active on Ingress: control-plane(default VRF)

IPV4 ACL ipAclLimitTest
 Total rules configured: 0
 Configured on Egress: Vl2148,2700
 Active on Egress: Vl2148,2700
```

13.4.9.48 show ip prefix-list

The **`show ip prefix-list`** command displays all rules for the specified IPv4 prefix list. The command displays all IPv4 prefix list rules if a prefix list name is not specified.

Command Mode

EXEC

Command Syntax

`show ip prefix-list` [**DISPLAY_ITEMS**]

Parameters

DISPLAY_ITEMS Specifies the name of prefix lists for which rules are displayed. Options include:

- ***no parameter*** All IPv4 prefix list rules are displayed.
- ***list_name*** Specifies the IPv4 prefix list for which rules are displayed.

Example

This command displays all rules in the route-one IPv4 prefix list.

```
switch(config-ip-pfx)# show ip prefix-list
ip prefix-list route-one
    seq 10 deny 10.1.1.0/24
    seq 20 deny 10.1.0.0/16
    seq 30 permit 12.15.4.9/32
    seq 40 deny 1.1.1.0/24
switch(config-ip-pfx)#
```

13.4.9.49 show ipv6 access-lists

The **show ipv6 access-list** command displays the contents of all IPv6 Access Control List (ACLs) on the switch. Use the **summary** option to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show ipv6 access-list [**LIST**][**SCOPE**]

Parameters

- **LIST** Name of lists to be displayed. Selection options include:
  - ***no parameter*** All IPv6 ACLs are displayed.
  - ***list_name*** Specified IPv6 ACL is displayed.
- **SCOPE** Information displayed. Selection options include:
  - ***no parameter*** All rules in the specified lists are displayed.
  - **summary** The number of rules in the specified lists are displayed.

Examples

- This command displays all rules in test1 IPv6 ACL.

```
switch# show ipv6 access-list list2
IP Access List list2
        10 permit ipv6 3891:3c58:6300::/64 any
        20 permit ipv6 any host 2fe1:b468:024a::
        30 deny ipv6 host 3411:91c1:: host 4210:cc23:d2de:::
switch#
```

- This command displays the name of, and number of rules in, each list on the switch.

```
switch# show ipv6 access-list summary
IPV6 ACL list2
        Total rules configured: 3

IPV6 ACL test1
        Total rules configured: 6

IPV6 ACL test_1
        Total rules configured: 1
```

```
Standard IPV6 ACL test_3
        Total rules configured: 0 switch#
```

13.4.9.50 show ipv6 prefix-list

The **show ipv6 prefix-list** command displays all rules for the specified IPv6 prefix list. The command displays all IPv6 prefix lists if a prefix list name is not specified.

Command Mode

EXEC

Command Syntax

show ipv6 prefix-list [**DISPLAY_ITEMS**]

Parameters

DISPLAY_ITEMS Specifies the name of prefix lists for which rules are displayed. Options include:

- ***no parameter*** All IPv6 prefix lists are displayed.
- ***list_name*** Specifies the IPv6 prefix list for which rules are displayed.

Examples

- This command displays all rules in the map1 IPv6 prefix list:

```
switch> show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64 switch>
```

- This command displays all prefix lists:

```
switch> show ipv6 prefix-list
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64
ipv6 prefix-list FREDD
ipv6 prefix-list route-five
ipv6 prefix-list map2
seq 10 deny 10:1:1:1::/64 ge 72 le 80
seq 20 deny 10:1::/32 switch>
```

13.4.9.51 show mac access-lists

The **show mac access-list** command displays the contents of all MAC Access Control List (ACLs) on the switch. Use the summary to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show mac access-lists [**LIST**][**SCOPE**]

Parameters

- **LIST** Name of lists to be displayed. Selection options include:
  - ***no parameter*** Command displays all ACLs.
  - ***list_name*** Command displays ACL specified by parameter.
- **SCOPE** Information displayed. Selection options include:
  - ***no parameter*** Command displays all rules in specified lists.
  - **summary** Command displays the number of rules in specified lists.

Examples

- This command displays all rules in ***mtest2*** MAC ACL.

```
switch# show mac access-list mlist2
IP Access List mlist2
        10 permit 1024.4510.F125 0.0.0 any aarp
        20 permit any 4100.4500.0000 0.FF.FFFF novell
        30 deny any any switch#
```

- This command displays the number of rules in each MAC ACL on the switch.

```
switch# show mac access-list summary
MAC ACL mlist1
        Total rules configured: 6

MAC ACL mlist2
        Total rules configured: 3

MAC ACL mlist3
        Total rules configured: 1

MAC ACL mlist4
        Total rules configured: 0 switch#
```

13.4.9.52 show platform arad acl tcam summary

The **show platform arad tcam summary** command displays the percentage of TCAM utilization per forwarding ASIC.

Command Mode

EXEC

Command Syntax

show platform arad acl tcam ***summary***

Parameter

summary Displays the ACL TCAM summary.

Example

This command displays the percentage of TCAM utilization per forwarding ASIC.

```
switch# show platform arad acl tcam summary
The total number of TCAM lines per bank is 1024.

=========================================================
Arad3/0:
=========================================================
 Bank      Used           Used %                Used By
    1         4                0               IP RACLs
Total Number of TCAM lines used is: 4
```

```
=========================================================
Arad3/4:
=========================================================
 Bank      Used           Used %                 Used By
    1         2                0                IP RACLs
Total Number of TCAM lines used is: 2
```

13.4.9.53 show platform arad acl tcam

The **show platform arad acl tcam** command displays the number of TCAM entries (hardware resources) occupied by the ACL on each forwarding ASIC.

This command is applicable only on DCS-7500E, DCS-7280E series switches.

Command Mode

EXEC

Command Syntax

show platform arad acl tcam [*scope*]

Parameters

scope Specifies the information displayed. Options include:

- **detail** Displays the ACL TCAM details.
- **diff** Displays the difference between hardware and shadow.
- **hw** Displays the ACL entries from hardware.
- **shadow** Displays the ACL entries from shadow.
- **summary** Displays the ACL TCAM summary.

Examples

- This command displays the number of TCAM entries used by Arad0 ASIC. In this example, ACL is applied on two VLANs (***Vl2148*** and ***Vl2700***) but number of TCAM entries occupied is only one.

```
switch# show platform arad acl tcam detail
ip access-list ipAclLimitTest (Shared RACL, 0 rules, 1 entries, direction
 out,
state success, Acl Label 2)
Fap: Arad0, Shared: true, Interfaces: Vl2148, Vl2700
Bank Offset Entries
0         0       1
Fap: Arad1, Shared: true, Interfaces: Vl2148
Bank Offset Entries
0         0       1
```

- This command displays the percentage of TCAM utilization per forwarding ASIC.

```
switch# show platform arad acl tcam summary
The total number of TCAM lines per bank is 1024.
=========================================================
Arad0:
=========================================================
 Bank   Used                  Used %             Used By
    0      1                       0   IP Egress PACLs/RACLs
Total Number of TCAM lines used is: 1
=========================================================
Arad1:
=========================================================
 Bank   Used                   Used %             Used By
    0      1                        0   IP Egress PACLs/RACLs
```

```
Total Number of TCAM lines used is: 1
```

13.4.9.54 show platform arad mapping

The **show platform arad mapping** command displays the mapping between the interfaces and the forwarding ASICs.

Command Mode

EXEC

Command Syntax

show platform arad ***chip_name*** **mapping**

Parameter

chip_name Specifies the Arad chip name.

Example

This command displays the mapping between the interfaces and the forwarding ASICs on the Arad3/0 chip.

```
switch# show platform arad arad3/0 mapping
Arad3/0  Port                      SysPhyPort    Voq   ( Fap,FapPort)    Xlge     Serdes
-------------------------------------------------------------------------------
         Ethernet3/1/1                    34    288        (0  ,  2)     n/a        (20)
.................................................................................
```

13.4.9.55 show platform fap acl

The **show platform fap acl** command displays the ACL information of Sand platform devices.

Command Mode

Privileged EXEC

Command Syntax

show platform fap acl [ipkgv | l4ops | mirroring | opkgv | pmf | tcam | udf | vsicfg]

Parameters

- **ipkgv** Displays the ACL Ingress Interface Specification (IPKGV) information.
- **l4ops** Displays the ACL Layer 4 Options (L4OPS) information.
- **mirroring** Displays the mirroring ACL information.
- **opkgv** Displays the ACL Egress Interface Specification (OPKGV) information.
- **pmf** Displays the Pmf.
- **tcam** Displays the ACL TCAM information.
- **udf** Displays the ACL UDF information.
- **vsicfg** Displays the ACL Virtual Switch Instance (VSI) CONFIG information.

Guidelines

This command is supported on DCS-7280SE and DCS-7500E series platforms only.

Example

This command displays the brief information of all installed mirroring ACLs.

```
switch(config)# show platform fap acl mirroring

==============
 Aggregate ACLs
==============
```

```
  (list2:0->2) type=2; version=0
   - list2 [ prio 0 ] => session 2

  (list1:10->1,list3:20->3) type=0; version=13
   - list3 [ prio 20 ] => session 3
   - list1 [ prio 10 ] => session 1

======================
 Interface-ACL Mapping
======================

   Ethernet1 => (list1:10->1,list3:20->3) [ ipv4 ]
   Ethernet33 => (list2:0->2) [ mac ]
```

13.4.9.56 show platform fap acl tcam

The **show platform fap tcam** command displays the number of TCAM entries (hardware resources) occupied by the ACL on each forwarding ASIC of Sand platform devices.

Command Mode

Privileged EXEC

Command Syntax

show platform fap acl tcam [detail | diff | hw | shadow | summary]

Parameter

- **detail** Displays the number of TCAM entries (hardware resources) occupied by the ACL on each forwarding ASIC.
- **diff** Displays the difference between hardware and shadow.
- **hw** Displays ACL entries from hardware.
- **shadow** Displays ACL entries from shadow.
- **summary** Displays the percentage of TCAM utilization per forwarding ASIC.

Example

This command displays the number of TCAM entries and other ACL TCAM detail.

```
switch# show platform fap acl tcam detail
ip access-list ipAcl0000 (RACL, 1 rules, 2 entries, direction in, state
 success)
 Shared: false
 Interface: Vlan0002
 -------------------
 Fap: Arad3/0
 Bank Offset Entries
 1         0       2
 Interface: Vlan0003
 -------------------
 Fap: Arad3/0
 Bank Offset Entries
 1         2       2
 Fap: Arad3/4
 Bank Offset Entries
 1         0       2
```

13.4.9.57 show platform fap acl tcam hw

The **show platform fap acl tcam hw** command displays the TCAM entries configured for each TCAM bank including policy-maps and corresponding traffic match.

This command is applicable only on DCS-7280(E/R), DCS-7500(E/R) series switches.

Command Mode

EXEC

Command Syntax

show platform fap ***fap_name*** **acl tcam hw**

Parameters

- ***fap_name*** Specifies the switch chip-set name.
- **acl** Specifies the Arad ACL information.
- **tcam** Specifies the Arad TCAM information.
- **hw** Specifies the ACL entries for hardware.

Example

This command displays the TCAM entries configured for each TCAM bank including policy maps and corresponding traffic matches.

```
switch# show platform fap Arad1 acl tcam hw
================================================================================
Arad1 Bank 0 Type: dbPdpIp, dbPdpIp6, dbPdpMpls, dbPdpNonIp, dbPdpTunnel
================================================================================
------------------------------------------------
|Offs|X|PR|TT|R|QI|V6MC|DPRT|SPRT|F|DEST |V|ACT  |H|
------------------------------------------------
|29  |4|59|  | |01|    |    |    | |     |3|0008f|0|
|    |4|59|  | |01|    |    |    | |     |0|00000|0|
|30  |4|33|  | |01|    |    |    | |     |3|0008f|0|
|    |4|33|  | |01|    |    |    | |     |0|00000|0|
|31  |4|32|  | |01|    |    |    | |     |3|0008f|0|
|    |4|32|  | |01|    |    |    | |     |0|00000|0|
|32  |4|  |  | |01|ff02|    |    | |     |3|00097|0|
|    |4|  |  | |01|ff02|    |    | |     |0|00000|0|
|33  |4|06|  | |01|    |    |00b3| |26ffd|3|0009b|0|
|    |4|06|  | |01|    |    |00b3| |26ffd|0|00000|0|
|34  |4|06|  | |01|    |00b3|    | |26ffd|3|0009b|0|
------------------------------------------------
|Offs|X|R|QI|DAHI|PT|DALO    |DEST |V|ACT  |H|
------------------------------------------------
--------------------------------------------------------------------------------
|Offs|X|TT0|QI|FOI|TT1|DEST |TT1P |PT|VX_DP|PN|F|MC|O|V|HDR OFFSETS |ACT  |H|
================================================================================
Arad1 Bank 1 Type: dbIpQos
================================================================================
--------------------------------------------------------------------------
|Offs|X|TC|CL|DPRT|SPRT|VQ|L4OPS |PP|PR|F|V4_DIP  |V4_SIP  |V|ACT  |H|
--------------------------------------------------------------------------
|0   |0|  |  |    |    |  |      |01|  | |        |        |3|00000|0|
|    |0|  |  |    |    |  |      |01|  | |        |        |0|00000|0|
--------------------------------------------------------------------------
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

13.4.9.58 show platform fap acl tcam summary

The **show platform fap acl tcam summary** command displays for each forwarding ASIC, the number of TCAM entries consumed per ACL type, and in which TCAM bank the entries are installed. A mirroring ACL does not consume TCAM resources unless attached to a mirroring source interface, and a mirroring destination is configured. If the mirroring destination is a GRE tunnel, at least one nexthop entry for the tunnel destination must be resolved before a TCAM entry is installed.

Command Mode

EXEC

Command Syntax

show platform fap acl tcam summary

Example

This command displays the number of TCAM entries consumed per ACL type, the bank installed, and ASIC. Three TCAM entries are consumed across two forwarding ASICs, two for IP ACLs, and one for MAC ACLs.

```
switch# show platform fap acl tcam summary
=======================================================
 Arad0:
=======================================================
    Bank   Used Used %    Used By
    0, 1      2      0    IP Mirroring
 Total Number of TCAM lines used is: 4
=======================================================
 Arad1:
=======================================================
   Bank   Used            Used %                 Used By
      2      1                 0           Mac Mirroring
```

13.4.9.59 show platform trident tcam

The **show platform trident tcam** command displays the TCAM entries configured for each TCAM group including policy maps and corresponding hits.

Command Mode

EXEC

Command Syntax

show platform trident tcam [acl | cpu-bound | **detail** | directed-broadcast | entry | mirror | pbr | pipe | qos | shared | summary]

Parameters

- ***no parameters*** Displays TCAM entries for each TCAM group.
- **acl** Displays the trident ACL information.
- **cpu-bound** Displays the trident cpu-bound information.
- **detail** Lists all TCAM entries.
- **directed-broadcast** Allows inbound broadcast IP packets with Source IP address as one of the permitted broadcast host.
- **entry** Displays the TCAM entry information.
- **mirror** Displays the trident Mirroring ACL information.
- **pbr** Displays the trident PBR ACL information.
- **pipe** Allows to specify a pipe for filtering.
- **qos** Displays the trident QOS information.
- **shared** Displays the ACL Sharing information.
- **summary** Displays the TCAM allocation information.

Guidelines

This command is applicable only on DCS-7010, DCS-7050/DCS-7050X, DCS7250X, DCS-7300X series switches.

Examples

- This command displays the Trident mirroring ACL information.

```
switch(config)# show platform trident tcam mirror
=== Mirroring ACLs on switch Linecard0/0 ===

Session: mir-sess2
```

```
INGRESS ACL mirAcl2* uses 2 entries
 Assigned to ports: Ethernet32/1
```

- This command displays the allowed IP Destination address from the in coming packets

```
switch# show platform trident tcam directed-broadcast
DirectedBroadcast Feature Tuples.
Src Ip          Dst Ip          Action          Hits
--------------- --------------- ------- ------------
10.1.1.1        192.164.2.15    Permit             0
20.1.1.1        192.164.2.15    Permit             0
30.1.1.1        192.164.2.15    Permit             0
10.1.1.1        192.166.2.15    Permit             0
20.1.1.1        192.166.2.15    Permit             0
30.1.1.1        192.166.2.15    Permit             0
10.1.1.1        192.168.2.255   Permit             0
20.1.1.1        192.168.2.255   Permit             0
30.1.1.1        192.168.2.255   Permit             0
*               192.164.2.15    Deny               0
*               192.166.2.15    Deny               0
*               192.168.2.255   Deny               0
```

.

- This command displays detailed information for the TCAM group.

```
switch# show platform trident tcam detail
=== TCAM detail for switch Linecard0/0 ===
TCAM group 9 uses 42 entries and can use up to 1238 more.
 Mlag control traffic uses 4 entries.
    589826               0 hits - MLAG - SrcPort UDP Entry
    589827               0 hits - MLAG - DstPort UDP Entry
    589828               0 hits - MLAG - SrcPort TCP Entry
    589829               0 hits - MLAG - DstPort TCP Entry
 CVX traffic reserves 6 entries (0 used).
 L3 Control Priority uses 23 entries.
    589836               0 hits - URM - SelfIp UDP Entry
    589837               0 hits - URM - SelfIp TCP Entry
589848              0 hits - OSPF - unicast
    589849           71196 hits - OSPFv2 - Multicast
    589850               0 hits - OSPFv3 - Multicast
    589851               0 hits - OSPF Auth ESP - Multicast
    589852               0 hits - OSPF Auth ESP - Unicast
    589853               0 hits - IP packets with GRE type and ISIS protocol
    589854               0 hits - RouterL3 Vlan Priority 6,7 Elevator
    589855               0 hits - RouterL3 DSCP 48-63 Elevator
    589856               0 hits - RouterL3 Priority Elevator
    589857               0 hits - NextHopToCpu, Glean
    589858               0 hits - L3MC Cpu OIF
 IGMP Snooping Flooding reserves 8 entries (6 used).
589864              0 hits - IGMP Snooping Restricted Flooding L3 from local
mlag peer
    589865               0 hits - IGMP Snooping Restricted Flooding L3
 L4 MicroBfd traffic reserves 1 entries (0 used).
TCAM group 13 uses 99 entries and can use up to 1181 more.
 Dot1x MAB traffic uses 1 entries.
    851968               0 hits - Dot1xMab Rule

<-------OUTPUT OMITTED FROM EXAMPLE-------->

ck338.22:14:38(config-pmap-qos-policy1)#
```

13.4.9.60 show route-map

The **`show route-map`** command displays the contents of configured route maps.

Command Mode

EXEC

Command Syntax

show route-map [***map_name***]

Parameters

- ***no parameter*** Displays the content of all configured route maps.
- ***map_name*** Displays the content of the specified route map.

Examples

- This command displays the ***map1*** route map.

```
switch(config)# show route-map map1
route-map map1 permit 10
  Description:
  Match clauses:
  SubRouteMap:
  Set clauses:
    set as-path prepend last-as 12
    set as-path prepend auto auto
```

- This command displays the ***map*** route map.

```
switch> show route-map map
route-map map permit 5
  Match clauses:
    match as 456
Set clauses:
route-map map permit 10
  Match clauses:
match ip next-hop 2.3.4.5
    match as-path path_2
  Set clauses:
    set local-preference 100
```

13.4.9.61 system profile

The **system profile** command creates a new Ternary Content-Addressable Memory (TCAM) profile in the running configuration.

The **default system profile** and **no system profile** commands delete non-default TCAM profiles from the running configuration.

Command Mode

Hardware TCAM

Command Syntax

system profile [***profile_name***| default | mirroring-acl | pbr-match-nexthop-group | qos | tap-aggregation-default | tap-aggregation-extended | tc-counters]

default system profile

no system profile

Parameters

- ***profile_name*** Creates a profile with the specified name.
- **default** Creates a default profile.
- **mirroring-acl** Creates a mirroring-ACL profile.
- **pbr-match-nexthop-group** Creates a pbr-match-nexthop-group profile.
- **qos** Creates a Quality of Service (QoS) profile.
- **tap-aggregation-default** Creates a tap-aggregation-default profile.
- **tap-aggregation-extended** Creates a tap-aggregation-extended profile.

- **tc-counters** Creates a tc-counters profile.

Guideline

These commands are compatible with the DCS-7280SE and DCS-7500E series switches only.

Examples

- These commands create a mirroring-ACL profile.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# system profile mirroring-acl
switch(config-hw-tcam)# show hardware tcam profile
                    Configuration        Status
FixedSystem         mirroring-acl        mirroring-acl
switch(config-hw-tcam)#
```

- These commands delete non-default TCAM profiles.

```
switch(config)# hardware tcam
switch(config-hw-tcam)#show hardware tcam profile
                    Configuration        Status
Linecard9           mirroring-acl        mirroring-acl
Linecard8           mirroring-acl        mirroring-acl
Linecard3           mirroring-acl        mirroring-acl
Linecard4           mirroring-acl        mirroring-acl
Linecard6           mirroring-acl        mirroring-acl
switch(config-hw-tcam)# default system profile
switch(config-hw-tcam)# show hardware tcam profile
                    Configuration        Status
Linecard9           default              default
Linecard8           default              default
Linecard3           default              default
Linecard4           default              default
Linecard6           default              default
switch(config-hw-tcam)#
```

- These commands delete TCAM profiles.

```
switch(config-hw-tcam)# show hardware tcam profile
                    Configuration        Status
Linecard9           tc-counters          tc-counters
Linecard8           tc-counters          tc-counters
Linecard3           tc-counters          tc-counters
Linecard4           tc-counters          tc-counters
Linecard6           tc-counters          tc-counters
switch(config-hw-tcam)# no system profile
switch(config-hw-tcam)# show hardware tcam profile
                    Configuration        Status
Linecard9           default              default
Linecard8           default              default
Linecard3           default              default
Linecard4           default              default
Linecard6           default              default
switch(config-hw-tcam)#
```

13.5 VRRP and VARP

A virtual IP (VIP) address is an IP address that does not directly connect to a specific interface. Inbound packets sent to a Virtual IP address are redirected to a physical network interface. VIPs support connection

redundancy by assigning the address to multiple switches. If one device becomes unavailable, packets sent to the address are still serviced by the functioning device.

Arista switches support virtual IP addresses through Virtual Router Redundancy Protocol, version 2 (VRRPv2), Virtual Router Redundancy Protocol, version 3 (VRRPv3), and Virtual-ARP (VARP). This chapter describes the Arista switch support of virtual IP addresses and contains these sections:

- VRRP and VARP Conceptual Overview
- VRRP and VARP Implementation Procedures
- VRRP and VARP Implementation Examples
- VRRP and VARP Configuration Commands

13.5.1 VRRP and VARP Conceptual Overview

This section review the following topics:

- VRRPv2
- VRRPv3
- VARP

13.5.1.1 VRRPv2

A virtual router, also known as a virtual router group, is defined by a Virtual Router Identifier (VRID) and a virtual IP address. A virtual routers mapping of VRID and IP address must be consistent among all switches implementing the virtual router group. A virtual routers scope is restricted to a single LAN.

A LAN may contain multiple virtual routers for distributing traffic. Each virtual router on a LAN is assigned a unique VRID. A switch may be configured with virtual routers among multiple LANs.

VRRP uses priority ratings to assign Master or Backup roles for each VRRP router configured for a virtual router group. The Master router sends periodic VRRP Advertisement messages along the LAN and forwards packets received by the virtual router to their destination. Backup routers are inactive but are available to assume Master router duties when the current Master fails.

A VRRP can be configured to allow VRRP routers with higher priority to take over Master router duties. Alternatively, the group can be configured to prevent a router from preemptively assuming the Master role. A VRRP router is always assigned the Master of any virtual router configured with the address owned by the VRRP router, regardless of the preemption prevention setting.

On 7280R3, 7500R3 and 7800R3 series, maximally 14 unique VRRP groups can be configured along with VARP and MLAG Peer gateway virtual MAC capabilities.

13.5.1.2 VRRPv3

RFC 5798 defines version 3 of the Virtual Router Redundancy Protocol (VRRP) for both IPv4 and IPv6. It is based on version 2 of VRRP, as defined in ***RFC 3768***.

13.5.1.3 VARP

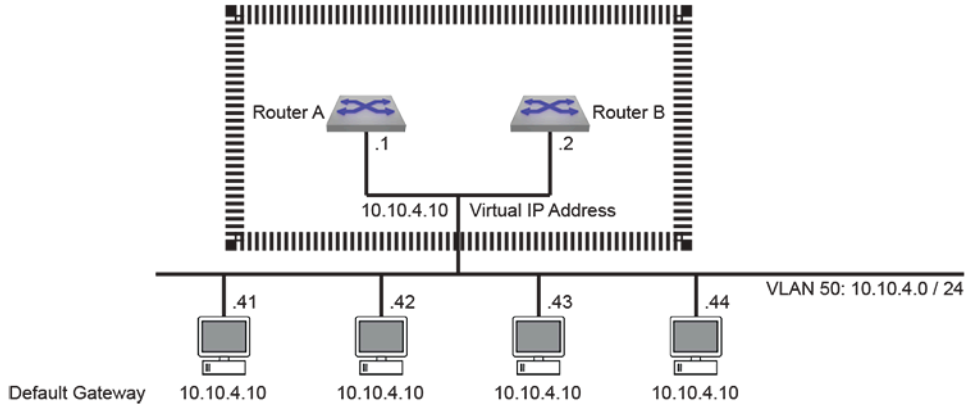
Virtual VRRP (VARP) allows multiple switches to simultaneously route packets from a common IP address in an active-active router configuration. Each switch is configured with the same set of virtual IP addresses on corresponding VLAN interfaces and a common virtual MAC address. In MLAG configurations, VARP is preferred over VRRP because VARP does not require traffic to traverse the peer-link to the master router as VRRP would.

A maximum of 500 virtual IP addresses can be assigned to a VLAN interface. All virtual addresses on all VLAN interfaces resolve to the same virtual MAC address.

VARP functions by having each switch respond to ARP and GARP requests for the configured router IP address with the virtual MAC address. The virtual MAC address is only for inbound packets and never used in the source field of outbound packets.

When ip routing is enabled, packets to the virtual MAC address are routed to the next hop destination.

Figure 13-2: VARP Configuration



The following sections are included in this chapter:

- VRRP and VARP Conceptual Overview
- VRRP and VARP Implementation Procedures
- VRRP and VARP Implementation Examples
- VRRP and VARP Configuration Commands

13.5.2 VRRP and VARP Implementation Procedures

This section contains the following configuration instructions:

- VRRP Configuration for IPv4
- VRRP Configuration for IPv6
- VARP Configuration

13.5.2.1 VRRP Configuration for IPv4

To implement a virtual router, it must be configured and enabled. A virtual router is typically configured before it is enabled; this ensures that the VRRP router operates as required before its priority settings immediately make it the master virtual router. Because assigning a primary address to a virtual router enables it, address assignment is normally performed after all other configuration tasks.

The **`no vrrp`** command removes all VRRP commands for the specified virtual router from ***running-config***.

13.5.2.1.1 Virtual Router Configuration

Most configuration tasks are optional because all mandatory parameters have a default value. The following virtual router parameters are configurable:

- VRRP version (default = version 2)
- Router priority (default = 100)
- Preemption option (default is enabled)
- Advertisement timer (default = one second)
- Description (optional parameter)
- Peer authentication (optional parameter)
- Secondary IP addresses (optional parameter)

VRRP Version

The **vrrp ipv4 version** command sets the version of VRRP for the corresponding IPv4 virtual router. IPv6 version is not configurable as it only supports version 3. The version selected in a VRRP group can either be same for all group members or independent of each other. By default, Arista switches use VRRP version 2, which supports only IPv4 environments. VRRP version 3 supports both IPv4 and IPv6 environments.

Example

This command causes ***vlan 20*** to use VRRP version 3.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 1 ipv4 version 3
switch(config-if-vl20)#
```

Master and Backup Router

The VRRP routers within a virtual router group determine the Master router through priority settings. Priority values range from ***254*** (highest priority) to ***1*** (lowest priority). Priority is either set by a CLI command or is assigned the default value of ***100***. A switch specifies priority settings for each of its virtual routers. Once set, VRRP priority level can also be changed by a tracked object. The **vrrp tracked-object** command configures the VRRP client process to track an object created by the **track** command and react if its status changes to ***down***.

Preemption mode determines when a VRRP router with a higher priority rating becomes the Master router. If preemption is enabled, the VRRP router with the highest priority immediately becomes the Master router. If preemption is disabled, a VRRP router with a higher priority value does not become the Master router unless the current Master becomes unavailable; this is applicable when a new VRRP router becomes available on the LAN or VRRP routers priority value changes for the virtual router.

The **vrrp priority-level** command configures the switchs priority setting for the specified virtual router.

Example

This command sets the priority value of ***250*** for the virtual router with ***VRID 15*** on ***VLAN 20***.

```
switch(config-if-vl20)# vrrp 15 priority-level 250
switch(config-if-vl20)#
```

The **vrrp preempt** command controls the preempt mode setting of the specified virtual router. By default, preempt mode is enabled.

Examples

- This command disables preempt mode for the virtual router ***15*** on ***vlan 20***.

```
switch(config-if-vl20)# no vrrp 15 preempt
switch(config-if-vl20)#
```

- This command enables preempt mode for the virtual router ***30*** on ***vlan 20***.

```
switch(config-if-vl20)#vrrp 30 preempt
switch(config-if-vl20)#
```

The **vrrp preempt delay** command configures a period between an event that elevates a switch to master VRRP router status and the switch's assumption of master VRRP router role. Command options configure delays during normal operation and after a switch reboot.

Advertisement Interval

The Master router sends periodic VRRP Advertisement messages to other VRRP routers. The **vrrp advertisement interval** command specifies the interval between successive advertisement message transmissions.

The advertisement interval also defines the timeout that determines when the switch assumes the Master router role. This timeout interval is three times the advertisement interval.

Example

This command sets the advertisement interval of ***10*** seconds for virtual router ***35*** on ***vlan 100***.

```
switch(config-if-vl100)# vrrp 35 advertisement interval 10
switch(config-if-vl100)#
```

Description

The **vrrp session description** command associates a text string to the specified virtual router. The maximum string length is ***80*** characters. The string has no functional impact on the virtual router.

Example

This command associates the text string ***Laboratory Router*** to virtual router ***15*** on ***vlan 20***.

```
switch(config-if-vl20)# vrrp 15 session description Laboratory Router
switch(config-if-vl20)#
```

Peer Authentication

VRRP peer authentication validates VRRP advertisement packets that the switch receives from other VRRP routers in a specified virtual router group. When a virtual router uses authentication, all VRRP routers in the group must use the same authentication parameters.

The **vrrp peer authentication** command configures virtual router authentication parameters for the specified virtual router.

Example

This command implements plain-text authentication, using ***12345*** as the key, for virtual router ***40*** on ***vlan 100***.

```
switch(config-if-vl100)#vrrp 40 peer authentication text 12345
switch(config-if-vl100)#
```

Secondary Addresses

The **vrrp ipv4 secondary** command assigns a secondary IP address to a virtual router. Secondary addresses are optional; a virtual routers configuration may include more than one secondary address command. The primary and secondary address list must be identical for all switches in a virtual router group.

A primary IP address is assigned to a virtual router with the **vrrp ipv4** command (Virtual Router Enabling and the Primary IP address).

Example

This command assigns the IP address of ***10.2.4.5*** as the secondary IP address for the virtual router ***15*** on ***vlan 20***.

```
switch(config-if-vl20)# vrrp 15 ipv4 10.2.4.5 secondary
switch(config-if-vl20)#
```

13.5.2.1.2 Virtual Router Enabling and the Primary IP address

The **vrrp ipv4** command configures the primary IP address of the specified virtual router and enables the virtual router if the primary address is contained within the configuration mode interfaces IP address subnet. A virtual routers configuration may contain only one primary IP address assignment command; subsequent **vrrp ipv4** commands reassign the virtual routers primary IP address.

Example

This command enables virtual router group ***15*** (VRID) on ***vlan 20*** and assigns ***10.1.1.5*** as the virtual routers primary address.

```
switch(config-if-vl20)# vrrp 15 ipv4 10.1.1.5
switch(config-if-vl20)#
```

13.5.2.1.3 Disabling VRRP

The **vrrp disabled** command places the switch in stopped state for the specified virtual router. While in stopped state, the switch cannot act as a Master or backup router for the virtual router group. The **no vrrp disabled** command changes the switchs virtual router state to **backup** or **master** if the virtual router is properly configured.

VRRP can also be shut down when the status of a tracked object configured by the **vrrp tracked-object** command changes to **down**.

Examples

- This command places the switch in stopped mode for virtual router ***24*** on ***vlan 20***.

```
switch(config-if-vl20)# vrrp 24 disabled
switch(config-if-vl20)#
```

- This command moves the switch out of stopped mode for virtual router ***24*** on ***vlan 20***.

```
switch(config-if-vl20)# no vrrp 24 disabled
switch(config-if-vl20)#
```

- This command configures the switch to enter stopped mode for virtual router ***24*** on ***vlan 20*** if the status of ***tracked-object interfaceE6/48*** changes to **down**.

```
switch(config-if-vl20)# vrrp 24 tracked-object interfaceE6/48 shutdown
switch(config-if-vl20)#
```

The **no vrrp** and **no vrrp ipv4** commands delete the specified virtual IP address from the interface. Additionally, the **no vrrp** command removes all residual VRRP commands for the virtual router.

Examples

- This command removes all VRRP configuration commands for virtual router ***10*** on ***vlan 15***.

```
switch(config-if-vl15)# no vrrp 10
switch(config-if-vl15)#
```

- This command disables virtual router ***25*** on ***vlan20*** and removes the primary IP address from its configuration.

```
switch(config-if-vl20)# no vrrp 25 ipv4 10.1.1.5
switch(config-if-vl20)#
```

13.5.2.2 VRRP Configuration for IPv6

To implement a virtual router, it must be configured and enabled. A virtual router is typically configured before it is enabled; this ensures that the VRRP router operates as required before its priority settings immediately make it the master virtual router. Because assigning a primary address to a virtual router enables it, address assignment is normally performed after all other configuration tasks.

The **no vrrp** command removes all VRRP commands for the specified virtual router from ***running-config***.

13.5.2.2.1 Configuring VRRP for IPv6

Specify the VRRP Version

The **vrrp ipv4 version** command sets the version of VRRP used on an interface. The version selected in a VRRP group must be the same for all group members. By default, Arista switches use VRRP version 2, which is not compatible with IPv6.

Example

This command causes ***vlan 20*** to use VRRP version 3.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 1 ipv4 version 3
switch(config-if-vl20)#
```

Create a VRRP Group and Configure a Virtual IPv6 Address

The **vrrp ipv6** command assigns an IPv6 address to the interface being configured and creates a VRRP group.

Example

These commands create VRRP group 3 and configure a virtual IPv6 address for the VRRP group on the ***vlan 20*** interface.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 3 ipv6 2001:db8:0:1::1
switch(config-if-vl20)#
```

Configure Tracking

The **vrrp tracked-object** command configures the VRRP client process to track an object created by the **track** command and react if its status changes to ***down***.

Example

This command causes interface ***vlan 20*** to disable VRRP when tracked object ETH8 changes state.

```
switch(config-if-vl20)# vrrp 1 tracked-object ETH8 shutdown
switch(config-if-vl20)#
```

Configure the Priority Level

The **vrrp priority-level** command configures the switchs priority setting for the specified virtual router.

Example

This command sets the priority value of ***250*** for the virtual router with ***VRID 15*** on ***vlan 20***.

```
switch(config-if-vl20)# vrrp 15 priority-level 250
```

```
switch(config-if-vl20)#
```

Configure the Preemption Mode

Preemption mode determines when a VRRP router with a higher priority rating becomes the Master router. If preemption is enabled, the VRRP router with the highest priority immediately becomes the Master router. If preemption is disabled, a VRRP router with a higher priority value does not become the Master router unless the current Master becomes unavailable; this is applicable when a new VRRP router becomes available on the LAN or VRRP routers priority value changes for the virtual router.

The **vrrp preempt** command controls the preempt mode setting of the specified virtual router. By default, preempt mode is enabled.

Example

This command enables preempt mode for the virtual router ***30*** on ***vlan 20***.

```
switch(config-if-vl20)# vrrp 30 preempt
```

Configure the VRRP Advertisement Interval

The **ip virtual-router mac-address advertisement-interval** command specifies the interval between advertisement packets sent by the master router to the VRRP group members.

Example

This command configures a MAC address advertisement interval of one minute (***60*** seconds).

```
switch(config)# interface vlan 20
switch(config-if-vl20)# ip virtual-router mac-address advertisement-interval 60
switch(config-if-vl20)#
```

13.5.2.2.2 Verify VRRP IPv6 Configurations

Use the following commands to display the VRRP configurations and status.

Show VRRP Group

The **show vrrp** command displays information about the Virtual Router Redundancy Protocol (VRRP) groups configured on a specified interface.

Example

This command displays a table of information for VRRP groups on the switch.

```
switch# show vrrp interface vlan 3060 brief
Interface Id  Ver Pri Time  State   VrIps
Vlan3060  1   3   100 3609  Master  2001::2
                                    2001::3
Vlan3060  2   3   100 3609  Master  2002::2
                                    2002::3 switch#
```

13.5.2.3 VARP Configuration

Implementing VARP consists of assigning virtual IP addresses to VLAN interfaces and configuring a virtual MAC address.

Virtual IP Addresses

The **ip virtual-router address** command assigns a virtual IP address to the VLAN interface being configured. Unlike VRRP, the virtual IP address does not have to be in the same subnet as the physical interface.

A virtual IPv4 address may optionally be configured with a subnet, but doing so will modify the behavior of ARP requests sent from the router. When the router sends an ARP request for an IPv4 address in a virtual subnet, the ARP request will use the virtual IPv4 address as the source IP address and the virtual MAC address as the source MAC address inside the ARP header. For virtual IP addresses configured without the subnet option, no modifications are made to outgoing ARP requests.

Examples

- These commands configure a Switch Virtual Interface (SVI) and a virtual IP address for ***VLAN 10***.

```
switch(config)# interface vlan 10
switch(config-if-Vl10)# ip address 10.0.0.2/24
switch(config-if-Vl10)# ip virtual-router address 10.0.0.6
switch(config-if-Vl10)# ipv6 address 2001::1/64
switch(config-if-Vl10)# ipv6 virtual-router address 2001::2
switch(config-if-Vl10)# exit
switch(config)#
```

- These commands configure a Switch Virtual Interface (SVI) and a virtual IPv4 address with a subnet for ***vlan 10***. A static route is added to indicate that the virtual subnet is reachable through ***vlan 10***.

```
switch(config)# ip route 192.0.0.0/24 vlan 10
switch(config)# interface vlan 10
switch(config-if-Vl10)# ip address 10.0.0.2/24
switch(config-if-Vl10)# ip virtual-router address 192.0.0.6/24
switch(config-if-Vl10)# exit
switch(config)#
```

Virtual MAC Address

The **ip virtual-router mac-address** command assigns a virtual MAC address to the switch. The switch maps all virtual router IP addresses to this MAC address. The address is receive-only; the switch never sends packets with this address as the source.

When the destination MAC of a packet destined to a remote network matches the virtual MAC address, the MLAG peer forwards the traffic to the next hop destination. Each MLAG peer must have the same routes available, either though static configuration or learned through a dynamic routing protocol.

Example

This command configures a virtual MAC address.

```
switch(config)# ip virtual-router mac-address 001c.7300.0099
switch(config)#
```

Show Virtual MAC Address

To display the virtual router MAC and IP addresses, enter the **show ip virtual-router** command.

Example

This command displays the virtual router addresses assigned on the switch.

```
switch# show ip virtual-router
IP virtual router is configured with MAC address: 24cd.5a29.cc31
Interface  IP Address        Virtual IP Address   Status           Protocol
```

```
Vlan15      10.1.1.3/24       10.1.1.15          up                  up
Vlan15      10.1.1.3/24       10.1.1.16          up                  up
Vlan15      10.1.1.3/24       10.1.1.17          up                  up
Vlan20      10.12.1.6/24      10.1.1.51          up                  up
Vlan20      10.12.1.6/24      10.1.1.53          up                  up
Vlan20      10.12.1.6/24      10.1.1.55          up                  up switch#
```

Show IPv6 Virtual-Router

The **show ipv6 virtual-router** command displays the virtual MAC address assigned to the switch and all virtual IPv6 addresses assigned to each VLAN interface.

Example

This command displays a table of information for IPv6 VRRP groups on the switch.

```
switch# show ipv6 virtual-router
IP virtual router is configured with MAC address: 001c.7300.0099
MAC address advertisement interval: 30 seconds
Interface Vlan4094
  State is up
  Protocol is up
  IPv6 address
    2001:b8:2001::1011/64
  Virtual IPv6 address
    2001:db8:ac10:fe01::
switch#
```

13.5.3 VRRP and VARP Implementation Examples

This section contains the following example set:

- VRRP Examples
- VARP Example

13.5.3.1 VRRP Examples

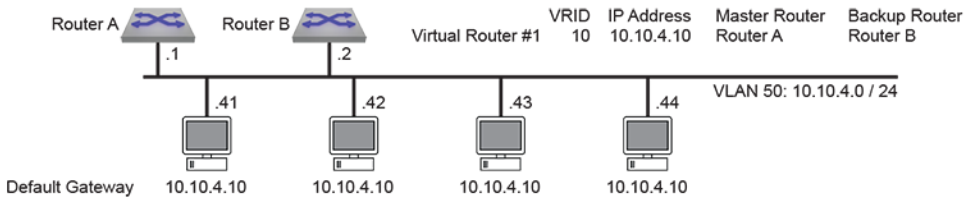
This section provides code that implements three VRRP configurations:

- Example 1 configures two switches in a single virtual router group. This implementation protects the LAN against the failure of one router.
- Example 2 configures two switches into two virtual routers within a single LAN. This implementation protects the LAN against the failure of one router and balances traffic between the routers.
- Example 3 configures three switches to implement virtual routers on two LANs. Each LAN contains two virtual routers. One switch is configured into four virtual routers – two on each LAN.

13.5.3.1.1 VRRP Example 1: One Virtual Router on One LAN

The network diagram displays the Example 1 network. Two switches are configured as VRRP routers to form one virtual router.

Figure 13-3: VRRP Example 1 Network Diagram



The following code configures the first switch (***Router A***) as the master router and the second switch (***Router B***) as a backup router for virtual router ***10*** on ***vlan 50***. ***Router A*** becomes the Master virtual router by setting its priority at ***200***; ***Router B*** maintains the default priority of ***100***. The advertisement interval is three seconds on both switches. Priority preemption is enabled by default.

Switch Code that Implements Router A on the First Switch

```
switch-A(config)# interface vlan 50
switch-A(config-if-vl50)# ip address 10.10.4.1/24
switch-A(config-if-vl50)# no vrrp 10
switch-A(config-if-vl50)# vrrp 10 priority 200
switch-A(config-if-vl50)# vrrp 10 advertisement interval 3
switch-A(config-if-vl50)# vrrp 10 ip 10.10.4.10
switch-A(config-if-vl50)# exit
```

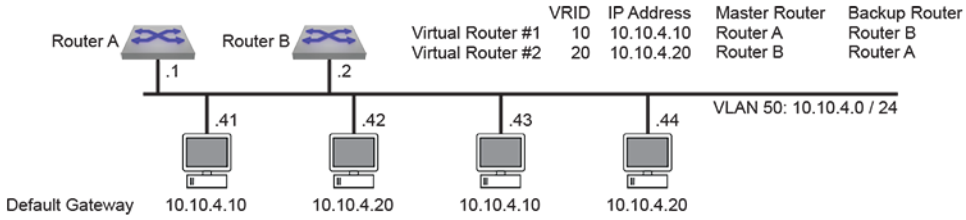
Switch Code that Implements Router B on the Second Switch

```
switch-B(config)# interface vlan 50
switch-B(config-if-vl50)# ip address 10.10.4.2/24
switch-B(config-if-vl50)# no vrrp 10
switch-B(config-if-vl50)# vrrp 10 advertisement interval 3
switch-B(config-if-vl50)# vrrp 10 ip 10.10.4.10
switch-B(config-if-vl50)# exit
```

13.5.3.1.2 VRRP Example 2: Two Virtual Routers on One LAN

The nnetwork diagram displays Example 2. Two switches are configured as VRRP routers to form two virtual routers on one LAN. Using two virtual routers distributes the LAN traffic between the switches.

Figure 13-4: VRRP Example 2 Network Diagram



The following code configures two switches as a master and a backup router for two virtual routers on ***vlan 50***.

- ***Router A*** is the master for virtual router ***10*** and backup for virtual router ***20***.
- ***Router B*** is the master for virtual router ***20*** and backup for virtual router ***10***.
- VRRP advertisement interval is ***3*** seconds on virtual router ***10*** and ***5*** seconds on virtual router ***20***.
- Priority preemption is enabled by default for both virtual routers.

Switch Code that Implements Router A on the First Switch

```
switch-A(config)# interface vlan 50
switch-A(config-if-vl50)# ip address 10.10.4.1/24
switch-A(config-if-vl50)# no vrrp 10
switch-A(config-if-vl50)# vrrp 10 priority 200
switch-A(config-if-vl50)# vrrp 10 advertisement interval 3
switch-A(config-if-vl50)# vrrp 10 ip 10.10.4.10
switch-A(config-if-vl50)# no vrrp 20
switch-A(config-if-vl50)# vrrp 20 advertisement interval 5
```

```
switch-A(config-if-vl50)# vrrp 20 ip 10.10.4.20
switch-A(config-if-vl50)# exit
```

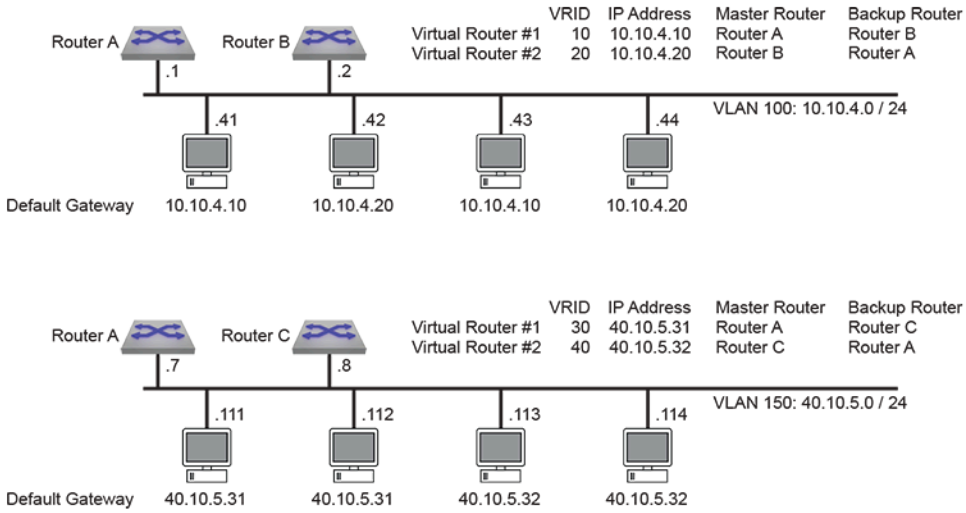
Switch Code that Implements Router B on the Second Switch

```
switch-B(config)# interface vlan 50
switch-B(config-if-vl50)# ip address 10.10.4.2/24
switch-B(config-if-vl50)# no vrrp 10
switch-B(config-if-vl50)# vrrp 10 advertisement interval 3
switch-B(config-if-vl50)# vrrp 10 ip 10.10.4.10
switch-B(config-if-vl50)# no vrrp 20
switch-B(config-if-vl50)# vrrp 20 priority 200
switch-B(config-if-vl50)# vrrp 20 advertisement interval 5
switch-B(config-if-vl50)# vrrp 20 ip 10.10.4.20
switch-B(config-if-vl50)# exit
```

13.5.3.1.3 VRRP Example 3: Two Virtual Routers on Two LANs

The network diagram displays Example 3. Three switches are configured as VRRP routers to form four virtual router groups two groups on each of two LANs.

Figure 13-5: VRRP Example 3 Network Diagram



The following code configures the three switches as follows:

- ***Router A*** is the master for virtual router ***10*** and backup for virtual router ***20*** on ***vlan 100***.
- ***Router A*** is the master for virtual router ***30*** and backup for virtual router ***40*** on ***vlan 150***.
- ***Router B*** is the master for virtual router ***20*** and backup for virtual router ***10*** on ***vlan 100***.
- ***Router C*** is the master for virtual router ***40*** and backup for virtual router ***30*** on ***vlan 150***.
- VRRP advertisement interval is set to one second on all virtual routers.
- Priority preemption is disabled on all virtual routers.

Switch Code that Implements Router A on the First Switch

```
switch-A(config)# interface vlan 100
switch-A(config-if-vl100)# ip address 10.10.4.1/24
switch-A(config-if-vl100)# no vrrp 10
switch-A(config-if-vl100)# vrrp 10 priority 200
switch-A(config-if-vl100)# no vrrp 10 preempt
switch-A(config-if-vl100)# vrrp 10 ip 10.10.4.10
```

```
switch-A(config-if-vl100)# vrrp 10 advertisement interval 1
switch-A(config-if-vl100)# no vrrp 20
switch-A(config-if-vl100)# no vrrp 20 preempt
switch-A(config-if-vl100)# vrrp 20 ip 10.10.4.20
switch-A(config-if-vl100)# interface vlan 150
switch-A(config-if-vl150)# ip address 40.10.5.7/24
switch-A(config-if-vl150)# no vrrp 30
switch-A(config-if-vl150)# vrrp 30 priority 200
switch-A(config-if-vl150)# no vrrp 30 preempt
switch-A(config-if-vl150)# vrrp 30 ip 40.10.5.31
switch-A(config-if-vl100)# vrrp 30 advertisement interval 1
switch-A(config-if-vl150)# no vrrp 40
switch-A(config-if-vl150)# no vrrp 40 preempt
switch-A(config-if-vl150)# vrrp 40 ip 40.10.5.32
switch-A(config-if-vl150)# exit
```

Switch Code that Implements Router B on the Second Switch

```
switch-B(config)# interface vlan 100
switch-B(config-if-vl100)# ip address 10.10.4.2/24
switch-B(config-if-vl100)# no vrrp 10
switch-B(config-if-vl100)# no vrrp 10 preempt
switch-B(config-if-vl100)# vrrp 10 ip 10.10.4.10
switch-B(config-if-vl100)# no vrrp 20
switch-B(config-if-vl100)# vrrp 20 priority 200
switch-B(config-if-vl100)# no vrrp 20 preempt
switch-B(config-if-vl100)# vrrp 20 ip 10.10.4.20
switch-A(config-if-vl100)# vrrp 20 advertisement interval 1
switch-B(config-if-vl100)# exit
```

Switch Code that Implements Router C on the Third Switch

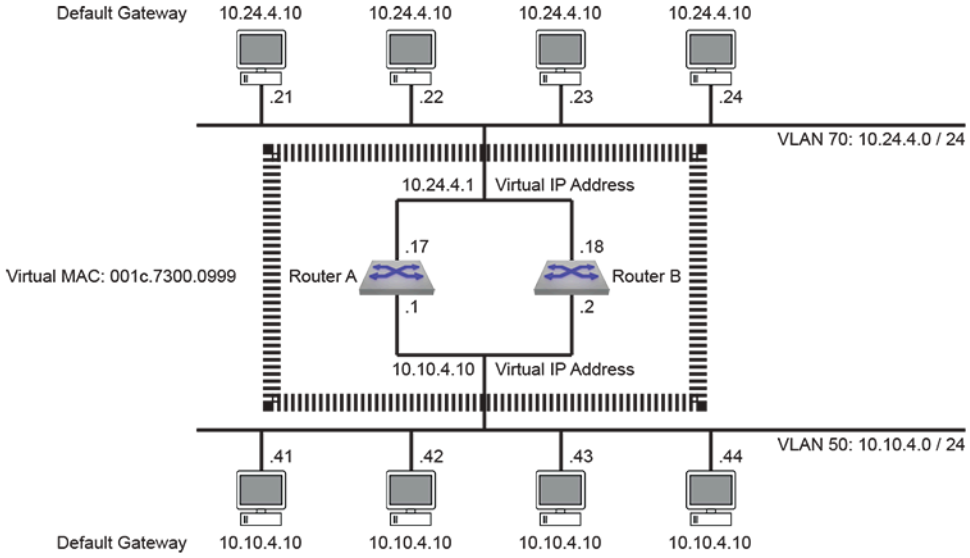
```
switch-C(config)# interface vlan 150
switch-C(config-if-vl150)# ip address 40.10.5.8/24
switch-C(config-if-vl150)# no vrrp 30
switch-C(config-if-vl150)# no vrrp 30 preempt
switch-C(config-if-vl150)# vrrp 30 ip 40.10.5.31
switch-C(config-if-vl150)# no vrrp 40
switch-C(config-if-vl150)# vrrp 40 priority 200
switch-C(config-if-vl150)# no vrrp 40 preempt
switch-C(config-if-vl150)# vrrp 40 ip 40.10.5.32
switch-A(config-if-vl100)# vrrp 40 advertisement interval 1
switch-C(config-if-vl150)# exit
```

13.5.3.2 VARP Example

This section provides code that implements a VARP configuration. The network diagram displays the Example 1 network. Two switches in an MLAG domain are configured as VARP routers.

The following code configures ***10.10.4.10*** as the virtual IP address for ***VLAN 50***, ***10.24.4.1*** as the virtual IP address for ***VLAN 70***, and ***001c.7300.0999*** as the virtual MAC address on both switches.

Figure 13-6: VARP Example Network Diagram



Switch Code that Implements VARP on the First Switch

```
switch-A(config)# ip virtual-router mac-address 001c.7300.0999
switch-A(config)# interface vlan 50
switch-A(config-if-vl50)# ip address 10.10.4.1/24
switch-A(config-if-vl50)# ip virtual-router address 10.10.4.10
switch-A(config-if-vl50)# interface vlan 70
switch-A(config-if-vl70)# ip address 10.24.4.17/24
switch-A(config-if-vl70)# ip virtual-router address 10.24.4.1
switch-A(config-if-vl70)# exit
```

Switch Code that Implements VARP on the Second Switch

```
switch-B(config)# ip virtual-router mac-address 001c.7300.0999
switch-B(config)# interface vlan 50
switch-B(config-if-vl50)# ip address 10.10.4.2/24
switch-B(config-if-vl50)# ip virtual-router address 10.10.4.10
switch-B(config-if-vl50)# interface vlan 70
switch-B(config-if-vl70)# ip address 10.24.4.18/24
switch-B(config-if-vl70)# ip virtual-router address 10.24.4.1
switch-B(config-if-vl70)# exit
```

13.5.4 VRRP and VARP Configuration Commands

This section contains descriptions of CLI commands that support VRRP and VARP.

Global Configuration Commands

- ip fhrp accept-mode
- ip virtual-router mac-address
- ip virtual-router mac-address advertisement-interval

Interface Configuration Commands Ethernet, Port Channel, and VLAN Interfaces

- ip virtual-router address
- ipv6 virtual-router address
- no vrrp
- vrrp advertisement interval
- vrrp disabled
- vrrp ipv4
- vrrp ipv4 checksum pseudo-header exclude
- vrrp ipv4 secondary
- vrrp ipv4 version
- vrrp ipv6
- vrrp mac-address advertisement-interval
- vrrp peer authentication
- vrrp preempt
- vrrp preempt delay
- vrrp priority-level
- vrrp session description
- vrrp timers delay reload
- vrrp tracked-object

Privileged EXEC Commands

- show ip virtual-router
- show ipv6 virtual-router
- show vrrp

13.5.4.1 ip fhrp accept-mode

The **`ip fhrp accept-mode`** command configures the switch to permit SSH access to the VRRP Master and VARP Master router. All routers within a VRRP or VARP group should be configured consistently. By default, SSH access to the VRRP and VARP Master routers is not permitted.

The **`no ip fhrp accept-mode`** and **`default ip fhrp accept-mode`** commands restores the default SSH access availability by removing the **`ip fhrp accept-mode`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip fhrp accept-mode`

`no ip fhrp accept-mode`

`default ip fhrp accept-mode`

Example

This command configures the switch to permit SSH access to the VRRP and VARP Master routers.

```
switch(config)# ip fhrp accept-mode
switch(config)# show running-config

!
ip fhrp accept-mode
!
```

```
switch(config)#
```

13.5.4.2 ip virtual-router address

The **ip virtual-router address** command assigns a virtual IPv4 address to the VLAN interface being configured. (To assign a virtual IPv6 address to a VLAN interface, use the **ipv6 virtual-router address** command.) Unlike VRRP, the virtual IP address does not have to be in the same subnet as the physical interface.

A virtual IP address may optionally be configured with a subnet, but doing so will modify the behavior of ARP requests sent from the router. When the router sends an ARP request for an IP address in a virtual subnet, the ARP request will use the virtual IP address as the source IP address and the virtual MAC address as the source MAC address inside the ARP header. For virtual IP addresses configured without the subnet option, no modifications are made to outgoing ARP requests.

A maximum of ***500*** virtual IP addresses can be assigned to a VLAN interface. All virtual addresses on all VLAN interfaces resolve to the same virtual MAC address configured through the **ip virtual-router mac-address** command.

This command is typically used in MLAG configurations to create identical virtual routers on switches connected to the MLAG domain through an MLAG.

The **no ip virtual-router address** and **default ip virtual-router address** commands remove the specified virtual IP address from the configuration mode interface by deleting the corresponding **ip virtual-router address** command from ***running-config***. If the command does not specify an address, all virtual IPv4 addresses are removed from the interface.

Command Mode

Interface-VLAN Configuration

Command Syntax

ip virtual-router address *ipv4_addr*

no ip virtual-router address [*ipv4_addr*]

default ip virtual-router address [*ipv4_addr*]

Parameter

ipv4_addr IP address of router. Dotted decimal notation.

Examples

- These commands configure a Switch Virtual Interface (SVI) and a virtual IP address for ***vlan 10***.

```
switch(config)# interface vlan 10
switch(config-if-Vl10)# ip address 10.0.0.2/24
switch(config-if-Vl10)# ip virtual-router address 10.0.0.6
switch(config-if-Vl10)# exit
```

- These commands configure a Switch Virtual Interface (SVI) and a virtual IP address with a subnet for ***vlan 10***. A static route is added to indicate that the virtual subnet is reachable through ***vlan 10***.

```
switch(config)# ip route 192.0.0.0/24 vlan 10
switch(config)# interface vlan 10
switch(config-if-Vl10)# ip address 10.0.0.2/24
switch(config-if-Vl10)# ip virtual-router address 192.0.0.6/24
switch(config-if-Vl10)# exit
```

13.5.4.3 ip virtual-router mac-address

The **ip virtual-router mac-address** command assigns a virtual MAC address to the switch. The switch maps all virtual router IP addresses to this MAC address. The address is receive-only; the switch never sends packets with this address as the source. The virtual router is not configured on the switch until this virtual mac-address is assigned.

This command is typically used in MLAG configurations to create identical virtual routers on switches connected to the MLAG domain through an MLAG. When the destination MAC of a packet destined to a remote network matches the virtual MAC address, the MLAG peer forwards the traffic to the next hop destination. Each MLAG peer must have the same routes available, either though static configuration or learned through a dynamic routing protocol.

The **no ip virtual-router mac-address** command removes a virtual MAC address from the interface by deleting the corresponding **ip virtual-router mac-address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip virtual-router mac-address ***mac_addr***

no ip virtual-router mac address [***mac_addr***]

Parameter

mac_addr MAC IP address (dotted hex notation). Select an address that will not otherwise appear on the switch.

Example

This command configures a virtual MAC address.

```
switch(config)# ip virtual-router mac-address 001c.7300.0099
switch(config)#
```

13.5.4.4 ip virtual-router mac-address advertisement-interval

The **ip virtual-router mac-address advertisement interval** command specifies the period between the transmission of consecutive gratuitous ARP requests that contain the virtual router mac address for each virtual-router IP address configured on the switch. The default period is ***30*** seconds.

The **no ip virtual-router mac-address advertisement-interval** command restores the default period of ***30*** seconds by removing the **ip virtual-router mac-address advertisement-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip virtual-router mac-address advertisement-interval ***period***

no ip virtual-router mac-address advertisement-interval

default ip virtual-router mac-address advertisement-interval

Parameter

period advertisement interval (seconds). Values range from ***0 to 86400***. Default is ***30***.

Example

This command configures a MAC address advertisement interval of one minute (***60*** seconds).

```
switch(config)# ip virtual-router mac-address advertisement-interval 60
switch(config)#
```

13.5.4.5 ipv6 virtual-router address

The **ipv6 virtual-router address** command assigns a virtual IPv6 address to the VLAN interface being configured. (To assign a virtual IPv4 address to a VLAN interface, use the **ip virtual-router address** command.) Unlike VRRP, the virtual IP address does not have to be in the same subnet as the physical interface.

A maximum of ***500*** virtual IP addresses can be assigned to a VLAN interface. All virtual addresses on all VLAN interfaces resolve to the same virtual MAC address configured through the **ip virtual-router mac-address** command.

This command is typically used in MLAG configurations to create identical virtual routers on switches connected to the MLAG domain through an MLAG.

The **no ipv6 virtual-router address** and **default ipv6 virtual-router address** commands remove the specified virtual IPv6 address from the configuration mode interface by deleting the corresponding **ipv6 virtual-router address** command from ***running-config***. If the command does not specify an address, all virtual IPv6 addresses are removed from the interface.

Command Mode

Interface-VLAN Configuration

Command Syntax

ipv6 virtual-router address ***net_addr***

no ipv6 virtual-router address [***net_addr***]

default ipv6 virtual-router address [***net_addr***]

Parameter

net_addr network IPv6 address.

Example

These commands configure a Switch Virtual Interface (SVI) and a virtual IPv6 address for ***vlan 10***.

```
switch(config)# interface vlan 10
switch(config-if-Vl10)# ipv6 address 2001:0DB8:0:1::1/64
switch(config-if-Vl10)# ipv6 virtual-router address 2001:0DB8:0:1::2
switch(config-if-Vl10)# exit
```

13.5.4.6 no vrrp

The **no vrrp** command removes all VRRP configuration commands for the specified virtual router on the configuration mode interface. The **default vrrp** command also reverts VRRP configuration parameters to default settings by removing the corresponding **vrrp** commands.

Commands removed by the **no vrrp** command include:

- vrrp advertisement interval
- vrrp disabled
- vrrp ipv4
- vrrp ipv4 secondary
- vrrp peer authentication

- vrrp preempt
- vrrp preempt delay
- vrrp priority-level
- vrrp session description

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`no vrrp` ***group***

`default vrrp` ***group***

Parameter

group Virtual Router Identifier (VRID). Values range from ***1 to 255***.

Example

This command removes all VRRP configuration commands for virtual router ***group 10*** on ***vlan 15***.

```
switch(config)# interface vlan 15
switch(config-if-vl15)# no vrrp 10
switch(config-if-vl15)#
```

13.5.4.7 show ip virtual-router

The **`show ip virtual-router`** command displays the virtual MAC address assigned to the switch and all virtual IP addresses assigned to each VLAN interface.

Command Mode

EXEC

Command Syntax

`show ip virtual-router`

Messages

- **IP virtual router is not configured** a virtual MAC address is not assigned to the switch.
- **No interface with virtual IP address** no virtual IP addresses are assigned to any VLAN interfaces.

Examples

- This command displays a table of information for VRRP groups on the switch.

```
switch# show ip virtual-router
IP virtual router is configured with MAC address: 24cd.5a29.cc31
Interface  IP Address        Virtual IP Address   Status            Protocol
Vlan15     10.1.1.3/24       10.1.1.15            up                up
Vlan15     10.1.1.3/24       10.1.1.16            up                up
Vlan15     10.1.1.3/24       10.1.1.17            up                up
Vlan20     10.12.1.6/24      10.1.1.51            up                up
Vlan20     10.12.1.6/24      10.1.1.53            up                up
Vlan20     10.12.1.6/24      10.1.1.55            up                up
 switch#
```

- This command generates a response that indicates a virtual MAC address is not assigned to the switch.

```
switch# show ip virtual-router
IP virtual router is not configured switch#
```

13.5.4.8 show ipv6 virtual-router

The **show ipv6 virtual-router** command displays the virtual MAC address assigned to the switch and all virtual IPv6 addresses assigned to each VLAN interface.

Command Mode

EXEC

Command Syntax

show ipv6 virtual-router

Messages

- **IPv6 virtual router is not configured** a virtual MAC address is not assigned to the switch.
- **No interface with virtual IPv6 address** no virtual IPv6 addresses are assigned to any VLAN interfaces.

Example

This command displays a table of information for IPv6 VRRP groups on the switch.

```
switch# show ipv6 virtual-router
IP virtual router is configured with MAC address: 001c.7300.0099
MAC address advertisement interval: 30 seconds
Interface Vlan4094
  State is up
  Protocol is up
  IPv6 address
    2001:b8:2001::1011/64
  Virtual IPv6 address
    2001:db8:ac10:fe01::
switch#
```

13.5.4.9 show vrrp

The **show vrrp** command displays information about the Virtual Router Redundancy Protocol (VRRP) groups configured on a specified interface. Parameter options control the amount and formatting of the displayed information.

Command Mode

Privileged EXEC

Command Syntax

show vrrp [**INFO_LEVEL**] [**STATES**]

show vrrp **INTF GROUP_NUM** [**INFO_LEVEL**] [**STATES**]

show vrrp **GROUP_NUM INTF_GROUP** [**INFO_LEVEL**] [**STATES**]

Parameters

- **INTF** specifies the VRRP groups for which the command displays status. When the parameter is omitted or specifies only an interface, the group list is filtered by the STATES parameter.
  - ***no parameter*** specified groups on all interfaces.
  - **interface ethernet *e_num*** specified groups on Ethernet interface.
  - **interface loopback *l_num*** specified groups on loopback interface.

  - **interface management *m_num*** specified groups on management interface.
  - **interface port-channel *p_num*** specified groups on port channel interface.
  - **interface vlan *v_num*** specified groups on VLAN interface.
  - **interface vxlan *vx_num*** specified groups on VXLAN interface.
- **GROUP_NUM** the VRRP ID number of the group for which the command displays status.
  - ***no parameter*** all groups on specified interface.
  - ***vrid_num*** Virtual Router Identifier (VRID). Value ranges from ***1*** to ***255***.
- **INFO_LEVEL** Specifies format and amount of displayed information. Options include:
  - ***no parameter*** displays a block of data for each VRRP group.
  - **brief** displays a single table that lists information for all VRRP groups.
- **STATES** Specifies the groups, by VRRP router state, that are displayed. Options include:
  - ***no parameter*** displays data for groups in the **master** or **backup** states.
  - **all** displays all groups, including groups in the **stopped** and **interface down** states.

Examples

- This command displays a table of information for VRRP groups on the switch.

```
switch# show vrrp brief
Interface Id  Ver Pri Time  State   VrIps
Vlan1006  3   2   100 3609  Master  127.38.10.2
Vlan1006  4   3   100 3609  Master  127.38.10.10
Vlan1010  1   2   100 3609  Master  128.44.5.3
Vlan1014  2   2   100 3609  Master  127.16.14.2 switch>
```

- This command displays data blocks for all VRRP groups on ***vlan 46***, regardless of the VRRP state.

```
switch# show vrrp interface vlan 1006 all
Vlan1010 - Group 1
  VRRP Version 2
  State is Stopped
  Virtual IPv4 address is 128.44.5.3
  Virtual MAC address is 0000.5e00.0101
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 0.0.0.0
  Master Advertisement interval is 1s
  Skew time is 0.609s
  Master Down interval is 3.609s switch#
```

- This command displays data for all VRRP group ***2*** on ***vlan 1014***.

```
switch# show vrrp interface vlan 1014 group 2
Vlan1006 - Group 2
  VRRP Version 2
  State is Master
  Virtual IPv4 address is 127.38.10.2
  Virtual MAC address is 0000.5e00.0103
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 127.38.10.1 (local), priority is 100
```

```
Master Advertisement interval is 1s
Skew time is 0.609s
Master Down interval is 3.609s switch#
```

13.5.4.10 vrrp advertisement interval

The **vrrp advertisement interval** command configures the interval between successive advertisement messages that the switch sends to VRRP routers in the specified virtual router group. The switch must be the groups Master virtual router to send advertisement messages. The advertisement interval must be configured identically on all physical routers in the virtual router group.

The advertisement interval also influences the timeout interval that defines when the virtual router becomes the master virtual router. When preemption is enabled, the virtual router becomes the master when three times the advertisement interval elapses after the switch detects master router priority conditions.

The **no vrrp advertisement interval** and **default vrrp advertisement interval** commands restore the default advertisement interval of one second for the specified virtual router by removing the corresponding **vrrp advertisement interval** command from ***running-config***. The **no vrrp** command also removes the **vrrp advertisement interval** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp ***group*** **advertisement interval** ***adv_time***

no vrrp ***group*** **advertisement interval**

default vrrp ***group*** **advertisement interval**

Parameters

- ***group*** Virtual Router Identifier (VRID). Values range from ***1 to 255***.
- ***adv_time*** advertisement interval (seconds). Values range from ***1 to 255***. Default value is ***1***.

Example

This command sets the advertisement interval of five seconds for the virtual router ***35*** on ***vlan 100***.

```
switch(config)# interface vlan 100
switch(config-if-vl100)# vrrp 35 advertisement interval 5
switch(config-if-vl100)#
```

13.5.4.11 vrrp disabled

The **vrrp disabled** command places the switch in stopped state for the specified virtual router. While in stopped state, the switch cannot act as a ***master*** or ***backup*** router for the virtual router group.

The **no vrrp disabled** and **default vrrp disabled** commands remove the corresponding **vrrp disabled** command from ***running-config***. This changes the switchs virtual router state to ***backup*** or ***master*** if the virtual router is properly configured.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp *group* disabled

no vrrp *group* disabled

default vrrp *group* disabled

Parameter

group Virtual Router IDentifier (VRID). Values range from ***1 to 255***.

Examples

- These commands place the switch in stopped mode for virtual router ***24*** on ***vlan 20***.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 24 disabled
switch(config-if-vl20)#
```

- This command moves the switch out of stopped mode for virtual router ***24*** on ***vlan 20***.

```
switch(config-if-vl20)# no vrrp 24 disabled
switch(config-if-vl20)#
```

13.5.4.12 vrrp ipv4

The **vrrp ipv4** command configures the primary IP address for the specified VRRP virtual router. The command also activates the virtual router if the primary address is contained in the interfaces subnet. A VRRP virtual routers configuration may contain only one primary IP address assignment command; subsequent **vrrp ipv4** commands replace the existing primary address assignment.

The **vrrp ipv4 secondary** command assigns a secondary IP address to the VRRP virtual router.

The **no vrrp ipv4**and **default vrrp ipv4** commands disable the VRRP virtual router and deletes the primary IP address by removing the corresponding **vrrp ipv4** statement from ***running-config.*** The **no vrrp** command also removes the **vrrp ipv4** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp *group* ipv4 *ipv4_address*

no vrrp *group* ipv4 *ipv4_address*

default vrrp *group* ipv4 *ipv4_address*

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1 to 255***.
- ***ipv4_address*** IPv4 address of the virtual router.

Related Command

vrrp ipv4 secondary

Example

This command enables virtual router ***15*** on ***vlan 20*** and designates ***10.1.1.5*** as the virtual routers primary address.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 15 ipv4 10.1.1.5
switch(config-if-vl20)#
```

13.5.4.13 vrrp ipv4 checksum pseudo-header exclude

This command excludes the pseudo-header in IPv4 VRRPv3 checksum calculation on the VRRP group on the configuration mode interface of the switch and supports IPv4 VRRPv3 interoperability.

The **no** form of the command deletes the **`vrrp ipv4 checksum pseudo-header exclude`** configuration from the Ethernet interface on the switch.

The **`exit`** command returns the switch to global configuration mode.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`vrrp` ***group*** **`ipv4 checksum pseudo-header exclude`**

`no vrrp` ***group*** **`ipv4 checksum pseudo-header exclude`**

Parameter

group Virtual Router IDentifier (VRID). Values range from ***1 to 255***.

Example

This command excludes the pseudo-header in IPv4 VRRPv3 checksum calculation on VRRP ***group 1*** on ***interface ethernet 1***.

```
switch(config-if-Et1)# vrrp 1 ipv4 checksum pseudo-header exclude
```

13.5.4.14 vrrp ipv4 secondary

The **`vrrp ipv4 secondary`** command assigns a secondary IP address to the specified virtual router. Secondary IP addresses are an optional virtual router parameter. A virtual router may contain multiple secondary address commands. The IP address list must be identical for all VRRP routers in a virtual router group.

The virtual router is assigned a primary IP address with the **`vrrp ipv4`** command.

The **`no vrrp ipv4 secondary`** and **`default vrrp ipv4 secondary`** commands remove the secondary IP address for the specified VRRP virtual router by deleting the corresponding **`vrrp ipv4 secondary`** statement from running-config. The **`no vrrp`** command also removes all **`vrrp ipv4 secondary`** commands for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp *group* ipv4*ipv4_addr* secondary

no vrrp *group* ipv4*ipv4_addr* secondary

default vrrp *group* ipv4*ipv4_addr* secondary

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1 to 255***.
- ***ipv4_addr*** secondary IPv4 address of the virtual router.

Related Command

vrrp ipv4

Example

This command assigns the IP address of ***10.2.4.5*** as the secondary IP address for the virtual router with VRID of ***15*** on ***vlan 20***.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 15 ipv4 10.2.4.5 secondary
switch(config-if-vl20)#
```

13.5.4.15 vrrp ipv4 version

The **vrrp ipv4 version** command enables VRRP on the configuration mode interface and configures the VRRP version for the specified VRRP virtual router.

The **no vrrp ipv4 version** and **default vrrp ipv4 version** commands restore the default VRRP version to VRRPv2 by removing the corresponding **vrrp ipv4** version statement from ***running-config.***

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp *group* ipv4 version VERSION_NUMBER

no vrrp *group* ipv4 version

default vrrp *group* ipv4 version

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1 to 255***.
- **VERSION_NUMBER** Specifies VRRP version that the switch uses. Default value is 2 (VRRPv2) Options include:
  - **2** VRRP v2 supports IPv4 environment.
  - **3** VRRP v3 supports IPv4 and IPv6 environment.

Examples

- This command enables VRRPv3 for IPv6 on ***interface ethernet 3***.

```
switch#(config)# interface ethernet 3
switch#(config-if-Et3)# vrrp 1 ipv4 version 3
```

- This command removes VRRPv3 from ***interface ethernet 3*** and reverts to the default VRRPv2.

```
switch#(config)# interface ethernet 3
```

```
switch#(config-if-Et3)# no vrrp 1 ipv4 version
```

13.5.4.16 vrrp ipv6

The **vrrp ipv6** command configures the IPv6 address for the specified VRRP virtual router. The command also activates the virtual router if the primary address is contained in the interfaces subnet.

The **no vrrp ipv6** and **default vrrp ipv6** commands disable the VRRP virtual router and deletes the IPv6 address by removing the corresponding **vrrp ipv6** statement from ***running-config.*** The **no vrrp** command also removes the **vrrp ipv6** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp ***group*** **ip** ***ipv6_address***

no vrrp ***group*** **ip** ***ipv6_address***

default vrrp ***group*** **ip** ***ipv6_address***

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1*** to ***255***.
- ***ipv6_address*** IPv6 address of the virtual router.

Examples

- This command enables address ***2001:db8:0:1::1*** for IPv6 VRRP on ***vlan 20***.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 3 ipv6 2001:db8:0:1::1
switch(config-if-vl20)#
```

- This command disables VRRPv3 on ***vlan 20*** from virtual router ***3***.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# no vrrp 3 ipv6 2001:db8:0:1::1
switch(config-if-vl20)#
```

13.5.4.17 vrrp mac-address advertisement-interval

The **vrrp mac-address advertisement-interval** command specifies the interval between advertisement packets sent by the master router to the VRRP group members.

The **vrrp mac-address advertisement-interval 0**, **no vrrp mac-address advertisement-interval** and **default vrrp mac-address advertisement-interval** commands disable the feature by removing the **vrrp mac-address advertisement-interval** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp *group* mac-address advertisement-interval *period*

no vrrp *group* mac-address

default vrrp *group* mac-address

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1 to 255***.
- ***period*** interval in which the master router sends advertisement packets (seconds). Value ranges from ***0 to 3600***. Selecting ***0*** as the interval disables this feature.

Examples

- This command specifies the interval between advertisement packets sent to the members of VRRP group ***3*** on ***vlan 20***.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 3 mac-address advertisement-interval 60
switch(config-if-vl20)#
```

- This command disables the feature on ***vlan 20***.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# no vrrp 3 mac-address advertisement-interval
switch(config-if-vl20)#
```

13.5.4.18 vrrp peer authentication

The **vrrp peer authentication** command configures parameters the switch uses to authenticate virtual router packets it receives from other VRRP routers in the group.

The **no vrrp peer authentication** and **default vrrp peer authentication** commands disable VRRP peer authentication of packets from the specified virtual router by removing the corresponding **vrrp peer authentication** command from ***running-config*** . The **no vrrp** command also removes the **vrrp peer authentication** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp *group* peer authentication AUTH_PARAMETER

no vrrp *group* peer authentication

default vrrp *group* peer authentication

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1 to 255***.
- **AUTH_PARAMETER** encryption level and authentication key used by router. Options include:
  - **text *text_key*** plain-text authentication, ***text_key*** is text.
  - ***text_key*** plain-text authentication, ***text_key*** is text.
  - **ietf-md5 key-string 0 *text_key*** IP authentication of MD5 key hash, ***text_key*** is text.
  - **ietf-md5 key-string *text_key*** IP authentication of MD5 key hash, ***text_key*** is text.
  - **ietf-md5 key-string 7 *coded_key*** IP authentication of MD5 key hash, ***coded_key*** is MD5 hash.

Guidelines

This command is applicable to VRRPv2 which supports IPv4 addresses only.

Examples

- This command implements plain-text authentication, using ***12345*** as the key, for virtual router ***40*** on ***vlan 100***.

```
switch(config)# interface vlan 100
switch(config-if-vl100)# vrrp 40 peer authentication text 12345
switch(config-if-vl100)#
```

- This command implements ietf-md5 authentication, using ***12345*** as the key.

```
switch(config-if-vl100)# vrrp 40 peer authentication ietf-md5 key-string 0
 12345
switch(config-if-vl100)#
```

- This command implements ietf-md5 authentication, using ***12345*** as the key. The key is entered as the MD5 hash equivalent of the text string.

```
switch(config-if-vl100)# vrrp 40 peer authentication ietf-md5 key-string 7
EA3TUPxdddFCLYT8mb+kxw==
switch(config-if-vl100)#
```

13.5.4.19 vrrp preempt

The **`vrrp preempt`** command controls a virtual routers preempt mode setting. When preempt mode is enabled, if the switch has a higher priority it will preempt the current master virtual router. When preempt mode is disabled, the switch can become the master virtual router only when a master virtual router is not present on the subnet, regardless of VRRP priority level settings. By default, preempt mode is enabled.

The **`no vrrp preempt`**cand **`default vrrp preempt`**ccommands disable preempt mode for the specified virtual router; the **`default vrrp preempt`** command stores a corresponding **`no vrrp preempt`** statement in ***running-config***. The **`vrrp preempt`** command enables preempt mode by removing the corresponding **`no vrrp preempt`** statement from running-config.

The **`no vrrp`** command also enables preempt mode by removing the **`no vrrp preempt`** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`vrrp` ***`group`*** **`preempt`**

`no vrrp` ***`group`*** **`preempt`**

`default vrrp` ***`group`*** **`preempt`**

Parameter

group Virtual Router IDentifier (VRID). Values range from ***1 to 255***.

Related Command

vrrp preempt delay

Examples

- This command disables preempt mode for virtual router ***20*** on ***vlan 40***.

```
switch(config)# interface vlan 40
switch(config-if-vl40)# no vrrp 20 preempt
switch(config-if-vl40)#
```

- This command enables preempt mode for virtual router ***20*** on ***vlan 40***.

```
switch(config-if-vl40)# vrrp 20 preempt
switch(config-if-vl40)#
```

13.5.4.20 vrrp preempt delay

The **`vrrp preempt delay`** command specifies the interval between a VRRP preemption event and the point when the switch becomes the master VRRP router. A preemption event is any event that results in the switch having the highest virtual router priority setting while preemption is enabled. The **`vrrp preempt`** command enables preemption for a specified virtual router.

The command configures two delay periods:

- **minimum** time delays master VRRP takeover when VRRP is fully implemented.
- **reload** time delays master VRRP takeover after VRRP is initialized following a switch reload (boot).

Note: If the switch senses that there are no other active switches in the virtual router group, it will bypass any configured preempt time delay and become the VRRP master after the standard master downtime interval (3*advertisement interval + skew time).

Running-config maintains separate delay statements for **minimum** and **reload** parameters. Commands may list both parameters. Commands that list one parameter do not affect the omitted parameter. Values range from ***0 to 3600*** seconds (one hour). The default delay is zero seconds for both parameters.

The **`no vrrp preempt delay`** and **`default vrrp preempt delay`** commands reset the specified delay to the default of zero seconds. Commands that do no list either parameter resets both periods to zero. The **`no vrrp`** command also removes all **`vrrp preempt delay`** commands for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`vrrp` ***group*** **`preempt delay`** [**MINIMUM_INTERVAL**] [**RELOAD_INTERVAL**]

`no vrrp` ***group*** **`preempt delay`** [**DELAY_TYPE**]

`default vrrp` ***group*** **`preempt delay`** [**DELAY_TYPE**]

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1 to 255***.
- **MINIMUM_INTERVAL** period between preempt event and takeover of master VRRP router role.
  - ***no parameter*** minimum delay is not altered by command.
  - **minimum** ***min_time*** delay during normal operation (seconds). Values range from ***0 to 3600***.
- **RELOAD_INTERVAL** period after reboot-VRRP initialization and takeover of master VRRP router role.
  - ***no parameter*** reload delay is not altered by command.
  - **reload** ***reload_time*** delay after reboot (seconds). Values range from ***0 to 3600***.
- **DELAY_TYPE** delay type that is reset to default value.
  - ***no parameter*** reload and minimum delays are reset to default.

- **minimum** minimum delay is reset to default.
- **reload** reload delay is reset to default.

(**DELAY_TYPE** parameter is only used in **`no vrrp preempt delay`** and **`default vrrp preempt delay`** commands).

Related Command

vrrp preempt

Examples

- This command sets the minimum preempt time of ***90*** seconds for virtual router ***20*** on ***vlan 40***.

```
switch(config)# interface vlan 40
switch(config-if-vl40)# vrrp 20 preempt delay minimum 90
switch(config-if-vl40)#
```

- This command sets the minimum and reload preempt time to ***0*** for virtual router ***20*** on ***vlan 40***.

```
switch(config-if-vl40)# no vrrp 20 preempt delay
switch(config-if-vl40)#
```

13.5.4.21 vrrp priority-level

The **`vrrp priority-level`** command configures the switchs priority setting for a VRRP virtual router. Priority values range from ***1*** to ***254***. The default value is ***100***.

The router with the highest VRRP priority level setting for a group becomes the master virtual router for that group. The master virtual router controls the IP address and is responsible for forwarding traffic sent. The **`vrrp preempt`** command controls the time when a switch can become the master virtual router.

The **`no vrrp priority-level`** and **`default vrrp priority-level`** commands restore the default priority of 100 to the virtual router on the configuration mode interface by removing the corresponding **`vrrp priority-level`** command from ***running-config***. The **`no vrrp`** command also removes the **`vrrp priority-level`** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`vrrp` ***group*** **`priority-level`** ***level***

`no vrrp` ***group*** **`priority-level`** ***level***

`default vrrp` ***group*** **`priority-level`** ***level***

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1 to 255***.
- ***level*** priority setting for the specified virtual router. Values range from ***1 to 254***.

Example

This command sets the virtual router priority value of ***250*** for virtual router ***group 45*** on ***vlan 20***.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 45 priority-level 250
switch(config-if-vl20)#
```

13.5.4.22 vrrp session description

The **vrrp session description** command associates a text string to a VRRP virtual router on the configuration mode interface. The string has no functional impact on the virtual router. The maximum length of the string is ***80*** characters.

The **no vrrp session description** and **default vrrp session description** commands remove the text string association from the VRRP virtual router by deleting the corresponding **vrrp session description** command from ***running-config***. The **no vrrp** command also removes the **vrrp session description** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp ***group*** **session description** ***label_text***

no vrrp ***group*** **session description**

default vrrp ***group*** **session description**

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1 to 255***.
- ***label_text*** text that describes the virtual router. Maximum string length is ***80*** characters.

Example

This command associates the text string ***Laboratory Router*** to virtual router ***15*** on ***vlan 20***.

```
switch(config)# interface vlan 20
switch(config-if-vl20)# vrrp 15 session description Laboratory Router
switch(config-if-vl20)#
```

13.5.4.23 vrrp timers delay reload

The **vrrp timers delay reload** command delays the time for VRRP initialization after a system reboot.

The **no vrrp timers delay reload** and **default vrrp timers delay reload** commands restore the default value of ***0*** by deleting the **vrrp timers delay reload** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp ***group*** **timers delay reload** [**INTERVAL**]

no vrrp ***group*** **timers delay reload**

default vrrp ***group*** **timers delay reload**

Parameters

INTERVAL The number of seconds for the delay (seconds). Options include:

- ***no parameter*** Default value of ***0*** seconds.

- ***0 to 3600*** seconds. Ranges between ***0 to 60*** minutes.

Examples

- These commands configure the VRRP reload delay interval to ***15*** minutes.

```
switch(config)# interface vlan 100
switch(config-if-Vl100)# vrrp 2 timers delay reload 900
switch(config-if-Vl100)#
```

- These commands removes the VRRP reload delay interval.

```
switch(config)# interface vlan 100
switch(config-if-Vl100)# no vrrp 2 timers delay reload
switch(config-if-Vl100)#
```

13.5.4.24 vrrp tracked-object

The **`vrrp tracked-object`** command configures the VRRP client process on the configuration mode interface to track the specified tracked object and react when its status changes to **down**. The tracked object is created by the **`track`** command.

The **`no vrrp tracked-object`** and **`default vrrp tracked-object`** commands cause the VRRP client process to stop tracking the specified tracked object by removing the corresponding **`vrrp tracked-object`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`vrrp` ***group*** **`tracked-object`** ***object_name*** **ACTION** ***amount***

`no vrrp` ***group*** **`tracked-object`** ***object_name*** **ACTION**

`default vrrp` ***group*** **`tracked-object`** ***object_name*** **ACTION**

Parameters

- ***group*** Virtual Router IDentifier (VRID). Values range from ***1*** to ***255***.
- ***object_name*** name of tracked object.
- ***amount*** amount to decrement VRRP priority level. Values range from ***1*** to ***254***.
- **ACTION** The action that VRRP is to take when the tracked objects status changes to **down**. Options include:
  - **decrement** decrease VRRP priority level by *amount*.
  - **shutdown** shut down VRRP on the configuration mode interface.
  - If both **decrement** and **shutdown** are configured on the same interface for the same VRRP group, then VRRP will be shut down on the interface if the tracked object is down.

Related Commands

track

Example

This command causes ***interface ethernet 5*** to disable VRRP when tracked object ***eth8*** changes state.

```
switch(config-if-Et5)# vrrp 1 tracked-object eth8 shutdown
switch(config-if-Et5)#
```

13.6 DirectFlow

This section describes Arista's DirectFlow implementation. Topics in this section include:

- Introduction
- DirectFlow Configuration
- DirectFlow Feature Interactions
- DirectFlow Commands

13.6.1 Introduction

DirectFlow allows you to define flows consisting of conditions to match, and actions to perform, that are a superset of the OpenFlow 1.0 specification. DirectFlow runs alongside the existing L2/L3 forwarding plane, enabling a network architecture that incorporates new capabilities such as TAP aggregation and custom traffic engineering, alongside traditional forwarding models. DirectFlow does not require a controller or any third party integration, as flows can be installed via the CLI.

DirectFlow exposes the underlying forwarding ASIC's capabilities through a programmable interface like EAPI or the standard CLI.

DirectFlow works in conjunction with all other aspects of standard Layer 2 or Layer 3 bridging or forwarding, and DirectFlow traffic is subject to the standard packet processing pipeline within the ASIC. You can think of DirectFlow as a stage in packet processing that processes traffic after ingress checks and before any egress actions.

DirectFlow enables you to configure flows that consist of matching criteria and actions, and to modify how traffic is processed, by overriding the L2 lookup decision or rewriting a MAC address or VLAN for example.

Features like MAC learning, STP state checks, ingress or egress VLAN membership checks on ports, ACLs, QoS, and others are all respected by DirectFlow. Traffic that does not match any programmed flow is processed normally, while traffic that matches programmed flows is now subject to the actions specified in the flows.

13.6.1.1 DirectFlow Flows

You can define a relative priority between flows and define idle or hard timeouts for the flow. DirectFlow also enables you to insert a flow entry that matches on specified criteria, and define actions to be taken on traffic that matches the specified matching conditions. You can define flows to match on TCP flags, IPv6 source and destination addresses, input ports, and more.

For more information, see:

- DirectFlow Non-persistent Flows
- Supported Matches
- Supported Actions

13.6.1.1.1 DirectFlow Non-persistent Flows

DirectFlow enables you to configure flows that are not visible in the startup or running configurations and do not persist over a reboot. This feature is designed to be used for flows that are configured by a custom agent using the EOS SDK or eAPI and age out (expire) after a specified time period.

For example, if you are using a custom agent that reacts to traffic sent to the CPU (the redirect to CPU action), and you want to use a flow that will drop all matching traffic for ***5*** minutes, the agent can program a non-persistent flow that expires after a hard timeout of ***300*** seconds.

Using a non-persistent flow for this purpose ensures that other administrator actions (for example, saving the configuration) do not result in the flow being resurrected on startup or reverting to the saved configuration. It also removes the need for the agent to delete the expired flow.

Note: By default, all DirectFlow flows are persistent. You must use the **no** persistent command to configure a non-persistent flow.

13.6.1.1.2 Supported Matches

DirectFlow supports all matches on VLAN, ether type, source or destination MAC address, COS, source or destination IP address, IP protocol, IP TOS, L4 source, destination ports, ICMP type, and code.

In addition, DirectFlow also allows matching on:

- TCP flags
- IPv6 source address
- IPv6 destination address
- Traffic injected from the CPU
- Input port

DirectFlow also permits re-using the same flow on multiple input ports, saving valuable TCAM space.

13.6.1.1.3 Supported Actions

DirectFlow supports the following actions:

- Setting the source or destination MAC address
- VLAN
- COS
- IP TOS
- Transmit queue
- Output port list and mirroring traffic pre-modification (ingress mirror) and post-modification (egress mirror)
- Redirect to CPU

The redirect to CPU action is useful in cases in which a custom agent is running on EOS and you want to trap specific traffic (matching traffic) and send the trapped traffic to the agent.

13.6.2 DirectFlow Configuration

Consider the following when using DirectFlow.

- DirectFlow takes effect ONLY after exiting the individual flow configuration sub-mode.
- Match criteria are connected with Boolean AND operators. Therefore they must ***all*** match for the condition to be true and action to be taken.
- CLI is automatically set to match the ethertype to IP if IP fields (such as source or destination address or L4 ports) are chosen as part of other match/ action commands.
- In a single flow, only the following fields can be matched along with IPv4 or IPv6 source and destination addresses:
  - VLAN priority
  - VLAN ID
  - EtherType
  - Source interface
  - Class of Service (CoS)

13.6.2.1 Commands Used to Enable DirectFlow, Configure and Display Flows

A number of different commands are provided for the DirectFlow feature. The different commands enable you to enter the DirectFlow configuration mode, enable DirectFlow, configure flows, and display configured flows.

Note: ALL match criteria specified in a flow definition must match in the packet for the actions specified to be applied to the traffic.

Enter the DirectFlow Configuration Mode

The directflow command places the switch in DirectFlow configuration mode.

```
switch(config)# directflow
switch(config-directflow)#
```

Enable DirectFlow

The shutdown (DirectFlow) command determines if the configuration takes effect or not. To enable DirectFlow, enter the following command.

```
switch(config-directflow)# no shutdown
```

Create the Flow

The flow (DirectFlow) command creates a new flow entry. It must be unique or it will be overwritten by an existing entry.

```
switch(config-directflow)# flow Test-1
switch(config-directflow-Test-1)#
```

Create the DirectFlow Match Criteria

The match (DirectFlow-flow mode) command allows you to configure a rule or a flow which match on L2, L3, L4 fields of a packet and specify a certain action to either modify, drop or redirect the packet.

```
switch(config-directflow-Test-1)# match ethertype ip
switch(config-directflow-Test-1)# match source ip 10.10.10.10
```

Action Set

The action set (DirectFlow-flow mode) command allows you to configure a packet to be routed out a layer three interface using a DirectFlow entry.

```
switch(config-directflow-Test-1)# action egress mirror ethernet 7
switch(config-directflow-Test-1)# action set destination mac 0000.aaaa.bbbb
```

Finalize the Flow

DirectFlow flows do not take effect until you exit the configuration sub-mode for the specified flow. Use the **exit** command to finalize the flow and put it into effect.

```
switch(config-directflow-Test-1)# exit
switch(config-directflow)#
```

Redirect to CPU

The action output interface cpu (DirectFlow-flow mode) command allows you to configure flows so that traffic that matches the matching conditions specified in the flow is redirected to the CPU.

```
switch(config)# directflow
switch(config-directflow)# flow redirect-http-cpu
switch(config-directflow-redirect-http=cpu)# match ip protocol tcp
switch(config-directflow-redirect-http-cpu)# match destination port 80
switch(config-directflow-redirect-http-cpu)# action output interface cpu
```

Configuring a Non-persistent Flow

DirectFlow flows are persistent by default. Use the **no** persistent command to configure non-persistent flows.

```
switch config)# directflow
switch(config-directflow)# flow example-non-persistent
switch(config-directflow-example-non-persistent)# match input interface
 ethernet 25
switch(config-directflow-example-non-persistent)# action drop
switch(config-directflow-example-non-persistent)# no persistent
switch(config-directflow-example-non-persistent)# timeout hard 300
```

Display Details for Configured Flows

The **detail** option of the show directflow flows command enables you to display the details of configured flows. You can use this command to verify that a non-persistent flow is deleted after the timeout period configured for the flow has elapsed.

The following example shows the use of this command to view the configuration of a non-persistent flow before the timeout period has elapsed, and a second time, after the timeout period has expired.

The initial use of the command displays the flow configuration (before the timeout expires).

```
switch(config-directflow)# show directflow flows example-non-persistent detail
Flow example-non-persistent: (Flow programmed)
 persistent: False
 priority: 0
 hard timeout: 300
 idle timeout: 0
 match:
  ingress interface:
      Et25
 actions:
  drop
 matched: 0 packets, 0 bytes
```

The second use of the command displays the flow details (after the timeout expires). The output shows that the flow is no longer programmed.

```
switch(config-directflow)# show directflow flows example-non-persistent detail
Flow example-non-persistent: (Flow not programmed)
persistent: False
priority: 0
hard timeout: 300
idle timeout: 0
match:
  ingress interface:
      Et25
actions:
  drop
matched: 0 packets, 0 bytes
```

13.6.3 DirectFlow Feature Interactions

DirectFlow flow entries can have one of the following actions:

- A set of egress ports for sending a matched packet
- Copy to CPU
- Redirect to CPU
- Drop

- No specified action (in this case, the traffic is output normally).

The only exception is the ingress or egress mirroring action, where the DirectFlow entry causes the packet to be mirrored.

When the ingress or egress packets are mirrored, the original traffic is sent out normally.

Bridging Features

- DirectFlow entries have precedence over all entries in the MAC table, including static MAC entries and static MAC drop entries. Packets that do not match DirectFlow entries are forwarded based on the MAC address table.
- VLANs: DirectFlow entries can modify the VLAN of a packet. MAC learning takes place in the original VLAN for DirectFlow entries that modify the VLAN. The modified packet will be subject to VLAN membership checks on the egress port. If a packet has no VLAN tag, DirectFlow assumes it came in on the native VLAN for the ingress interface. A VLAN override causes the packet to obey the VLAN rules on the egress port.
- **Q-in-Q:** Q-in-Q is supported as DirectFlow entries match only on the outer tag.
- **Counters:** All packets that match DirectFlow entries cause interface counters to increment as usual.

Spanning Tree

DirectFlow runs alongside MSTP, RSTP, and PVST. DirectFlow entries do not match on packets that ingress an STP discarding port. DirectFlow entries that cause a packet to be forwarded out an STP discarding port will result in the packets being dropped on egress.

When STP is enabled, BPDUs will always be trapped to the CPU. When STP is disabled, BPDUs will be subject to DirectFlow entries and not be copied to the CPU by default.

LLDP, LAGs, and LACP

- LLDP packets are always trapped to the CPU. DirectFlow entries can never match LLDP packets.
- LAGs are fully supported, and can be part of a match criteria and part of an output action to an interface.
- LACP packets are always trapped to CPU. DirectFlow entries can never match LACP packets.

sFlow

sFlow is unaffected by DirectFlow.

IGMP Snooping

IGMP control packets are trapped to the CPU when IGMP Snooping is enabled. DirectFlow entries can match IGMP Snooping control traffic and override the trap to CPU.

Link-local-multicast packets are flooded in hardware in the VLAN via a TCAM entry. DirectFlow entries can match link-local-multicast packets and change the flooding behavior. As DirectFlow entries have to specify output interfaces or drop, the action will conflict and so matching DirectFlow entries will get precedence.

When IGMP snooping is enabled, unknown IPV4 multicast packets are flooded to the multicast-router ports in the VLAN. If DirectFlow entries match unknown IPV4 multicast packets, they will override the flooding behavior.

Data packets in groups under IGMP snooping control are sent to the group members through a MAC table entry. Matching DirectFlow entries override the MAC table entries.

ACLs

DirectFlow entries are lower priority than any configured Port ACLs (ingress). Packets coming in on a port that match DirectFlow entries obey any configured ACL on that port, and will only apply to packets that have a **permit** action.

DirectFlow entries are higher priority than any configured RACLs. Packets coming in on an L3 interface that match DirectFlow entries ignore any RACLs configured on that interface.

DirectFlow entries are lower priority than any configured Egress ACLs.

13.6.3.1 Layer Three Features and DirectFlow

DirectFlow runs alongside IP routing. If a packet is routed out a layer three interface using a DirectFlow entry, the actions associated with the entry will have to specify the new source MAC and destination MAC for the packet, as well as the physical port or LAG. If there are no output ports specified in an entry, packets that match that entry will be dropped.

Unicast Routing

When unicast routing is enabled, DirectFlow entries that match take precedence for all packets that would have been otherwise been routed. The three exceptions are the ingress mirror, egress mirror and copy-to-CPU actions where the packets will be routed normally in addition to the action being performed. Routed packets that do not match DirectFlow entries are forwarding based on the L3 lookup.

Multicast Routing

When multicast routing is enabled, DirectFlow entries that match take precedence for all packets that would have otherwise been multicast routed. The packets are not replicated based on the hardware multicast tables, but are forwarded strictly according to the actions specified by the DirectFlow entry. The entry can specify a set of output interfaces, which will result in the packet being replicated based on the DirectFlow entry.

13.6.3.2 Displaying DirectFlow Configurations

The show directflow flows command displays the contents of the flow table, showing each entry with its match rules, actions, and packet counters.

- This example shows the status of a default (persistent) flow.

```
switch(config-directflow)# show directflow flows
Flow Test1:
priority: 0
match:
ingress interface: Ethernet1
   ethertype ip
source ip address: 10.10.10.10
actions:
output mirror: Ethernet2
matched: 0 packets, 0 bytes
switch(config-directflow)#
```

- This example shows the status of a non-persistent flow. The flow will be deleted once ***5*** minutes have elapsed.

```
switch(config-directflow)# show directflow flows example-non-persistent
Flow example-non-persistent:
 persistent: False
 priority: 0
 hard timeout: 300
 idle timeout: 0
 match:
   ingress interface:
       Et25
 actions:
   drop
 matched: 0 packets, 0 bytes
```

13.6.4 DirectFlow Commands

DirectFlow Global Configuration Mode

- directflow

DirectFlow Configuration Commands

- action drop (DirectFlow-flow mode)
- action mirror (DirectFlow-flow mode)
- action output (DirectFlow-flow mode)
- action output interface cpu (DirectFlow-flow mode)
- action set (DirectFlow-flow mode)
- flow (DirectFlow)
- match (DirectFlow-flow mode)
- persistent
- priority (DirectFlow-flow mode)
- shutdown (DirectFlow)
- timeout (DirectFlow-flow mode)

DirectFlow and Clear Commands

- show directflow
- show directflow flows

13.6.4.1 action drop (DirectFlow-flow mode)

The **`action drop`** command configures packets that match an entry to be dropped.

The **`no action drop`** and **`default action drop`** commands remove the statement from the DirectFlow ***configuration mode***.

Command Mode

Directflow-flow Configuration

Command Syntax

`action drop`

`no action drop`

`default action drop`

Example

This command sets the action for packets from ***Test-1*** to be dropped.

```
switch(config-directflow-Test-1)# action drop
switch#
```

13.6.4.2 action mirror (DirectFlow-flow mode)

The **`action mirror`** command can be used to ingress or egress mirror traffic to a mirror destination. This requires a mirror destination to be setup on the switch. If a packet comes in or goes out an interface that is part of another mirror session, then the destination for that destination as well as the DirectFlow destination will receive a copy of the packet.

The **no action mirror** and **default action mirror** commands remove the statement from DirectFlow ***configuration mode.***

Command Mode

Directflow-flow Configuration

Command Syntax

action DIRECTION mirror INT_NAME

no action DIRECTION mirrorINT_NAME

default action DIRECTION mirror INT_NAME

Parameters

- **DIRECTION** transmission direction of traffic to be mirrored.
  - **ingress** mirrors before any rewrites.
  - **egress** mirrors after rewrites.
- ***INT_NAME*** Source interface for the mirroring session.
  - **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
  - **port-channel *p_range*** Port channel interfaces specified by ***p_range***.

Example

This command configures mirror traffic to ***ethernet 2***.

```
switch(config-directflow)# flow Test1
switch(config-directflow-Test1)# match ethertype ip
switch(config-directflow-Test1)# match source ip 10.10.10.10
switch(config-directflow-Test1)# action egress mirror ethernet 2
switch(config-directflow-Test1)#
```

13.6.4.3 action output (DirectFlow-flow mode)

The **action output** command configures an Ethernet or port channel interface as the output of a specified port mirroring session.

The **no action output** and **default action output**commands remove the statement from ***DirectFlow*** configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

action output DESTINATION

no action output DESTINATION

default action output DESTINATION

Parameters

DESTINATION transmission direction of traffic to be mirrored.

- **all** mirrors transmitted and received traffic.
- **flood** mirrors received traffic only.
- **interface ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
- **interface port-channel *p_range*** Port channel interfaces specified by ***p_range***.

- **nexthop vrf *vrf_name ip_addr***. If the next hop is reachable in the default VRF, the default VRF does not need to be specfied.

Examples

- This command configures ***interface ethernet 7*** as the output for the mirroring session.

```
switch(config-directflow-Test1)# action output interface ethernet 7
switch(config-directflow-Test1)#
```

- The following commands configure a flow redirecting all traffic from ***10.10.1.2*** to ***e_range*** next hop assuming an appropriately configured TCAM profile.

```
switch(config-directflow)# flow flow-sip-10_10_1_2-redirect-
to-10_30_1_2
switch(config-directflow-Test1)# match ethertype ip
switch(config-directflow-Test1)# match source ip 10.10.1.2
switch(config-directflow-Test1)# action output nexthop 10.30.1.2
switch(config-directflow-Test1)#
```

13.6.4.4 action output interface cpu (DirectFlow-flow mode)

The **`action output interface cpu`** command configures the action (other commands are used to define the traffic matching conditions).

The **`no action output interface cpu`** and **`default action output`** commands remove the statement from ***DirectFlow*** configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

`action output` **DESTINATION**

`no action output` **DESTINATION**

`default action output` **DESTINATION**

Parameters

DESTINATION transmission direction of traffic to be mirrored.

- **all** mirrors transmitted and received traffic.
- **flood** mirrors received traffic only.
- **interface cpu** Ethernet interfaces specified by ***e_range***.

Examples

- This command configures ***interface ethernet 7*** as the output for the mirroring session.

```
switch(config-directflow-Test1)# action output interface ethernet 7
switch(config-directflow-Test1)#
```

- These commands configure the action to redirect traffic matching the flow to the CPU and the matching conditions for the flow.

```
switch (config)# directflow
switch(config-directflow)# flow redirect-http-cpu
switch(config-directflow-redirect-http=cpu)# match ip protocol tcp
```

```
switch{config-directflow-redirect-http-cpu)# match destination p
 cpuort 80
switch(config-directflow-redirect-http-cpu)# action output interface
```

13.6.4.5 action set (DirectFlow-flow mode)

The **action set** command allows you to configure a packet to be routed out a layer three interface using a DirectFlow entry. The actions associated with the entry will have to specify the new source MAC and destination MAC for the packet, as well as the physical port or LAG. If there are no output ports specified in an entry, packets that match that entry will be dropped.

The **no action set** and **default action set** commands remove **action set** statement from ***DirectFlow*** configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

action set **CONDITION**

no action set **CONDITION**

default action set **CONDITION**

Parameters

CONDITION specifies parameter and value. Options include:

- **cos *0 to 7*** Cost of service.
- **destination mac *mac_addr*** Dotted hex notation.
- **ip tos *0 to 255*** Type of service.
- **source mac *mac_addr*** Dotted hex notation.
- **traffic-class *0 to 7*** Dotted hex notation.
- **vlan *0 to 4094*** Number of VLAN.

The **no action set** and **default action set** commands require only the CONDITION type without a specific condition value.

Example

These commands change the destination MAC of the frame.

```
switch(config-directflow)# flow Test1
switch(config-directflow-Test1)# action egress mirror ethernet 7
switch(config-directflow-Test1)# action set destination mac
 0000.aaaa.bbbb
```

13.6.4.6 directflow

The **directflow** command places the switch in DirectFlow configuration mode.

The **no directflow** and **default directflow** commands delete the DirectFlow configuration mode statements from ***running-config***.

DirectFlow configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. The **exit** command returns the switch to ***global configuration mode***.

Command Mode

Global Configuration

Command Syntax

`directflow`

`no directflow`

`default directflow`

Commands Available in DirectFlow-Flow configuration mode:

- flow (DirectFlow)
- shutdown (DirectFlow)

Examples

- This command places the switch in ***DirectFlow*** configuration mode.

```
switch(config)# directflow
switch(config-directflow)#
```

- This command returns the switch to ***global management*** mode.

```
switch(config-directflow)# exit
switch(config)#
```

13.6.4.7 flow (DirectFlow)

The **`flow`** command places the switch in ***flow*** configuration mode.

The **`flow`** command specifies the name of the flow that subsequent commands modify and creates a newflow definition if it references a nonexistent flow. All changes in a ***flow*** configuration mode edit session are pending until the session ends:

- The **`exit`** command saves pending changes to ***running-config*** and returns the switch to ***DirectFlow*** configuration mode. Changes are also saved by entering a different configuration mode.
- The **`abort`** command discards pending changes, returning the switch to ***DirectFlow*** configuration mode.

The **`no flow`** and **`default flow`** commands delete the specified role by removing the role and its statements from ***running-config***.

Command Mode

DirectFlow Configuration

Command Syntax

`flow` ***flow_name***

`no flow` ***flow_name***

`default flow` ***flow_name***

Parameters

flow_name Name of flow.

Commands Available in DirectFlow-Flow configuration mode:

- action drop (DirectFlow-flow mode)
- action mirror (DirectFlow-flow mode)
- action output (DirectFlow-flow mode)
- action set (DirectFlow-flow mode)

- match (DirectFlow-flow mode)

13.6.4.8 match (DirectFlow-flow mode)

The **match** command allows you to configure a rule or a flow which could match on L2, L3, L4 fields of a packet and specify a certain action to modify, drop or redirect the packet.

All traffic ingressing on the switch will be matched against the flows installed. In cases where none of the packets match, normal switching or routing behavior will take over. When multiple entries match a packet, precedence is given to the entry that was installed first.

The **no match** and **default match** commands remove the **match** statement from the configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

match CONDITION

no match CONDITION

default match CONDITION

Parameters

CONDITION specifies criteria for evaluating a route. Options include:

- **cos *0 to 7*** cost of service.
- **destination ip *ipv4_sub*** destination IPv4 subnet. L3 fields valid only if ethertype is IP (0x0800).
- **destination mac *mac_addr*** Add to the existing community. Dotted hex notation.
- **destination mac *mac_addr* mask *mac_mask*** Add to the sting community. Dotted hex notation.
- **destination port *0 to 65535*** Fields accepted only if protocol is TCP|UDP.
- **ethertype *0 to 65535*** Layer 4 destination port.
- **ethertype ARP** Layer 4 destination port.
- **ethertype IP** Layer 4 destination port.
- **icmp code *0 to 255*** Fields accepted only if protocol is ICMP.
- **icmp type *0 to 255*** Fields accepted only if protocol is ICMP.
- **input interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **input interface port-channel *p_num*** Port channel interface specified by ***p_num***.
- **ip protocol *0 to 255*** Type of service.
- **ip protocol icmp** L3 fields valid only if ethertype is IP (0x0800).
- **ip protocol tcp** L3 fields valid only if ethertype is IP (0x0800).
- **ip protocol udp** L3 fields valid only if ethertype is IP (0x0800).
- **ip tos *0 to 255*** L3 fields valid only if ethertype is IP (0x0800).
- **source ip *ipv4_subnet*** L3 fields valid only if ethertype is IP (0x0800).
- **source mac *mac_addr*** Add to the existing community. Dotted hex notation.
- **source mac *mac_addr* mask *mac_mask*** Add to the sting community. Dotted hex notation.
- **source port *0 to 65535*** Fields accepted only if protocol is TCP| UDP.
- **tcp flag ack** Layer 4 destination port.
- **tcp flag fin** Layer 4 destination port.
- **tcp flag psh** Layer 4 destination port.
- **tcp flag rst** Layer 4 destination port.
- **tcp flag syn** Layer 4 destination port.
- **tcp flag urg** Layer 4 destination port.
- **tcp flag urg** Layer 4 destination port.
- **vlan *1 to 4094* mask *1 to 4095*** Number of VLAN.

The **no match** and **default match** commands require only the **CONDITION** type without a specific condition value.

Example

This command creates the rules to match on Ethertype IP and Source IP ***10.10.10.10***.

```
switch(config-directflow)# flow Test1
switch(config-directflow-Test1)# persistent
switch(config-directflow-Test1)# match ethertype ip
switch(config-directflow-Test1)# match source ip 10.10.10.10
```

13.6.4.9 persistent

DirectFlow flows are persistent by default. Once finalized, they appear in the running configuration, and if saved to ***startup config*** they will persist over a reboot. The **no** form of the **persistent** command prevents the flow from showing up in ***running config***, ensuring that it will not persist over a reboot.

Command Mode

Directflow-flow Configuration

Command Syntax

persistent

no persistent

Example

These commands create and enable a non-persistent DirectFlow flow.

```
switch(config)# directflow
switch(config-directflow)# flow example-non-persistent
switch(config-directflow-example-non-persistent)# match input
 interface ethernet 25
switch(config-directflow-example-non-persistent)# action drop
switch(config-directflow-example-non-persistent)# no persistent
switch(config-directflow-example-non-persistent)# timeout hard 300
switch(config-directflow-example-non-persistent)# exit
switch(config-directflow)#
```

13.6.4.10 priority (DirectFlow-flow mode)

The **priority** command sets the priority for the flow match rules. Each flow-table entry has an optional priority field, with a higher number indicating a higher priority. Flows with the same priority may be loaded in any order, and the order may be changed at any time. If multiple entries match a packet, precedence is given to the entry that was installed first.

Priority numbers range from ***0 to 65535***. The default is ***0***. The higher priority rules match first.

The **no priority** and **default priority** commands remove **priority** statement from the ***DirectFlow*** configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

priority ***priority_value***

no priority

default priority

Parameter

priority_value priority ***xxx***. Value ranges from ***0 to 65535.*** Default is ***0***.

Example

These commands assign the priority of ***150*** to flow ***Test-1***.

```
switch(config-directflow-Test-1)# priority 150
switch(config-directflow-Test-1)#
```

13.6.4.11 show directflow

The **show directflow** command displays summary information for DirectFlow. With the **counters** or **details** options, it displays counters or details for all flows configured on the switch.

Command Mode

EXEC

Command Syntax

show directflow [counters | details]

Examples

- This command displays summary information for DirectFlow.

```
switch# show directflow
DirectFlow configuration: Enabled
Total matched: 0 packets
Total programmed flows: 3 flows switch#
```

- This command displays counters for all DirectFlow flows configured on the switch.

```
switch# show directflow counters
Flow Name      Source      Matched packets      Matched bytes
---------      ------      ---------------      -------------
test3          config      0                     0
test2          config      0                     0
test1          config      0                     0

Total matched packets: 0 switch>
```

- This command displays details for all DirectFlow flows configured on the switch.

```
switch# show directflow detail
Flow test3: (Flow programmed)
  persistent: True
  priority: 0
  priorityGroupType: default
  tableType: ifp
  hard timeout: 0
  idle timeout: 0
  match:
    Ethernet type: 0x86dd
```

```
    source IPv6 address: fcaa::/ffff:ffff:ffff:ffff:ffff:ffff:ff
ff:ffff
  actions:
    output interfaces:
        Et32
  source: config
  matched: 0 packets, 0 bytes
Flow test2: (Flow programmed)
  persistent: True
  priority: 0
  priorityGroupType: default
  tableType: ifp
  hard timeout: 0
  idle timeout: 0
  match:
    Ethernet type: IPv4
    source IPv4 address: 10.1.2.12/255.255.255.255
    IPv4 protocol: TCP
    destination TCP/UDP port or ICMP type: 8080
  actions:
    output interfaces:
        Et3/1
  source: config
  matched: 0 packets, 0 bytes
Flow test1: (Flow programmed)
  persistent: True
  priority: 0
  priorityGroupType: default
  tableType: ifp
  hard timeout: 0
  idle timeout: 0
  match:
    ingress interface:
        Et1/1
  actions:
    output interfaces:
        Et2/1
  source: config
  matched: 0 packets, 0 bytes
Flows: 3 programmed, 0 rejected switch#
```

13.6.4.12 show directflow flows

The **`show directflow flows`** command displays the contents of the flow table, showing each entry with its match rules, actions, and packet counters. Including the name of a specific flow limits the output to information about the specified flow.

Command Mode

EXEC

Command Syntax

`show directflow flows` [***flow_name*** [counters | detail]]

Parameters

- ***flow_name*** name of flow for which to display information. If no flow name is entered, command displays information for all flows.
- **counters** displays DirectFlow counters for the specified flow.
- **detail** displays detailed information for the specified flow.

Examples

- This command displays the contents of the flow table.

```
switch# show directflow flows
Flow test3:
  persistent: True
  priority: 0
  priorityGroupType: default
  tableType: ifp
  hard timeout: 0
  idle timeout: 0
  match:
    Ethernet type: 0x86dd
    source IPv6 address: fcaa::/ffff:ffff:ffff:ffff:ffff:ffff:ff
ff:ffff
  actions:
    output interfaces:
        Et32
  source: config
  matched: 0 packets, 0 bytes
Flow test2:
  persistent: True
  priority: 0
  priorityGroupType: default
  tableType: ifp
  hard timeout: 0
  idle timeout: 0
  match:
    Ethernet type: IPv4
    source IPv4 address: 10.1.2.12/255.255.255.255
    IPv4 protocol: TCP
    destination TCP/UDP port or ICMP type: 8080
  actions:
    output interfaces:
        Et3/1
  source: config
  matched: 0 packets, 0 bytes
Flow test1:
  persistent: True
  priority: 0
  priorityGroupType: default
  tableType: ifp
  hard timeout: 0
  idle timeout: 0
  match:
    ingress interface:
        Et1/1
  actions:
    output interfaces:
        Et2/1
  source: config
  matched: 0 packets, 0 bytes switch#
```

- This command displays information about flow ***test-1***.

```
switch# show directflow flows test-1
Flow test1:
  persistent: True
  priority: 0
  priorityGroupType: default
  tableType: ifp
  hard timeout: 0
```

```
  idle timeout: 0
  match:
    ingress interface:
        Et1/1
  actions:
    output interfaces:
        Et2/1
  source: config
  matched: 0 packets, 0 bytes switch#
```

- This command displays detailed information for flow ***test-1***.

```
switch# show directflow flows test-1 detail
switch>show directflow flows test1 detail
Flow test1: (Flow programmed)
  persistent: True
  priority: 0
  priorityGroupType: default
  tableType: ifp
  hard timeout: 0
  idle timeout: 0
  match:
    ingress interface:
        Et1/1
    source Ethernet address: 00:aa:aa:aa:aa:aa/ff:ff:ff:ff:ff:ff
    VLAN ID: 10
  actions:
    output interfaces:
  copy ingress to mirror dest interfaces: Ethernet1
    forward normally
  source: config
  matched: 0 packets, 0 bytes switch#
```

- This command displays detailed information for all flows regardless of their status as installed, rejected, configured or others.

```
switch# show directflow detail
Flow test-3: (Flow programmed)
  persistent: False
  priority: 0
  priorityGroupType: default
  hard timeout: 0
  idle timeout: 0
  match:
    ingress interface:
        Et11
  actions:
    copy ingress to mirror dest interfaces: Ethernet1
    forward normally
  source: config
  matched: 0 packets, 0 bytes
Flow test-1: (Flow programmed)
  persistent: True
  priority: 0
  priorityGroupType: default
  hard timeout: 0
  idle timeout: 0
  match:
    ingress interface:
        Et10
    source Ethernet address: 00:aa:aa:aa:aa:aa/ff:ff:ff:ff:ff:ff
    VLAN ID: 10
  actions:
```

```
    copy ingress to mirror dest interfaces: Ethernet1
    forward normally
  source: config
  matched: 0 packets, 0 bytes
Flow test-2: (Flow rejected due to invalid match criteria)
  persistent: True
  priority: 0
  priorityGroupType: default
  hard timeout: 0
  idle timeout: 0
  match:
    Ethernet type: IPv4
    IPv4 protocol: ICMP
    source TCP/UDP port or ICMP type: 3
    destination TCP/UDP port or ICMP type: 6
  actions:
    copy ingress to mirror dest interfaces: Ethernet1
    forward normally
  source: config
  matched: 0 packets, 0 bytes
Flows: 2 programmed, 1 rejected switch#
```

- This command displays counters for flow ***test-1***.

```
switch# show directflow flows test-1 counters
Flow Name     Source      Matched packets    Matched bytes
---------     ------      ---------------    -------------
test1         config                    0                0 switch#
```

- This command displays match counters per flow.

```
switch# show directflow counters
Flow Name     Source      Matched packets    Matched bytes
---------     ------      ---------------    -------------
test1         config                    0              146
Total matched packets: 1 switch#
```

13.6.4.13 shutdown (DirectFlow)

The **shutdown** command, in DirectFlow mode, disables DirectFlow on the switch. DirectFlow is disabled by default.

The **no shutdown** command re-enables DirectFlow.

Command Mode

Directflow Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- These commands enable DirectFlow on the switch.

```
switch(config)# directflow
switch(config-directflow)# no shutdown
```

```
switch(config-directflow)#
```

- This command disables DirectFlow Flow.

```
switch(config-directflow-Test1)# shutdown
```

13.6.4.14 timeout (DirectFlow-flow mode)

The **`timeout`** command, in ***DirectFlow*** mode, command configures the connection timeout period for connection sessions. The connection timeout period defines the interval between a users most recently entered command and an automatic connection shutdown. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

Command Mode

Directflow-flow Configuration

Command Syntax

`no timeout` ***hard***

`no timeout` ***idle***

Parameters

- ***idle*** session idle timeout length.
  - ***0*** Automatic connection timeout is disabled.
  - ***1-4294967295*** Automatic timeout period (seconds).
- ***hard*** session hard timeout length.
  - ***0*** Automatic connection timeout is disabled.
  - ***1-4294967295***

Example

- These commands enable a hard timeout period of ***5*** seconds on the switch.

```
switch(config)# directflow
switch(config-directflow-Test1)# timeout hard 5
switch(config-directflow-Test1)#
```

- These commands enable DirectFlow on the switch.

```
switch(config)# directflow
switch(config-directflow-Test1)# no timeout hard
switch(config-directflow-Test1)#
```

13.7 Decap Groups

These sections describe the Decap groups:

- Decap Groups Description
- Decap Groups Configuration
- Decap Commands

13.7.1 Decap Groups Description

The decap group is a data structure that receives encapsulated packets and extracts the payload. The switch then processes or forwards the extracted payload as required. Although packets cannot be transmitted through decap groups, nexthop groups can be used to create a packet's reverse path. Decap groups support payload extraction of packets received from Generic Routing Encapsulation (GRE) and IP-in-IP tunnels.

The decap capabilities are further enhanced to support the Generic UDP Encapsulation (GUE). GUE is a general method for encapsulating packets of random IP protocols within a UDP tunnel. GUE provides an extensible header format with optional data.

The switch identifies a GUE packet based on the outer UDP destination port. Then it terminates the tunnel based on the outer destination IP and removes the outer IP/UDP encapsulation. The outer UDP destination port is also used to determine if the inner payload is IP or MPLS. For IP payloads, the first nibble of the inner payload is used to distinguish between IPv4 vs IPv6. The decapsulated packet is then either IP or MPLS forwarded based on the inner payload.

Decap groups have these limitations:

- Tunnels are terminated using destination IP address; source IP address has no influence.
- Packets matching a decap group are not processed through their ingress interface and VLAN.
- During a tunnel termination, ingress ACL filter each decap group packet’s inner header.
- Packet counters are not available.
- VRF is not supported.

The decap support over GUE has the following limitaions:

- Outer UDP checksum validation is not supported.
- Outer IP options handling is not supported.
- Outer IP TTL check is not performed.
- No exception handling on inner IP packet. Specifically, packets with inner IP TTL=1 or IP options will simply be attempted to be forwarded in HW.
- VXLAN is not supported along with GUE decap option on DCS-7020, DCS-7280R, DCS-7280R2, DCS-7500R, and DCS-7500R2 switch series.
- When GUE is enabled with outer IP hashing, inner IP fields are not included in the load balance key of MPLS over GRE packets in tc-counters TCAM profile on DCS-7020, DCS-7280R, DCS-7280R2, DCS-7500R, and DCS-7500R2 switch series.

Decap groups are defined by their tunnel type and decap IP address:

- **Tunnel type** specifies the tunnel protocol that the switch uses to extract payload.
- **Decap IP address** specifies the IP address where the switch receives decap group packets.

Decap groups support Generic Routing Encapsulation (GRE) and IP-in-IP tunnels.

13.7.1.1 Generic UDP Encapsulation (GUE) Decap Configuration

The **`ip decap-group`** configuration is used to specify GUE tunnel termination as follows:

```
switch(config)# ip decap-group test
switch(config-dg-test)# tunnel type UDP
switch(config-dg-test)# tunnel decap-ip 2.2.2.2
switch(config-dg-test)# tunnel decap-interface et1
```

The global UDP destination port to payload type mapping is configured as follows:

```
switch(config)# ip decap-group type udp destination port 6080 payload ip
switch(config)# ip decap-group type udp destination port 5555 payload mpls
```

Note: Note that each payload type can be mapped to only one UDP port. Otherwise, the following error will be shown:

```
switch(config)# ip decap-group type udp destination port 6081 payload ip
% There can be only one UDP destination port per payload type
```

MPLSoverGUE Decap-group Configuration

The MPLSoGUE decap-group requires the following platform configurations. First, the **mpls-over-gre** must be enabled since this supports the mpls-over-gre support in hardware:

```
switch(config)# platform fap mpls-over-gre
```

Second, outer IP hashing must be enabled as follows:

```
(config)# load-balance policies
switch(config-load-balance-policies)# load-balance sand profile default
! profile default is a reserved profile
! profile default is the current global profile
switch(config-sand-load-balance-profile-default)# packet-type gue outer-ip
```

Note: The above platform configurations are not required on DCS-7280R3, DCS-7500R3, and DCS-7800R3.

13.7.2 Decap Groups Configuration

Decap groups are configured in ***decap-group*** configuration mode. The ***decap-group*** configuration mode is not a group change mode; the ***running-config*** is changed immediately upon entering commands. However, when exiting, the ***decap-group*** configuration mode does not affect running-config. The **exit** command returns the switch to ***global*** configuration mode.

- The static CLI entry for the incoming label is specified by the mpls static command.
- The tunnel type is specified by the tunnel type (Decap Group) command.
- The Decap IP address is specified by the tunnel decap-ip (Decap Group) command.
- The locally configured IP addresses are added to the Layer 3 interfaces using the tunnel decap-interface (Decap Group) command for a specified decap group.

Decap groups do not define a default destination address or tunnel type and is not functional until both parameters are configured. A decap group can contain multiple **tunnel decap-ip** statements.

Examples

- This command defines a static CLI entry for the incoming-label.

```
switch(config)# mpls static top-label 3400 ethernet 3/3/3 10.14.4.4
 pop payload-type ipv4
```

- This command creates a decap group named ***DC-1*** and configures the group to terminate packets from GRE tunnel packets with the destination IP address of ***10.14.3.2***.

```
switch(config)# ip decap-group DC-1
switch(config-dg-DC-1)# tunnel type gre
switch(config-dg-DC-1)# tunnel decap-ip 10.14.3.2
switch(config-dg-DC-1)# show active
 ip decap-group DC-1
   tunnel type gre
   tunnel decap-ip 10.14.3.2
switch(config-dg-DC-1)# end
```

```
switch(config)#
```

13.7.3 Decap Commands

Decap Configuration Commands

- ip decap-group
- tunnel decap-interface (Decap Group)
- tunnel decap-ip (Decap Group)
- tunnel type (Decap Group)

Decap Show Command

- show ip decap-group

13.7.3.1 ip decap-group

The **`ip decap-group`** command places the switch in ***decap-group*** configuration mode, through which decap groups are created or modified. A decap group is a data structure that defines a method of extracting the payload from an encapsulated packet that the switch receives on a specified IP address.

Decap groups do not specify a default IP address group or tunnel type. These parameters must be explicitly configured before a decap group can function.

The ***decap-group*** configuration mode is not a group change mode; the ***running-config*** is changed immediately upon entering commands. Exiting the ***decap-group*** configuration mode does not affect ***running-config***. The **`exit`** command returns the switch to ***global*** configuration mode.

The **`no ip decap-group`** and **`default ip decap-group`** commands delete previously configured commands in the specified **decap-group** mode.

Command Mode

Global Configuration

Command Syntax

`ip decap-group` ***group_name***

`no ip decap-group` ***group_name***

`default ip decap-group` ***group_name***

Parameters

group_name Decap group name.

Commands Available in Decap-group Configuration Mode

- tunnel decap-ip (Decap Group) Specifies the IP address of packets handled by the decap group.
- tunnel type (Decap Group) Specifies the tunnel protocol for extracting payload.
- show ip decap-group

Examples

- This command creates a decap group named ***DC-1***.

```
switch(config)# ip decap-group DC-1
switch(config-dg-DC-1)#
```

- This command exits the ***decap-group*** mode for the ***DC-1*** decap group.

```
switch(config-dg-DC-1)# exit
switch(config)#
```

- This command delete the decap group named ***DC-1***.

```
switch(config)# no ip decap-group DC-1
switch(config)#
```

13.7.3.2 show ip decap-group

The **`show ip decap-group`** command displays the IP decap groups that are available in the switch.

Command Mode

Global Configuration

Command Syntax

`show ip decap-group` [**decap-group name** | **dynamic**]

Parameters

- ***decap-group name*** The decap group name.
- **dynamic** Displays the dynamic entries only.

Related Commands

- show ip decap-group
- tunnel decap-ip (Decap Group)
- tunnel type (Decap Group)

Examples

- This command displays the IP decap groups that are available in a switch.

```
switch(config)# show ip decap-group
NOTE: "D" column indicates dynamic entries
D | Name                | Type       | Info            | Version | Addr Type
--|---------------------|------------|-----------------|---------|--------
* | d1                  | GRE        | 1.2.3.4         |         |
* | d2                  | IP-in-IP   | Ethernet12/3    | IPv4    | primary
  | gre-with-intf       | GRE        |                 |         |
  | ipip-with-decapall  | IP-in-IP   | all             | IPv4    | all
  | ipip-with-decapall  | IP-in-IP   | all             | IPv6    | all
  | ipip-with-intf      | IP-in-IP   | Ethernet11/3    | IPv6    | all
  | ipip-with-intf      | IP-in-IP   | Ethernet11/3    | IPv4    | primary
* | ipip-with-ip        | IP-in-IP   | 1001::1         | IPv6    |
* | ipip-with-ip        | IP-in-IP   | 1.1.1.1         | IPv4    |
  | p1                  | GRE        | 100.100.100.100 |         |
  | p2                  | UNKNOWN    |                 |         |
```

- This command displays the UDP decap groups that are available in a switch.

```
switch(config)# show ip decap-group
NOTE: "D" column indicates dynamic entries
D | Name    | Type  | Info       | Version | Address  | UDP Dest | Payload
  |         |       |            |         | Type     | Port     | Type
--|---------|-------|------------|---------|----------|----------|----------
  | foo     | UDP   | Ethernet1  | IPv4    |  primary | 6080     | ip
  |         |       |            |         |          | 5555     | mpls
  | bar     | UDP   | 2.2.2.2    | IPv4    |          | 6080     | ip
  |         |       |            |         |          | 5555     | mpls
```

- The following command displays the decap-groups configured in HW:

```
switch(config)# show platform fap decap-group
```

```
DecapIp                                      LIF
-------------------------------------------   ---------
2.2.2.2                                     |   0
3.3.3.3                                     |   1
```

13.7.3.3 tunnel decap-interface (Decap Group)

The **tunnel decap-interface** command adds all locally configured IP addresses to the specific Layer 3 interface per decap group.

The **no tunnel decap-interface** command and the **default tunnel decap-interface** command restores the default state by removing the locally added IP addresses from the decap group.

Command Mode

Decap-Group Configuration

Command Syntax

tunnel decap-interface{ Ethernet | Loopback | Management | Port-Channel | Tunnel | Vlan | Vxlan | all}

no tunnel decap-interface {Ethernet | Loopback | Management | Port-Channel | Tunnel | Vlan | Vxlan | all}

default tunnel decap-interface{Ethernet | Loopback | Management | Port-Channel | Tunnel | Vlan | Vxlan | all}

Parameters

- **Ethernet *e_num*** Ethernet interface specified by ***e_num***. The Ethernet port number ranges from ***1*** to ***36***.
- **Loopback *l_num*** Loopback interface specified by ***l_num***. The loopback interface number ranges from ***0*** to ***1000***.
- **Management *m_num*** Management interface specified by ***m_num***. The management port number ranges from ***1*** to ***1***.
- **Port-channel *p_num*** Port-channel interface specified by ***p_num***. Options include:
  - **Port-channel** interface number. The port-channel interface number ranges from ***1*** to ***2000***.
  - **Port-channel** sub interface number. The port-channel sub interface number ***1-2000***,***1-4094***.
- **tunnel *t_num*** Tunnel interface specified by ***t_num***. The tunnel interface number ranges from ***0*** to ***255***.
- **vlan *v_num*** VLAN interface specified by ***v_num***. The VLAN interface number ranges from ***1*** to ***4094***.
- **vxlan *vx_num*** VXLAN interface specified by ***vx_num***. The VXLAN tunnel interface number ranges from ***1*** to ***1***.
- **all *address-family*** This parameter configures all L3 interfaces as a decap interface.

Related Commands

- ip decap-group
- tunnel type (Decap Group)
- show ip decap-group

Example

These commands add locally configured IP addresses to the ***interface ethernet 1/1*** for the ***dg1*** decap group.

```
switch(config)# ip decap-group dg1
switch(config-dg-dg1)# tunnel decap-interface Ethernet1/1

switch(config-dg-dg1)# show active
ip decap-group dg1
   tunnel type ipip
```

```
    tunnel decap-interface Ethernet1/1
    tunnel decap-interface all address-family ipv6 address all
```

13.7.3.4 tunnel decap-ip (Decap Group)

The **`tunnel decap-ip`** command specifies the IP address of packets that are handled by the configuration mode decap group. A decap group is a data structure that defines a method of extracting the payload from an encapsulated packet that the switch receives on a specified IP address.

Decap groups do not define a default decap-ip address. A decap group is not functional until an IP address is specified. Decap groups can contain only one tunnel decap-ip statement; subsequent commands replace any previously configured statements.

Command Mode

Decap-Group Configuration

Command Syntax

`tunnel decap-ip` ***ipv4_address***

Parameters

ipv4_addr An IPv4 address.

Related Commands

- The ip decap-group command places the switch in ***decap-group*** configuration mode.
- The tunnel type (Decap Group) command specifies the tunnel protocol for extracting payload.
- The show ip decap-group command.

Guidelines

A decap group does not specify a default IP address group or tunnel type. These parameters must be explicitly configured before a decap group can function.

Example

These commands configure ***10.14.3.2*** as the decap-IP address for the ***DC-1*** decap group.

```
switch(config)# ip decap-group DC-1
switch(config-dg-DC-1)# tunnel decap-ip 10.14.3.2
switch(config-dg-DC-1)# show active
 ip decap-group DC-1
   tunnel decap-ip 10.14.3.2
switch(config-dg-DC-1)#
```

13.7.3.5 tunnel type (Decap Group)

The **`tunnel type`** command specifies the tunnel protocol for extracting payload from encapsulated packets that arrive on the IP address specified for the configuration mode decap group. Supported tunnel protocols include General Routing Encapsulation (GRE) and IP-in-IP.

Decap groups do not define a default tunnel type. A decap group is not functional until an IP address is specified. Decap groups can contain only one tunnel decap-ip statement; subsequent commands replace any previously configured statements.

Command Mode

Decap-group Configuration

Command Syntax

`tunnel type gre`

Related Commands

- The ip decap-group command places the switch in ***decap-group*** configuration mode.
- The tunnel decap-ip (Decap Group) command specifies the IP address of packets handled by the decap group.
- The show ip decap-group command.

Guidelines

A decap group does not specify a default IP address group or tunnel type. These parameters must be explicitly configured before a decap group can function.

Example

This command configures decap group ***DC-1*** to terminate packets from GRE tunnel packets.

```
switch(config)# ip decap-group DC-1
switch(config-dg-DC-1)# tunnel type gre
switch(config-dg-DC-1)# show active
 ip decap-group DC-1
   tunnel type gre
switch(config-dg-DC-1)#
```

13.8 Nexthop Groups

These sections describe the Nexthop groups:

- Nexthop Group Description
- Nexthop Group Configuration
- Support for IPv6 Link-Local Addresses in Nexthop Groups Entries
- Nexthop Group Commands

13.8.1 Nexthop Group Description

Each routing table entry provides the next hop address to its specified destination. A nexthop address is the address of the next device on the path to the entry's specified destination.

A nexthop group is a data structure that defines a list of nexthop addresses and a tunnel type for packets routed to the specified address. When an IP route statement specifies a nexthop group as the nexthop address, the switch configures a static route with a nexthop group member as the nexthop address and encapsulates packets forwarded to that address as required by the group's tunnel type.

The nexthop group size is a configurable parameter that specifies the number of entries that the group contains. Group entries that are not explicitly configured are filled with drop routes. The switch uses ECMP hashing to select the address within the nexthop group when forwarding packets. When a packet's hash selects a drop route, the packet is dropped.

Nexthop groups are supported on Trident platform switches and subject to the following restrictions:

- Each switch can support 512 IPv4 or IPv6 Tunnels
- Nexthop groups can contain 256 nexthops.
- The switch supports 1024 nexthop groups.
- Multiple routes can share a tunnel.

- Tunnels do not support IP multicast packets.

Nexthop groups support IP-in-IP tunnels. The entry IP address family within a particular nexthop group cannot be mixed, i.e. either they are all IPv4 or they are all IPv6 entries.

13.8.2 Nexthop Group Configuration

Nexthop groups are configured and modified in nexthop-group configuration mode. After a group is created, it is associated to a static route through an ip route nexthop-group statement.

These tasks are required to configure a nexthop group and apply it to a static route.

- Creating and Editing Nexthop Groups
- Configuring a Group's Encapsulation Parameters
- Configuring the Group's Size
- Creating Nexthop Group Entries
- Displaying Nexthop Groups
- Applying a Nexthop Group to a Static Route

Creating and Editing Nexthop Groups

Nexthop groups are created by a nexthop-group command that specifies a group that isn't already configured. The switch enters ***nexthop-group*** configuration mode for the new group. ***Nexthop-group*** mode is also accessible for modifying existing groups. When in ***nexthop-group*** configuration mode, the **`show active`** command displays the group's configuration.

- This command creates a nexthop group named ***NH-1***.

```
switch(config)# nexthop-group NH-1
switch(config-nexthop-group-NH-1)#
```

- These commands enter ***nexthop-group*** configuration mode for the group named ***NH3***, then displays the previously configured group parameters.

```
switch(config)# nexthop-group NH3
switch(config-nexthop-group-NH3)#show active
 nexthop-group NH3
   size 4
   ttl 10
   entry 0 tunnel-destination 10.14.21.3
   entry 1 tunnel-destination 10.14.21.5
   entry 2 tunnel-destination 10.14.22.5
   entry 3 tunnel-destination 10.14.22.6
switch(config-nexthop-group-NH3)#
```

Configuring a Group's Encapsulation Parameters

Packets in static routes that are associated with the nexthop group are encapsulated to support the group's tunnel type. Nexthop groups support IP-in-IP tunnels. The group also defines the source IP address and TTL field contents that are included in the packet encapsulation.

- This command configures the TTL setting to ***32*** for nexthop group ***NH-1*** encapsulation packets.

```
switch(config)# nexthop-group NH-1
```

```
switch(config-nexthop-group-NH-1)# ttl 32
switch(config-nexthop-group-NH-1)# show active
 nexthop-group NH-1
   size 128
   ttl 32
switch(config-nexthop-group-NH-1)#
```

The address is inserted in the encapsulation source IP fields is specified by tunnel-source (Nexthop Group).

- These commands create ***interface loopback 100***, assign an IP address to the interface, then specifies that address as the tunnel source for packets designated by nexthop-group ***NH-1***.

```
switch(config)# interface loopback 100
switch(config-if-Lo100)# ip address 10.1.1.1/32
switch(config-if-Lo100)# exit
switch(config)# nexthop-group NH-1
switch(config-nexthop-group-NH-1)# tunnel-source intf loopback 100
switch(config-nexthop-group-NH-1)# show active
 nexthop-group NH-1
   size 256
   ttl 32
   tunnel-source intf Loopback100
switch(config-nexthop-group-NH-1)#
```

Configuring IP-in-IP Encapsulation

Through IP-in-IP encapsulation, IP packets matching a static Nexthop-Group route are encapsulated within an IP-in-IP tunnel and forwarded.

This command configures a static Nexthop-Group route and an IP-in-IP Nexthop-Group for IP-in-IP encapsulation.

```
switch(config)# ip route 124.0.0.1/32 nexthop-group abc
switch(config)# nexthop-group abc type ip-in-ip
switch(config-nexthop-group-abc)# size 512
switch(config-nexthop-group-abc)# tunnel-source 1.1.1.1
switch(config-nexthop-group-abc)# entry 0 tunnel-destination 1.1.1.2
switch(config-nexthop-group-abc)# entry 1 tunnel-destination 10.1.1.1
switch(config-nexthop-group-abc)# ttl 64
switch(config-nexthop-group-abc)#
```

Configuring the Group's Size

The group's size specifies the number of entries in the group. A group can contain up to ***256*** entries, which is the default size. The group's size is specified by size (Nexthop Group).

This command configures the nexthop group ***NH-1*** to contain ***128*** entries.

```
switch(config)# nexthop-group NH-1
switch(config-nexthop-group-NH-1)# size 128
switch(config-nexthop-group-NH-1)# show active
 nexthop-group NH-1
   size 128
   ttl 64
```

```
switch(config-nexthop-group-NH-1)#
```

Creating Nexthop Group Entries

Each entry specifies a nexthop address that is used to forward packets. A nexthop group contains one entry statement for each nexthop address. The group's size specifies the number of entry statements the group may contain. Each entry statement is assigned an index number to distinguish it from other entries within the group; entry index numbers range from zero to the group size minus one.

Nexthop group entries are configured by entry (Nexthop Group).

- These commands set the nexthop group size at four entries, then create three entries. Packets that are hashed to the fourth entry are dropped.

```
switch(config)# nexthop-group NH-1
switch(config-nexthop-group-NH-1)# size 4
switch(config-nexthop-group-NH-1)# entry 0 tunnel-destination
 10.13.4.4
switch(config-nexthop-group-NH-1)# entry 1 tunnel-destination
 10.15.4.22
switch(config-nexthop-group-NH-1)# entry 2 tunnel-destination
 10.15.5.37
switch(config-nexthop-group-NH-1)# show active
 nexthop-group NH-1
   size 4
   ttl 64
   entry 0 tunnel-destination 10.13.4.4
   entry 1 tunnel-destination 10.15.4.22
   entry 2 tunnel-destination 10.15.5.37
switch(config-nexthop-group-NH-1)#
```

- These commands configure a nexthop group with three IPv6 nexthop entries.

```
switch(config)# nexthop-group nhg-v6-mpls type ip
switch(config-nhg-v6-mpls)# size 3
switch(config-nhg-v6-mpls)# entry 0 nexthop 2002::6401:1
switch(config-nhg-v6-mpls)# entry 1 nexthop 2002::6404:1
switch(config-nhg-v6-mpls)# entry 2 nexthop 2002::6404:2
switch(config-nhg-v6-mpls)#
```

- These commands configure an IPv4 route to point to the nexthop group **nhg-v6-mpls**. (Both IPv4 routes and IPv6 routes can point to this nexthop group.)

```
switch# ip route 100.5.0.0/16 Nexthop-Group nhg-v6-mplsp
switch#
```

Displaying Nexthop Groups

The show nexthop-group command displays a group's configured parameters.

This command displays the properties of the nexthop group named ***NH-1***.

```
switch> show nexthop-group NH-1
Name             Id       type     size   ttl    sourceIp
```

```
NH-1              4        ipInIp   256    64     0.0.0.0 switch>
```

Applying a Nexthop Group to a Static Route

The ip route nexthop-group associates a nexthop group with a specified destination address and configures the encapsulation method for packets tunneled to that address.

This command creates a static route in the default VRF, using the nexthop group of ***NH-1*** to determine the next hop address.

```
switch(config)# ip route 10.17.252.0/24 nexthop-group NH-1
switch(config)#
```

The **`show ip route`** command displays the routing table for a specified VRF. Routes that utilize a nexthop group entry are noted with a route type code of ***NG***.

This command displays a routing table that contains a static route with its nexthop specified by a nexthop group.

```
switch> show ip route
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I - ISIS, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route

Gateway of last resort is not set

 C      10.3.3.1/32 is directly connected, Loopback0
 C      10.9.1.0/24 is directly connected, Ethernet51/3
 C      10.10.10.0/24 is directly connected, Ethernet51/1
 S      10.20.0.0/16 [20/0] via 10.10.10.13, Ethernet51/1
 C      10.10.11.0/24 is directly connected, Ethernet3
 NG     10.10.3.0/24 [1/0] via ng-test1, 5
 C      10.17.0.0/20 is directly connected, Management1
 S      10.17.0.0/16 [1/0] via 10.17.0.1, Management1
 S      10.18.0.0/16 [1/0] via 10.17.0.1, Management1
 S      10.19.0.0/16 [1/0] via 10.17.0.1, Management1
 S      10.20.0.0/16 [1/0] via 10.17.0.1, Management1
 S      10.22.0.0/16 [1/0] via 10.17.0.1, Management1 switch>
```

13.8.3 Support for IPv6 Link-Local Addresses in Nexthop Groups Entries

IPv6 Link-local addresses in Nexthop Groups entries support IPv6 link-local nexthops belonging to a Nexthop Group. Only the MPLS Nexthop Group supports IPv6 and because of this, IPv6 is limited to getting support only by the Nexthop Group of MPLS. An advantage is that you can use these devices even when they are not configured with globally routable IPv4 or IPv6 addresses.

13.8.3.1 Configuration

An MPLS next-hop group with IPv6 address now accepts an interface if the IPv6 address is a link-local. Note the use of percentages between the IPv6 address and the interface.

```
switch(config)# nexthop-group nhg1 type mpls
switch(config-nexthop-group-nhg1)# entry 0 push label-stack 606789 nexthop
 fe80::fe80:2%Ethernet2
switch(config-nexthop-group-nhg1)# entry 1 push label-stack 204164 nexthop
 fe80::fe80:2%Ethernet3
```

13.8.3.2 Show Commands

Use the **`show nexthop-group`** command to display the current status of the nexthop-groups.

```
switch# show nexthop-group
nhg1
  Id                1
  Type              mpls
  Size              12
  Entries (left most label is the top of the stack)
    0  push label-stack 606789   nexthop fe80::fe80:2
         Tunnel destination directly connected, Ethernet2
         00:d4:27:77:e9:77, Ethernet2
    1  push label-stack 204164   nexthop fe80::fe80:2
         Tunnel destination directly connected, Ethernet3
         00:79:21:32:0f:32, Ethernet3
```

13.8.3.3 Limitations

Review the following limitations for the support of IPv6 link-local address in nexthop group entries:

- Only the nexthop-group of MPLS supports an IPv6 address. Therefore, link-local IPv6 addresses are only supported for this type of nexthop-group.
- Nexthop-groups are configured and exist in the default VRF. The link-local IPv6 addresses for nexthop-group entries can only be resolved for interfaces in the default VRF.

13.8.4 Nexthop Group Commands

Nexthop Commands

- entry (Nexthop Group)
- ip route nexthop-group
- nexthop-group
- size (Nexthop Group)
- ttl (Nexthop Group)
- tunnel-source (Nexthop Group)

Nexthop Show Command

- show nexthop-group

13.8.4.1 entry (Nexthop Group)

The **`entry`** command defines a nexthop entry in the ***nexthop group*** configuration mode . Each nexthop entry specifies a nexthop IP address for static routes to which the nexthop group is assigned. The group size (size (Nexthop Group)) specifies the quantity of entries a group contains. Each entry is created by an individual command. Entries within a group are distinguished by an index number.

The **no entry** and **default entry** commands delete the specified nexthop group entry, as referenced by index number, by removing the corresponding **entry** statement from ***running-config.***

Command Mode

Nexthop-group Configuration

Command Syntax

entry ***index*** tunnel-destination ***ipv4_address***

no entry ***index***

default entry ***index***

Parameters

- ***index*** Entry index. Values range from **0** to **group-size – 1**.
- ***ipv4_address*** Nexthop IPv4 address.

group-size is the group's entry capacity, as specified by the size (Nexthop Group) command.

Example

These commands sets the nexthop group size at four entries, then creates three nexthop entries. Packets that are hashed to the fourth entry are dropped.

```
switch(config)# nexthop-group NH-1
switch(config-nexthop-group-NH-1)# size 4
switch(config-nexthop-group-NH-1)# entry 0 tunnel-destination
 10.13.4.4
switch(config-nexthop-group-NH-1)# entry 1 tunnel-destination
 10.15.4.22
switch(config-nexthop-group-NH-1)# entry 2 tunnel-destination
 10.15.5.37
switch(config-nexthop-group-NH-1)# show active
 nexthop-group NH-1
   size 4
   ttl 64
   entry 0 tunnel-destination 10.13.4.4
   entry 1 tunnel-destination 10.15.4.22
   entry 2 tunnel-destination 10.15.5.37
switch(config-nexthop-group-NH-1)#
```

13.8.4.2 ip route nexthop-group

The **ip route nexthop-group** command creates a static route. The destination is a network segment. The nexthop address is one of the IP addresses that comprise the specified nexthop group. Packets forwarded as a result of this command are encapsulated as specified by the tunnel-type parameter of the specified nexthop group.

When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence. When a route created through this command has the same administrative distance as another static route (ECMP), the route that was created earliest has preference; ***running-config*** stores static routes in the order that they are created.

By default, the administrative distance assigned to static routes is ***1***. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of ***200*** is overridden by OSPF intra-area routes, which have a default distance of ***110***.

The **no ip route nexthop-group** and **default ip route nexthop-group** commands delete the specified route by removing the corresponding **ip route nexthop-group** command from ***running-config***. **Ip route nexthop-group** statements for an IP address in multiple VRFs must be removed separately.

A **no ip route** or **default ip route** command without a nexthop parameter deletes all corresponding **ip route nexthop-group** statements. Deleting a user-defined VRF also deletes its static routes.

Command Mode

Global Configuration

Command Syntax

ip route [**VRF_INST** ***dest_net*** **nexthop-group** ***nhgp_name*** [***dist***][**TAG_OPTION**][**RT_NAME**]

no ip route [**VRF_INST**] ***dest_net*** [**nexthop-group** ***nhgroup_name***][***distance***]

default ip route [**VRF_INST**] ***dest_net*** [**nexthop-group** ***nhgroup_name***][***distance***]

Parameters

- **VRF_INST** Specifies the VRF instance being modified.
  - ***no parameter*** Changes are made to the default VRF.
  - **vrf** ***vrf_name*** Changes are made to the specified VRF.
- ***dest_net*** Destination IPv4 subnet (CIDR or address-mask notation).
- ***nhgp_name*** Name of nexthop group.
- ***dist*** Administrative distance assigned to route. Options include:
  - ***no parameter*** Route assigned default administrative distance of one.
  - **1-255** The administrative distance assigned to route.
- **TAG_OPTION** Static route tag. Options include:
  - ***no parameter*** Assigns default static route tag of ***0***.
  - **tag** ***t_value*** Static route tag value. ***t_value*** ranges from **0** to **4294967295**.
- **RT_NAME** Associates descriptive text to the route. Options include:
  - ***no parameter*** No text is associated with the route.
  - **name** ***descriptive_text*** The specified text is assigned to the route.

Related Commands

The **ip route** command creates a static route that specifies the nexthop address without using nexthop groups.

Example

This command creates a static route in the default VRF, using the nexthop group of ***NH-1*** to determine the next hop address.

```
switch(config)# ip route 10.17.252.0/24 nexthop-group NH-1
switch(config)#
```

13.8.4.3 nexthop-group

The **nexthop-group** command places the switch in ***nexthop-group*** configuration mode, through which nexthop groups are created or modified. The command also specifies the tunnel protocol for extracting payload from encapsulated packets that arrive through an IP address upon which the group is applied.

A nexthop group is a data structure that defines a list of nexthop addresses and the encapsulation process for packets routed to the specified address. The command either accesses an existing ***nexthop group***

configuration or creates a new group if it specifies a non-existent group. Supported tunnel protocols include IP ECMP and IP-in-IP.

The ***nexthop-group*** configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***nexthop-group*** configuration mode does not affect ***running-config***. The **exit** command returns the switch to ***global*** configuration mode.

The **no nexthop-group** and **default nexthop-group**commands delete previously configured commands in the specified **nexthop-group** mode. When the command does not specify a group, it removes all nexthop-groups. When the command specifies a tunnel type without naming a group, it removes all nexthop-groups of the specified type.

Command Mode

Global Configuration

Command Syntax

nexthop-group ***group_name*** type **TUNNEL_TYPE**

no nexthop-group [***group_name***][type **TUNNEL_TYPE**]

default nexthop-group [***group_name***][type**TUNNEL_TYPE**]

Parameters

- ***group_name*** Nexthop group name.
- **TUNNEL_TYPE** Tunnel protocol of the nexthop-group. Options include:
  - **ip** ECMP nexthop.
  - **ip-in-ip** IP in IP tunnel.
  - **gre** Encapsules the Layer 3 protocols overs IP networks.
  - **mpls-over-gre** Tunnels MPLS over a non-MPLS network.
    - **entry** Nexthop Group Entry Configuration.
    - **size** Nexthop Group Entry Size.
    - **tos** Tunnel encapsulation IP type of service.
    - **ttl** Tunnel encapsulation TTL value.
    - **tunnel-source** Source Interface or Address.

Commands Available in Nexthop-group Configuration Mode

- entry (Nexthop Group)
- size (Nexthop Group)
- ttl (Nexthop Group)
- tunnel-source (Nexthop Group)

Restrictions

Tunnel type availability varies by switch platform.

Examples

- This command creates a nexthop group named ***NH-1*** that specifies ECMP nexthops.

```
switch(config)# nexthop-group NH-1 type ip
switch(config-nexthop-group-NH-1)#
```

- This command exits nexthop-group mode for the ***NH-1*** nexthop group.

```
switch(config-nexthop-group-NH-1)# exit
switch(config)#
```

- These commands creates a nexthop group ***NH-2*** of type MPLS over GRE.

```
switch(config)# nexthop-group NH-2 type mpls-over-gre
switch(config-nexthop-group-NH-2)# tunnel-source 11.1.1.1
switch(config-nexthop-group-NH-2)# ttl 32
switch(config-nexthop-group-NH-2)# tos 20
switch(config-nexthop-group-NH-2)# entry 0 push label-stack 16000 tunnel-
destination 11.1.1.2
switch(config)# ip route 100.1.1.1/32 Nexthop-Group NH-2

Counters for nexthop group may be enabled using the following command
switch(config)# hardware counter feature nexthop
```

13.8.4.4 show nexthop-group

The **`show nexthop-group`** command displays properties of the specified nexthop group.

Command Mode

EXEC

Command Syntax

`show nexthop-group` ***nhgroup_name*** [**VRF_INST**]

Parameters

- ***nhgroup_name*** Name of the group displayed by command.
- **VRF_INST** Specifies the VRF instance for which data is displayed.
  - ***no parameter*** Context-active VRF.
  - **vrf *vrf_name*** Specifies the name of VRF instance. System default VRF is specified by **default**.

Related Commands

The show nexthop-group command places the switch in the ***nexthop-group*** configuration mode to create a new group or modify an existing group.

Example

This command displays the nexthop group information.

```
switch(config)# show nexthop-group
  Id          107
  Type        mplsOverGre
  Size        1  (auto size enabled, programmed size 1)
  TTL         32
  Source IP   11.1.1.1
  Entries (left most label is the top of the stack)
    0  push label-stack 16000   tunnel-destination 11.1.1.2
         Tunnel destination directly connected, Ethernet1
         00:00:aa:aa:aa:aa, Ethernet1

With nexthop group counter enabled
switch(config)# show nexthop-group
  Id          1
  Type        mplsOverGre
  Size        1  (auto size enabled, programmed size 1)
  TTL         64
  Source IP   0.0.0.0
  Entries (left most label is the top of the stack)
    0  push label-stack 16000   tunnel-destination 1.1.1.2
         Tunnel destination directly connected, Ethernet1
         00:00:aa:aa:aa:aa, Ethernet1
```

```
         0 packets, 0 bytes

switch(config)#show nexthop-group summary
Number of Nexthop Groups configured: 1
Number of unprogrammed Nexthop Groups: 0

  Nexthop Group Type   Configured
-------------------- ------------
       MPLS over GRE            1

  Nexthop Group Size   Configured
-------------------- ------------
                   1            1
```

13.8.4.5 size (Nexthop Group)

The **size** command configures the quantity of nexthop entries in the ***nexthop group*** configuration mode . Each entry specifies a nexthop IP address for static routes to which the group is assigned. Entries are configured with the **entry (Nexthop Group)** command. The default size is ***256*** entries.

The **no size** and **default size** commands restore the size of the configuration mode nexthop group to its default of ***256*** by removing the corresponding **size** command from ***running-config***.

Command Mode

Nexthop-group Configuration

Command Syntax

size ***entry_size***

no size ***entry_size***

default size ***entry_size***

Parameter

entry_size Group size (entries). Value ranges from **1** to **255**. Default value is **256**.

Example

This command configures the nexthop group ***NH-1*** to contain ***128*** entries.

```
switch(config)# nexthop-group NH-1
switch(config-nexthop-group-NH-1)# size 128
switch(config-nexthop-group-NH-1)# show active
 nexthop-group NH-1
   size 128
   ttl 64
switch(config-nexthop-group-NH-1)#
```

13.8.4.6 ttl (Nexthop Group)

The **ttl** command specifies the number entered into the TTL (time to live) encapsulation field of packets that are transmitted to the address designated by the configuration mode nexthop group. The default TTL value is ***64***.

The **no ttl** and **default ttl** commands restore the default TTL value written into TTL fields for the ***nexthop group*** configuration mode by deleting the corresponding **ttl** command from ***running-config***.

Command Mode

Nexthop-group Configuration

Command Syntax

`ttl` ***hop_expiry***

`no ttl` ***hop_expiry***

`default ttl` ***hop_expiry***

Parameters

hop_expiry Period that the packet remains valid (seconds or hops) Value ranges from **`1`** to **`64`**.

Restrictions

This command is available only to Nexthop groups for tunnels of type **IP-in-IP**, **GRE**, **MPLS**, and **MPLS over GRE**.

Related Commands

The nexthop-group command places the switch in the ***nexthop-group*** configuration mode.

Examples

- This command configures the **`ttl`** setting to ***32*** for nexthop group ***NH-1*** packets.

```
switch(config)# nexthop-group NH-1
switch(config-nexthop-group-NH-1)# ttl 32
switch(config-nexthop-group-NH-1)# show active
 nexthop-group NH-1
   size 128
   ttl 32
switch(config-nexthop-group-NH-1)#
```

- This command restores the **`no ttl`** setting for nexthop group ***NH-1*** packets.

```
switch(config-nexthop-group-NH-1)# no ttl
switch(config-nexthop-group-NH-1)# show active nexthop-group NH-1
   size 128
   ttl 64
switch(config-nexthop-group-NH-1)#
```

13.8.4.7 tunnel-source (Nexthop Group)

The **`tunnel-source`** command specifies the address that is entered into the source IP address encapsulation field of packets that are transmitted as designated by the ***nexthop group*** configuration mode . The command may directly specify an IP address or specify an interface from which an IP address is derived. The default source address IP address is ***0.0.0.0***.

The **`no tunnel-source`** and **`default tunnel-source`** commands remove the source IP address setting from the configuration mode nexthop group by deleting the **`tunnel-source`** command from ***running-config***.

Command Mode

Nexthop-group Configuration

Command Syntax

`tunnel-source` **SOURCE**

`no tunnel-source` **SOURCE**

`default tunnel-source` **SOURCE**

Parameters

SOURCE IP address or derivation interface. Options include:

- ***ipv4_addr*** An IPv4 address.
- **intf ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **intf loopback *l_num*** Loopback interface specified by ***l_num***.
- **intf management *m_num*** Management interface specified by ***m_num***.
- **intf port-channel *p_num*** Port-channel interface specified by ***p_num***.
- **intf vlan *v_num*** VLAN interface specified by ***v_num***.

Restrictions

This command is available only to Nexthop groups for tunnels of type **ip-in-ip**.

Related Commands

The nexthop-group command places the switch in the ***nexthop-group*** configuration mode.

Example

These commands create **`interface loopback 100`**, assign an IP address to the interface, then specifies that address as the tunnel source for packets designated by nexthop-group ***NH-1***.

```
switch(config)# interface loopback 100
switch(config-if-Lo100)# ip address 10.1.1.1/32
switch(config-if-Lo100)# exit

switch(config)# nexthop-group NH-1
switch(config-nexthop-group-NH-1)# tunnel-source intf loopback 100
switch(config-nexthop-group-NH-1)# show active nexthop-group NH-1
   size 256
   ttl 64
   tunnel-source intf Loopback100
switch(config-nexthop-group-NH-1)# show nexthop-group NH-1
Name             Id       type     size   ttl    sourceIp
NH-1             2        ipInIp   256    64     10.1.1.1

switch(config-nexthop-group-NH-1)#
```

13.9 Global Knob to Set MTU for all Layer 3 Interfaces

- Global Layer 3 Maximum Transfer Unit (MTU) feature provides a CLI command to set the MTU value for all Layer 3 interfaces.
- The default value for global Layer 3 MTU and for each interface Layer 3 MTU is ***1500***.
- Any interface which has an MTU configured is not affected by the global L3 MTU.
- When the global L3 MTU is changed, any existing interfaces which are using the default are updated to use the new MTU. Any additional interfaces which are configured in the future will also use the new default MTU.

The related sections are:

- Global Knob to Set MTU for all Layer 3 Interfaces Configuration
- Show Commands
- Limitations

13.9.1 Global Knob to Set MTU for all Layer 3 Interfaces Configuration

The global L3 **mtu** command is under ***interface-defaults*** sub-mode. In this example, the MTU value is set to ***1600***.

```
(config)# interface defaults
(config-interface-defaults)# mtu 1600
```

13.9.2 Limitations

The following are limitations for the Global knob to set MTU for all Layer 3 interfaces feature:

- Only Ethernet, Port-Channel, and VLAN interfaces are supported.
- Other interface types, such as Management, Loopback, VXLAN, and Tunnel, are not supported.

13.9.3 Show Commands

Use the **show interface** command to display the current MTU value applied on the selected interface. In the following example, ***interface ethernet1***, ***interface ethernet2***, and ***interface ethernet3*** are selected for display.

```
(config)# interface ethernet3
(config-if-Et3)# no switchport
(config-if-Et3)# mtu 1600
(config-if-Et3)# interface defaults
(config-interface-defaults)# mtu 1700
(config-interface-defaults)# show interfaces ethernet1
  ...
  IP MTU 1700 bytes (default) , BW 10000000 kbit
  ...
(config-interface-defaults)# show interfaces ethernet2
  ...
  IP MTU 1700 bytes (default) , BW 10000000 kbit
  ...
(config-interface-defaults)# show interfaces ethernet3
  ...
  IP MTU 1600 bytes , BW 10000000 kbit
  ...
```

In this example,

- Switch ***interface ethernet3*** from Layer 2 to Layer 3 and explicitly set its MTU to ***1600***.
- Enter the ***interface-defaults sub-mode***, set global L3 MTU to ***1700***.
- The **show interface** command states ***interface ethernet1*** and ***interface ethernet2*** are using global L3 MTU value, while ***interface ethernet3*** are using its local MTU value.
- The **default** in the output indicates whether or not the interface is using global L3 MTU.

13.10 Support for Layer 3 MTU on 7280R3/7500R3/7800R3

Support for Layer 3 Maximum Transmission Unit (MTU) on 7280R3/7500R3/7800R3 switches includes the following:

- Enforces the MTU for Layer 3 packets on 7280R3/7500R3/7800R3 switches. The MTU can be set on any SVI and the MTU of that specific SVI is enforced when the packets egress out of a trunk port. This behavior is not supported on 7280E/R/R2 and 7500E/R/R2 line cards.
- MTU enforcement is done after adding all of the encapsulation headers in the Egress packet. For example, if an interface is configured with an MTU value of ***1500*** and the packet size is larger than ***1500*** with all encapsulation headers added, the packet is considered MTU violated. In 7280E/R/R2 and 7500E/R/R2 line cards, MTU is enforced before adding encapsulation headers.
- Actual MTU configured in hardware is MTU configured in the CLI + ***18*** bytes. ***18*** bytes is added to accommodate the L2 header. If the outgoing packet is untagged, the actual MTU configured in hardware is the MTU configured in the CLI + four (***4***) bytes.

13.10.1 Layer 3 MTU Configuration

You can configure Layer 3 MTU on Layer 3 interfaces and SVI's using the **`mtu`** command within interface configuration mode.

This command sets the MTU size of ***1492*** bytes on ***interface ethernet 1/1***.

```
(config)# interface ethernet 1/1
(config-if-Et1/1)# mtu 1492
```

MTU is independently configurable on all routable interfaces. The switch supports MTU sizes ranging from ***68*** to ***9214*** bytes. The default MTU size is ***1500*** bytes.

The **`no mtu`** and **`default mtu`** commands restore the interface's MTU to the default value by removing the corresponding **`mtu`** command from ***running-config***.

13.10.2 Layer 3 MTU Show Commands

The **`show interfaces`** command displays the MTU configured on the L3 interface. In this example, ***ethernet 1/1*** interface is selected to display. **`IP MTU`** represents the Layer 3 MTU of the interface.

```
switch(config-if-Et9/1)# show interfaces ethernet 1/1
...
  IP MTU 1600 bytes , BW 100000000 kbit
  Full-duplex, 100Gb/s over 4 lanes, auto negotiation: off, uni-link:
 n/a
...
```

13.10.3 L3 MTU Limitations

The following limitations apply to support of Layer 3 MTU:

- Only three (3) unique MTU values can be configured on a switch. If the number of unique MTU values exceeds three (3), additional unique MTU definitions will not be enforced on an interface.
    - When the number of unique MTU values becomes less than three (3) (after unconfiguring MTU on some interfaces), MTU on interfaces that are previously unprogrammed, are not reprogrammed automatically. CLI commands **`no mtu`** followed by **`mtu`** must be called to program MTU on an interface.

- When the startup configuration has more than three (3) unique MTU values defined on different interfaces of the switch, there is no guarantee which unique MTU values will be active when the switch is rebooted.
- L3 MTU is not enforced on Multicast packets for Jericho2 based platforms.

13.11 Segment Security

- Overview of MSS-Group
- Configuring MSS-Group
- Limitations
- Show Commands
- Segment Security Commands

13.11.1 Overview of MSS-Group

Hosts and networks can be grouped into segments based on their prefixes; the MSS-Group feature (also called Segment Security) allows policies to be applied to segments rather than to interfaces or subnets. Policies define inter-segment and intra-segment rules; for example, segment A is allowed to communicate with segment B, or hosts in segment B are not allowed to communicate with each other.

By default, traffic directed to a segment is dropped; an explicit allow policy is required to allow communication. The two directions of traffic are handled independently; to allow traffic between two segments, forward policy must be configured in both segments.

13.11.2 Configuring MSS-Group

To configure MSS-Group (segment security) to control groups of IPv4 and/or IPv6 addresses (called “segments”), define one or more match lists, create segments based on those match lists, create policies governing traffic to individual segments, define default policy for all segments, and enable the MSS-Group feature. Up to 60 segments can be defined across all VRFs. Traffic to and from VLANs with no SVI configured are considered part of the default VRF, and are subject to the policies defined in the default VRF. This feature does not require routing to be enabled on the switch, even though the mode name starts with the word “router.”

Define Match Lists

Use the **`match-list input`** command to define an IPv4 or IPv6 subnet list. Each match list must contain only one type of prefix, either IPv4 or IPv6. It cannot contain a mixture. Each match list name of a given type must be unique, but an IPv4 match list and an IPv6 match list can have the same name.

Example

- The following commands define two IPv4 match lists named camera-prefixes and admin-prefixes and two IPv6 match lists also named camera-prefixes and admin-prefixes, and add a total of seven prefixes.

```
switch(config)# match-list input prefix-ipv4 camera-prefixes
switch(config-match-list-prefix-ipv4-camera-prefixes)# match prefix-ipv4
 69.89.31.200/32
switch(config-match-list-prefix-ipv4-camera-prefixes)# match prefix-ipv4
 69.89.31.201/32
switch(config-match-list-prefix-ipv4-camera-prefixes)# match prefix-ipv4
 70.89.31.0/24
switch(config-match-list-prefix-ipv4-camera-prefixes)# exit
switch(config)# match-list input prefix-ipv6 camera-prefixes
switch(config-match-list-prefix-ipv6-camera-prefixes)# match prefix-ipv6
 2001:0:9d38:6ab8::/64
switch(config-match-list-prefix-ipv6-camera-prefixes)# match prefix-ipv6
 2002:0:9d38:6ab8::3/128
switch(config-match-list-prefix-ipv6-camera-prefixes)# exit
```

```
switch(config)# match-list input prefix-ipv4 admin-prefixes
switch(config-match-list-prefix-ipv4-admin-prefixes)# match prefix-ipv4
 80.80.0.0/16
switch(config-match-list-prefix-ipv4-admin-prefixes)# exit
switch(config)# match-list input prefix-ipv6 admin-prefixes
switch(config-match-list-prefix-ipv6-admin-prefixes)# match prefix-ipv6
 2003:0:9d38:6ab8::/64
switch(config-match-list-prefix-ipv6-admin-prefixes)# exit
switch(config)#
```

Define Segments using Match Lists

Use the **segment** command to define a segment. A segment contains one or two match lists, one of type IPv4 and the other of type IPv6.

Example

- The following commands define segments using the match lists configured above.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
switch(config-router-seg-sec-vrf-default)# segment camera
switch(config-router-seg-sec-vrf-segment-camera)# definition
switch(config-router-seg-sec-vrf-segment-def)# match prefix-ipv4 camera-
prefixes
switch(config-router-seg-sec-vrf-segment-def)# match prefix-ipv6 camera-
prefixes
switch(config-router-seg-sec-vrf-segment-def)# exit
switch(config-router-seg-sec-vrf-segment-camera)# exit
switch(config-router-seg-sec-vrf-default)# segment secure-admin
switch(config-router-seg-sec-vrf-segment-secure-admin)# definition
switch(config-router-seg-sec-vrf-segment-def)# match prefix-ipv4 admin-
prefixes
switch(config-router-seg-sec-vrf-segment-def)# match prefix-ipv6 admin-
prefixes
switch(config-router-seg-sec-vrf-segment-def)#  exit
switch(config-router-seg-sec-vrf-segment-secure-admin) exit
switch(config-router-seg-sec-vrf-default)# exit
switch(config-router-seg-sec)# exit
switch(config)#
```

Define Policies Between Segments

Use the **policies** command to drop or forward traffic to a segment from specific other segments. Two built-in policies are available: policy-forward-all to forward traffic between segments, and policy-drop-all to drop traffic between segments. By default, the drop-all policy is enabled.

Example

- The following commands allow bidirectional traffic between the two segments defined above.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
switch(config-router-seg-sec-vrf-default)# segment camera
switch(config-router-seg-sec-vrf-segment-camera)# policies
switch(config-router-seg-sec-vrf-segment-policies)# from secure-admin policy
 policy-forward-all
switch(config-router-seg-sec-vrf-segment-policies)# exit
switch(config-router-seg-sec-vrf-segment-camera)# exit
switch(config-router-seg-sec-vrf-default)# segment secure-admin
switch(config-router-seg-sec-vrf-segment-secure-admin)# policies
switch(config-router-seg-sec-vrf-segment-policies)# from secure-admin policy
 policy-forward-all
```

```
switch(config-router-seg-sec-vrf-segment-policies)# exit
switch(config-router-seg-sec-vrf-segment-secure-admin)# exit
switch(config-router-seg-sec-vrf-default)# exit
switch(config-router-seg-sec)# exit
switch(config)#
```

Enable MSS-Group

By default, MSS-Group is not enabled. Use the **no shutdown** command to enable it. Use the **shutdown** command to disable it.

Examples

- The following commands enable MSS-Group.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# no shutdown
switch(config-router-seg-sec)# exit
switch(config)#
```

- The following commands disable MSS-Group.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# shutdown
switch(config-router-seg-sec)# exit
switch(config)#
```

Configuring Default Forward/Drop Behavior

By default, when MSS-Group is first enabled, all traffic to nodes in a segment is dropped unless explicitly allowed by a "forward-all" policy as shown above. This includes traffic within the segment. Use the **no segment policy** command to change this behavior to allow intra-segment traffic.

Example

- The following commands allow all traffic within each segment as well as between segments.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# no segment policy policy-drop-all default
switch(config-router-seg-sec)# exit
switch(config)#
```

You can modify the policy for each segment and in greater detail with the **policies** command.

Example

- The following commands prevent nodes in the **camera** segment from communicating with each other.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
switch(config-router-seg-sec-vrf-default)# segment camera
switch(config-router-seg-sec-vrf-segment-camera)# policies
switch(config-router-seg-sec-vrf-segment-policies)# from camera policy
 policy-drop-all
switch(config-router-seg-sec-vrf-segment-policies)# exit
switch(config-router-seg-sec-vrf-segment-camera)# exit
switch(config-router-seg-sec-vrf-default)# exit
switch(config-router-seg-sec)# exit
switch(config)# exit
```

13.11.3 Limitations

- Multicast and Link Local prefixes are not supported.
- Traffic disruption during prefix and policy configuration is expected. We do not support atomicity during segment and prefix configuration.
- MSS-Group and URPF feature interaction is not supported. If both features are configured (misconfiguration), the platform gives URPF higher priority and removes any existing segment configurations from hardware.
- Prefixes entries failed to get installed in hardware (because of insufficient hardware resources) are retried periodically till resources become available and prefixes are successfully installed. However, the same is not true for policy entries. There is no retry mechanism implemented for failed policy entries. The user needs to free up hardware resources and re-enable the MSS-Group feature after removing it once.
- Custom policies can not be configured. Clients can choose from two built-in policies ‘policy-drop-all’ and ‘policy-forward-all’.
- A given prefix can only be part of a single segment in VRF. Attempting to configure the same prefix in more than one segment leads to undefined traffic forwarding behavior.
- The same prefix can not be configured in both MSS-Group and MSS-L3 configurations.
- SSU can be performed with MSS-Group configured but the traffic flows for MSS-Group configuration will not be hitless.
- DHCP discovery packets with broadcast destination IP of ***255.255.255.255*** will only match ***0.0.0.0/0*** prefix.
- All traffic sourced from and/or destined to switch owned IPs are allowed regardless of MSS-Group configuration.
- Due to source and destination IP lookup being required, the capacity of the LPM table is halved when MSS-G is enabled. The host table capacity is unchanged as the source and destination lookup is always enabled by default.

13.11.4 Show Commands

The `show` commands available to examine the configuration and status of MSS-Group include:

- **show segment-security [vrf *<vrf-name>*][segment *<seg-name>*]**

```
switch# show segment-security
VRF : default
  Segment      Interfaces Prefix IPv4     Prefix IPv6     From Segment
 Policy
  ------------ ---------- --------------- --------------- ------------
 ------------------
  camera                  camera-prefixes camera-prefixes secure-admin
 policy-forward-all
  secure-admin            admin-prefixes  admin-prefixes  camera
 policy-forward-all
```

- **show match-list {prefix-ipv4 | prefix-ipv6}[*<list-name>*]**
-

```
switch# show match-list prefix-ipv4
Name            Prefix
--------------- ---------------
admin-prefixes  80.80.0.0/16
camera-prefixes 69.89.31.200/32
                69.89.31.201/32
                70.89.31.0/24

switch# show match-list prefix-ipv6
Name            Prefix
--------------- ----------------------
admin-prefixes  2003:0:9d38:6ab8::/64
camera-prefixes 2001:0:9d38:6ab8::/64
```

```
                  2002:0:9d38:6ab8::3/128
```

- **show segment-security hardware summary [vrf<*vrf-name*>][segment<*seg-name*>]**

  This command shows the hardware ID, number of prefixes, and number of successfully programmed prefixes for each VRF and segment specified. By default, all VRFs and segments are shown.

```
switch# show segment-security hardware summary
VRF: default
Segment              Hardware ID     Prefixes   Programmed
-------------------- --------------- ---------- ---------------
camera               63              5          5
secure-admin         62              2          2
```

- **show segment-security hardware detail [vrf<*vrf-name*>][segment<*seg-name*>]**

  This command shows the hardware ID assigned to each segment, the prefixes in each segment, and the adjacency index for each prefix (as determined from L3 hardware tables).

```
switch# show segment-security hardware detail
VRF: default
Segment              Hardware ID     Prefixes
 Adj Index
-------------------- --------------- ---------------------------------------
 ---------------
camera               63              69.89.31.200/32
 1
                                     69.89.31.201/32
 1
                                     70.89.31.0/24
 1
                                     2001:0:9d38:6ab8::/64
 2
                                     2002:0:9d38:6ab8::3/128
 2
secure-admin         62              80.80.0.0/16
 1
                                     2003:0:9d38:6ab8::/64
 2
```

- **show segment-security hardware routes [vrf<*vrf-name*>][segment<*seg-name*>]**

  Since MSS Group prefixes use L3 hardware tables, the prefixes can overlap with FIB routes. So each prefix is assigned a route type. There are three possible classifications for a prefix:

  1. The prefix does not overlap with an FIB route. This prefix has route type 'S'.
  2. The prefix is also configured in the FIB. If a segment prefix is identical to an FIB prefix, it is given the route type 'S,F'.
  3. The prefix overlaps with an FIB entry but there is no exact match in the FIB. This prefix has the route type 'F'.

  The following command shows the route types for prefixes in hardware.

```
switch# show segment-security hardware routes
Codes: S - Segment prefix
       F - FIB route
       S,F - Segment prefix which is also present in FIB
VRF: default
Segment              Hardware ID     Routes
 Route Type
-------------------- --------------- ---------------------------------------
 ----------
camera               63              69.89.31.200/32
 S
```

```
                                        69.89.31.201/32
 S
                                        70.89.31.0/24
 S,F
                                        2001:0:9d38:6ab8::/64
 S
                                        2002:0:9d38:6ab8::3/128
 S
secure-admin          62                80.80.0.0/16
 S
                                        2003:0:9d38:6ab8::/64
 S
```

- **show segment-security hardware counters[vrf<*vrf-name*>]**

  This command displays the counters for policies in each segment, including the default policies. For each policy configured between two segments, the Hit counter shows all hits, whether the packets were dropped or forwarded. The Drop counter shows which of those hits were dropped. There are also lines for the default policy of each segment, and the Drop counter includes packets which do not match a configured policy but are dropped by these default policies.

```
switch# show segment-security hardware counters
VRF: default
Policy                Hit         Drop
--------------------- ----------- ----------
policy-drop-all       6           6
policy-forward-all    13          0
Dest Segment          Source Segment        Policy                Hit         Drop
--------------------- --------------------- --------------------- ----------
 ----------
camera                *                     n/a                   0           3
camera                camera                                      6           6
camera                secure-admin                                4           0
secure-admin          *                     n/a                   0           12
secure-admin          camera                                      9           0
```

- **clear segment-security hardware counters**

  This command clears the Hit and Drop counters for each policy, setting them to ***0***.

13.11.5 Segment Security Commands

Global Configuration Commands

- match-list input
- router segment-security

Match-List Input Configuration Commands

- match (match-list input)

Router Segment-Security Configuration Commands

- segment policy policy-drop-all default
- shutdown (router segment-security)
- vrf (router segment-security)

Router Segment-Security VRF Configuration Commands

- segment

Router Segment-Security VRF Segment Configuration Commands

- definition (segment)
- policies (segment)

Router Segment-Security VRF Segment Policies Configuration Commands

- from (segment policies)

Router Segment-Security VRF Segment Definition Configuration Commands

- match (segment definition)

Segment-Security Clear and Show Commands

- clear segment-security hardware counters
- show match-list
- show segment-security
- show segment-security hardware counters
- show segment-security hardware detail
- show segment-security hardware routes
- show segment-security hardware summary

13.11.5.1 clear segment-security hardware counters

The **`clear segment-security hardware`** command clears the MSS-Group (segment security) Hit and Drop counters for all hits, and the hits and drops for each separate segment's policy. All MSS-Group counters are set to 0.

Command Mode

Privileged EXEC

Command Syntax

`clear segment-security hardware counters`

Examples

- This command clears all counters for MSS-Group.

```
switch# clear segment-security hardware counters
switch#
```

13.11.5.2 definition (segment)

The **`definition`** command enters Router Segment-Security VRF Segment Definition Configuration mode. This is not a group change mode. Changes are applied to ***running-config*** immediately. The **`exit`** command does not affect the configuration.

The **`no definition`** and **`default definition`** commands clear the segment definitions from ***running-config***.

Command Mode

Router Segment-Security VRF Segment Configuration

Command Syntax

`definition`

`no definition`

`default definition`

Commands Available in Router Segment-Security VRF Segment Definition Configuration Mode

`match (segment definition)`

Example

- These commands enter Router Segment-Security VRF Segment Definition mode for the segment "admin".

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
switch(config-router-seg-sec-vrf-default)# segment admin
switch(config-router-seg-sec-vrf-segment-admin)# definition
switch(config-router-seg-sec-vrf-segment-def)#
```

13.11.5.3 from (segment policies)

The **from** command adds a policy to a segment in order to filter traffic from a specified segment (the same segment or a different segment). The policy can be either policy-drop-all or policy-forward-all. The default is policy-drop-all. Therefore, for a segment to allow traffic among its own members, it requires a policy-forward-all policy for itself. You can add any number of policies.

The **no from** and **default from** commands clear the segment policy from the ***running-config***.

Command Mode

Router Segment-Security VRF Segment Policies Configuration

Command Syntax

from ***segment_name*** **policy** ***policy_type***

no from ***segment_name*** [**policy** ***policy_type***]

default from ***segment_name*** [**policy** ***policy_type***]

Parameters

policy_type The type of policy. The possible values are "policy-drop-all" and "policy-forward-all". The default is "policy-drop-all".

segment_name The name of the segment to filter. This can be the segment currently being configured, to give you control over traffic within the segment.

Related Command

segment policy policy-drop-all default

Example

These commands add three policies to the segment ***admin***. One policy allows traffic within the ***admin*** segment itself. The second policy drops all traffic from segment ***seg1***. The third policy forwards all traffic from ***seg2***.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
switch(config-router-seg-sec-vrf-default)# segment admin
switch(config-router-seg-sec-vrf-segment-admin)# policies
switch(config-router-seg-sec-vrf-segment-policies)# from admin policy policy-
forward-all
switch(config-router-seg-sec-vrf-segment-policies)# from seg1 policy policy-
drop-all
switch(config-router-seg-sec-vrf-segment-policies)# from seg2 policy policy-
forward-all
switch(config-router-seg-sec-vrf-segment-policies)#
```

13.11.5.4 match-list input

The **match-list input** command enters Match List Configuration mode for the specified match list, creating one if it does not exist. The commands in this mode apply changes to ***running-config*** immediately. The **exit** command is not needed to save the changes to the configuration.

The **no match-list input** and **default match-list input** commands remove the specified match list from ***running-config***.

Command Mode

Global Configuration Mode

Command Syntax

match-list input {**prefix-ipv4|prefix-ipv6**} ***match_list_name***

no match-list input {**prefix-ipv4|prefix-ipv6**} ***match_list_name***

no match-list input {**prefix-ipv4|prefix-ipv6**} ***match_list_name***

Parameters

- prefix-ipv4 This match list has IPv4 prefixes only.
- prefix-ipv6 This match list has IPv6 prefixes only.
- ***match_list_name*** The name of the match-list to add to. If it does not exist it will be created.

Examples

- The following command creates an IPv4 match list called **camera-prefixes** and enters Match List Configuration mode.

```
switch(config)# match-list input prefix-ipv4 camera-prefixes
switch(config-match-list-prefix-ipv4-admin-prefixes)#
```

- The following command removes the IPv4 match list **camera-prefixes** from ***running-config***.

```
switch(config)# no match-list input prefix-ipv4 camera-prefixes
switch(config)#
```

13.11.5.5 match (match-list input)

The **match** command adds an entry to a match list. Each entry in a given match list must be of the same type, either IPv4 or IPv6. This command updates ***running-config*** immediately. It is not necessary to use the **exit** command to save changes.

The **no match** and **default match** commands remove the specified match list entry from the match list in ***running-config***.

Command Mode

Match List inpur Configuration Mode

Command Syntax

match {prefix-ipv4|prefix-ipv6} ***ip_address_prefix***

no match {prefix-ipv4|prefix-ipv6} ***ip_address_prefix***

default match {prefix-ipv4|prefix-ipv6} ***ip_address_prefix***

Parameters

- prefix-ipv4 This prefix is IPv4. You cannot mix prefix types in a single match list.
- prefix-ipv6 This prefix is IPv6. You cannot mix prefix types in a single match list.

- ***ip_address_prefix*** The prefix to add. For IPv4, it is of the form **A.B.C.D/E**. For IPv6, it is of the form **A:B:C:D:E:F:G:H/I**.

Examples

- The following commands add two IPv4 entries to the match list **camera-prefixes**.

```
switch(config)# match-list input prefix-ipv5 camera-prefixes
switch(config-match-list-prefix-ipv4-camera-prefixes)# match prefix-ipv4
 69.89.31.200/32
switch(config-match-list-prefix-ipv4-camera-prefixes)# match prefix-ipv4
 69.89.31.201/32
switch(config-match-list-prefix-ipv4-camera-prefixes)#
```

- The following command removes one entry from the **camera-prefixes** match list.

```
switch(config)# match-list input prefix-ipv4 camera-prefixes
switch(config-match-list-prefix-ipv4-camera-prefixes)# no match prefix-ipv4
 69.89.31.201/32
switch(config-match-list-prefix-ipv4-camera-prefixes)#
```

13.11.5.6 match (segment definition)

The **match** command adds a match list to a segment definition. The match list cannot contain both IPv4 and IPv6 prefixes. One match list of each type can be added. The segment definition is updated in ***running-config*** immediately.

The **no match** command removes the specified match list from the segment definition in ***running-config***.

The **default match** command removes the specified match list from the segment definition in ***running-config***.

Command Mode

Router Segment-Security VRF Segment Definition Configuration

Command Syntax

match {prefix-ipv4|prefix-ipv6} ***match_list_name***

no match {prefix-ipv4|prefix-ipv6} ***match_list_name***

default match {prefix-ipv4|prefix-ipv6} ***match_list_name***

Parameters

prefix-ipv4 The match list contains IPv4 prefixes.

prefix-ipv6 The match list contains IPv6 prefixes.

match_list_name The name of the match list.

Examples

These commands add two match lists to the segment ***admin***, an IPv4 match list named **admin-prefixes** and an IPv6 match list also named **admin-prefixes**.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
switch(config-router-seg-sec-vrf-default)# segment admin
switch(config-router-seg-sec-vrf-segment-admin)# definition
switch(config-router-seg-sec-vrf-segment-def)# match prefix-ipv4 admin-prefixes
switch(config-router-seg-sec-vrf-segment-def)# match prefix-ipv6 admin-prefixes
switch(config-router-seg-sec-vrf-segment-def)#
```

13.11.5.7 policies (segment)

The **policies** command places the switch in Router Segment Security VRF Segment Policies Configuration mode. In this mode, the command **from** creates a policy for the segment. A segment can contain multiple policies.

The **no policies** command clears the segment policies from ***running-config***.

The **default policies** command clears the segment policies from ***running-config***.

Command Mode

Router Segment-Security VRF Segment Configuration

Command Syntax

policies

no policies

default policies

Examples

This command places the switch in Router Segment-Security VRF Segment Policies configuration mode for the segment ***admin***.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
switch(config-router-seg-sec-vrf-default)# segment admin
switch(config-router-seg-sec-vrf-segment-admin)# policies
switch(config-router-seg-sec-vrf-segment-policies)#
```

13.11.5.8 router segment-security

The **router segment-security** command enters Router Segment-Security Configuration Mode. This mode is required to enable or disable MSS-Group (segment security), and to enter the Router Segment-Security VRF configuration mode to create segments from match lists and to configure MSS-Group.

The **no router segment-security** command removes the MSS-Group configuration from ***running-config.***

The **default router segment-security** command removes the MSS-Group configuration from ***running-config.***

Command Mode

Global Configuration Mode

Command Syntax

router segment-security

no router segment-security

default router segment-security

Commands Available In Router Segment-Security Configuration Mode

segment

shutdown

vrf

Examples

- The following command enters Router Segment-Security configuration Mode.

```
switch(config)# router segment-security
switch(config-router-seg-sec)#
```

- The following command disables MSS-Group and removes the MSS-Group configuration from the ***running-config.***

```
switch(config)# no router segment-security
switch(config)#
```

13.11.5.9 segment

The **segment** command enters Router Segment-Security VRF Segment Configuration mode, creating a segment if one does not exist. The commands in this mode apply changes to ***running-config*** immediately. The **exit** command does not affect the configuration.

The **no segment** command and the **default segment** command clear the segment from ***running-config***.

Command Mode

Router Segment-Security VRF Configuration

Command Syntax

segment ***segment_name***

no segment ***segment_name***

default segment ***segment_name***

Parameters

- ***segment_name*** the name of the segment.

Commands Available in Router Segment-Security VRF Segment Configuration Mode

- **definition**
- **policies**

Example

The following command creates a new segment called ***admin*** and enters Segment Configuration mode.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
switch(config-router-seg-sec-vrf-default)# segment admin
switch(config-router-seg-sec-vrf-segment-admin)#
```

13.11.5.10 segment policy policy-drop-all default

The **segment policy policy-drop-all default** command configures the switch to drop all traffic to all segments. This is the default.

The **no segment policy policy-drop-all default** command allows segments to receive traffic. This is necessary to allow traffic within a segment.

The **default segment policy policy-drop-all default** command restores the default, so that all traffic to all segments is dropped.

Command Mode

Router Segment-Security Configuration

Command Syntax

segment policy policy-drop-all default

no segment policy policy-drop-all default

default segment policy policy-drop-all default

Example

This command removes the policy-drop-all policy from the general segment security configuration.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# no segment policy policy-drop-all default
switch(config-router-seg-sec)#
```

13.11.5.11 show match-list

The **show match-list** command displays match lists of type IPv4 or IPv6.

Command Mode

Privileged EXEC

Command Syntax

show match-list {prefix-ipv4 | prefix-ipv6} [*list-name*]

Parameters

- **prefix-ipv4** IPv4 prefix list.
- **prefix-ipv6** IPv6 prefix list.
- ***list-name*** match list name.

Examples

- The following command displays all the IPv4 match lists and their contents.

```
switch# show match-list prefix-ipv4
Name            Prefix
--------------- ---------------
admin-prefixes  80.80.0.0/16
camera-prefixes 69.89.31.200/32
                69.89.31.201/32
                70.89.31.0/24 switch#
```

- The following command displays the contents of the IPv6 match list camera-prefixes.

```
switch# show match-list prefix-ipv6 camera-prefixes
Name            Prefix
--------------- ----------------------
camera-prefixes 2001:0:9d38:6ab8::/64
                2002:0:9d38:6ab8::3/128 switch#
```

13.11.5.12 show segment-security

The **show segment-security** command shows the status and configuration of MSS-Group (segment security).

Command Mode

Privileged EXEC

Command Syntax

show segment-security [{[**vrf** ***vrf_name***] [**segment** ***seg_name***] | **application** [***application_name***] | **policy** [***policy_name***] | **segment** ***segment_name*** | **sessions** [**vrf** ***vrf_name***] | **status** [**vrf** ***vrf_name***] [**segment** ***seg_name***]}]

Parameters

- **vrf** Show information for a particular VRF. By default, all VRFs are shown.
  - ***vrf_name*** VRF name to show. The default VRF instance is named "default".
- **segment** Show information for a particular segment. By default, all segments are shown.
  - ***segment_name*** The name of the segment to show.
- **application** Show status and configuration for applications. By default, no application information is shown.
  - ***application_name*** The name of the application to show. If this is omitted, all applications are shown.
- **policy** Show information about policies.
  - ***policy_name*** The name of the policy to show. If this is omitted, all policies are shown.
- **sessions** Show information about sessions.
  - **vrf** Show session information about a particular VRF.
    - ***vrf_name*** The VRF for which to show session information. The default VRF is named "default".
- **status** Show status information.

Guidelines

If both ***vrf*** and ***segment*** parameters are specified, the ***vrf*** parameter must precede the ***segment*** parameter. Command syntax such as **show segment-security hardware detail segment** ***segment_name*** **vrf** ***vrf_name*** is not valid.

Examples

- This command displays the MSS-Group configuration for all VRF instances and all segments.

```
switch# show segment-security
VRF : default
  Segment      Interfaces Prefix IPv4     Prefix IPv6     From Segment
 Policy
  ------------ ---------- --------------- --------------- ------------
 ------------------
  camera                  camera-prefixes camera-prefixes secure-admin
 policy-forward-all
  secure-admin            admin-prefixes  admin-prefixes  camera
 policy-forward-all switch#
```

- This command shows the MSS-Group configuration for the default VRF instance only.

```
switch# show segment-security vrf default
VRF : default
  Segment      Interfaces Prefix IPv4     Prefix IPv6     From Segment
 Policy
  ------------ ---------- --------------- --------------- ------------
 ------------------
  camera                  camera-prefixes camera-prefixes secure-admin
 policy-forward-all
  secure-admin            admin-prefixes  admin-prefixes  camera
 policy-forward-all switch#
```

- This command shows the MSS-Group configuration for the **camera** segment.

```
switch# show segment-security segment camera
VRF : default
```

```
 Segment      Interfaces Prefix IPv4     Prefix IPv6     From Segment
Policy
 ------------ ---------- --------------- --------------- ------------
-----------------
 camera                  camera-prefixes camera-prefixes secure-admin
policy-forward-all switch#
```

- This command shows information for all applications.

```
switch# show segment-security applications
application: app-match-all
   protocol: all switch#
```

- This command shows information for the policy **policy-drop-all**.

```
switch# show segment-security policy policy-drop-all
policy: policy-drop-all [readonly]
   10 application app-match-all action drop stateless switch#
```

13.11.5.13 show segment-security hardware counters

The **show segment-security hardware counters** command displays the counters for policies in each segment, including the default policies. For each policy configured between two segments, the Hit counter shows all hits, whether the packets were dropped or forwarded. The Drop counter shows which of those hits were dropped. There are also lines for the default policy of each segment, and the Drop counter includes packets which do not match a configured policy but are dropped by these default policies. To clear the Hit and Drop counters for each policy, setting them to 0, use the **clear segment-security hardware counters** command.

Command Mode

Privileged EXEC

Command Syntax

show segment-security hardware counters [vrf ***vrf_name***]

Parameters

- **vrf** Show details for a specific VRF. If this parameter is omitted, details for all VRFs are shown.
- ***vrf_name*** The VRF to show. To show the default VRF, specify "default".

Example

This command displays the policy and counters for policies configured for all segments in VRF ***site_b***.

```
switch# show segment-security hardware counters vrf site_b
VRF: site_b
Policy               Hit        Drop
-------------------- ---------- ----------
policy-drop-all      6          6
policy-forward-all   13         0

Dest Segment         Source Segment       Policy               Hit        Drop
-------------------- -------------------- -------------------- ----------
 ----------
camera               *                    n/a                  0          3
camera               camera                                    6          6
camera               secure-admin                              4          0
secure-admin         *                    n/a                  0          12
secure-admin         camera                                    9          0
 switch#
```

13.11.5.14 show segment-security hardware detail

The **show segment-security hardware detail** command displays the hardware ID allocated to each segment, the prefixes programmed in hardware for each segment, and the adjacency index used by each prefix (as determined from L3 hardware tables).

Command Mode

Privileged EXEC

Command Syntax

show segment-security hardware detail [**vrf** ***vrf_name***][**segment** ***seg_name***]

Parameters

- **vrf** Show details for a specific VRF. If this parameter is omitted, details for all VRFs are shown.
- ***vrf_name*** The name of the VRF to show details for. To show details for the default VRF, you must specify "default".
- **segment** Show details for a specific segment. If this parameter is omitted, details for all segments are shown.
- ***seg_name*** The name of the segment to show details for.

Guidelines

If both **vrf** and **segment** parameters are specified, the **vrf** parameter must come first. The command syntax **show segment-security hardware detail segment *segment_name* vrf *vrf_name*** is not valid.

Example

This command displays the hardware IDs allocated to each segment in vrf ***site_a***, the prefixes in each segment, and the adjacency index for each prefix (as determined from L3 hardware tables).

```
switch# show segment-security hardware detail vrf site_a
VRF: site_a
Segment         Hardware ID     Prefixes                  Adj Index
--------------- --------------- ------------------------- ---------------
camera          63              69.89.31.200/32            1
                                69.89.31.201/32            1
                                70.89.31.0/24              1
                                2001:0:9d38:6ab8::/64      2
                                2002:0:9d38:6ab8::3/128    2
secure-admin    62              80.80.0.0/16               1
                                2003:0:9d38:6ab8::/64      2 switch#
```

13.11.5.15 show segment-security hardware routes

The **show segment-security hardware routes** command displays the route and type for each programmed prefix in hardware. Since MSS-Group prefixes use L3 hardware tables, the prefixes can overlap with FIB routes, so each prefix is assigned a route type. There are three possible classifications for a prefix:

1. The prefix does not overlap with an FIB route. This prefix has route type S.
2. The prefix is also configured in the FIB. If a segment prefix is identical to an FIB prefix, it is given the route type S,F.
3. The prefix overlaps with an FIB entry but there is no exact match in the FIB. This prefix has the route type F.

Command Mode

Privileged EXEC

Command Syntax

show segment-security hardware routes [**vrf** *vrf-name*][**segment** ***seg-name***]

Parameters

- **vrf** Show details for a specific VRF. If this parameter is omitted, details for all VRFs are shown.
- ***vrf_name*** The name of the VRF to show details for. To show details for the default VRF, you must specify "default".
- **segment** Show details for a specific segment. If this parameter is omitted, details for all segments are shown.
- ***seg_name*** The name of the segment to show details for.

Guidelines

If both **vrf** and **segment** parameters are specified, the **vrf** parameter must come first. The command syntax **`show segment-security hardware detail`** **segment *segment_name* vrf *vrf_name*** is not valid.

Example

This command displays the route and type for programmed prefixes in hardware for the VRF named ***site_a*** and the segment ***camera***.

```
switch# show segment-security hardware routes vrf site_a segment camera
Codes: S - Segment prefix
       F - FIB route
       S,F - Segment prefix which is also present in FIB

VRF: site_a
Segment       Hardware ID   Routes                      Route Type
------------- ------------- --------------------------- ----------
camera        63            69.89.31.200/32              S
                            69.89.31.201/32              S
                            70.89.31.0/24                S,F
                            2001:0:9d38:6ab8::/64        S
                            2002:0:9d38:6ab8::3/128      S switch#
```

13.11.5.16 show segment-security hardware summary

The **`show segment-security hardware summary`** command displays the hardware ID, number of prefixes, and number of successfully programmed prefixes for each VRF and segment specified. By default, all VRFs and segments are shown.

Command Mode

Privileged EXEC

Command Syntax

`show segment-security hardware summary` [vrf ***vrf-name***][segment ***seg-name***]

Parameters

- **vrf** Show details for a specific VRF. If this parameter is omitted, details for all VRFs are shown.
- ***vrf_name*** The name of the VRF to show details for. To show details for the default VRF, you must specify "default".
- **segment** Show details for a specific segment. If this parameter is omitted, details for all segments are shown.
- ***seg_name*** The name of the segment to show details for.

Guidelines

If both **vrf** and **segment** parameters are specified, the **vrf** parameter must come first. The command syntax **`show segment-security hardware detail`** **segment *segment_name* vrf *vrf_name*** is not valid.

Example

This command displays the hardware ID allocated to each configured segment, the number of prefixes configured, and the number of prefixes successfully programmed in hardware for all VRFs and all segments.

```
switch# show segment-security hardware summary

VRF: default
Segment              Hardware ID     Prefixes   Programmed
-------------------- --------------- ---------- ---------------
camera               63              5          5
secure-admin         62              2          2 switch#
```

13.11.5.17 shutdown (router segment-security)

The **shutdown** command disables MSS-Group (segment security) in the switch. This is the default. The **no shutdown** and **default shutdown** commands enable MSS-Group.

Command Mode

Router Segment-Security Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Example

This command enables MSS-Group in the switch.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# no shutdown
switch(config-router-seg-sec)#
```

13.11.5.18 vrf (router segment-security)

The **vrf** command enters Router Segment-Security VRF Configuration mode, creating a VRF instance if necessary, to create and configure MSS-Group segments.

Command Mode

Router Segment-Security Configuration

Command Syntax

vrf ***vrf_instance***

Parameters

- ***vrf_instance*** The name of the VRF instance. To configure MSS-Group for the default VRF instance, specify "default".

Commands Available in Router Segment-Security VRF Configuration Mode

- **segment**

Example

The following command enters Router Segment-Security VRF Configuration mode for the default VRF instance.

```
switch(config)# router segment-security
switch(config-router-seg-sec)# vrf default
```

```
switch(config-router-seg-sec-vrf-default)#
```

Chapter 14

IP Services

The Internet Protocol Services (IP Services) chapter contains the following section(s):

- CloudVision eXchange (CVX)

14.1 CloudVision eXchange (CVX)

CloudVision eXchange (CVX) provides a single access point for real-time provisioning, orchestration and integration with third-party controllers. CVX aggregates and distributes operational state information across a set of EOS switches to support applications that provide network services. See the *CloudVision User Guide*, https://www.arista.com/en/support/product-documentation, for additional information.

Topics in this section include:

- Upgrading CVX
- CVX Overview
- CVX Services
- Deploying CVX
- CVX Configuration
- CVX Secure out-of-band Connection
- CVX High Availability
- CVX VIP
- CVX Commands

14.1.1 Upgrading CVX

You can upgrade CVX from a previous version to the current version by performing a few simple tasks. You can use the following procedure to upgrade any previous version of CVX to the current version.

Requirements

Make sure you follow these requirements during the upgrade process.

- If you have CVP, CVX and client switches in your environment, make sure you upgrade each component in the following order:
  - Upgrade CVP first.
  - Upgrade the CVX cluster.
  - Upgrade the client switches. The reason for this is to ensure backward compatibility.
  - You must upgrade the CVX cluster before you upgrade the client switches.
  - If the CVX cluster is a three node cluster, make sure that only one node of the cluster is down at any one time during the upgrade process. (The order in which you upgrade the nodes does not matter.)

Pre-requisites

Before you begin the upgrade, make sure that:

- You perform a backup to ensure that you can restore data if needed.

- You download the latest version of CVX from Arista's Software Download page (https://www.arista.com/en/support/software-download).

Complete the following steps to upgrade CVX.

1. Login to the cluster to be upgraded. (You can login to any node.)
2. Upgrade the node. You must deploy a new image to perform the upgrade.
3. Wait for the node you are upgrading to rejoin the cluster. Once the node has rejoined, go to the next step. (The node automatically rejoins the cluster as a follower node.)
4. Repeat steps 1 through 3 to upgrade the two remaining nodes one node at a time. It does not matter the order in which you upgrade the remaining nodes.

14.1.2 CVX Overview

A CVX deployment includes CVX and a set of CVX clients to which CVX provides services. CVX is not part of the data plane, nor does it receive data-path traffic. All CVX components exist as agents that run on EOS instances.

For more information, see:

- System Requirements
- CVX Infrastructure
- CVX Features
- CVX Clients

14.1.2.1 System Requirements

Certain hardware and software is required to be able to use CloudVision eXchange in your CloudVision virtual appliance implementation.

The CloudVision eXchange should be installed on a single system along with CloudVision Portal.

The following table lists the minimum hardware and software required to use CloudVision eXchange.

Required Hardware

The hardware required to use the CloudVision eXchange are:

- CPU: 4 cores (base), 8 cores (recommended)
- RAM: 4G (base), 8G (recommended)
- Disk: 4G

Required Software

The software required to use the CloudVision eXchange are:

- EOS switches: Recommend 4.16.8M or later

  Note: It is a best practice and highly recommended that the version of CVX should match the version running on the switches.
- CloudVision Portal: version 2016.1

  (CloudVision Portal software is required if you want to use it in conjunction with CloudVision eXchange. If you plan to use only CloudVision eXchange, CloudVision Portal software is not required.)

  Note: CVX supports live vMotion.

14.1.2.2 CVX Infrastructure

CVX provides a single integration point into network-wide services running across CVX clients. CVX is typically deployed as an EOS instance running on a VM (vEOS). The CVX infrastructure consists of a CVX instance functioning as a server and a set of CVX clients. The CVX server uses a heartbeat keepalive (KA) mechanism to maintain contact with its clients.

When de-configuring or shutting down CVX, client services should be shut down first.

14.1.2.3 CVX Features

CVX manages communications among the network CVX clients, and provides an integration point for services to those clients. CVX also discovers the physical network topology by aggregating topology information it receives from its client devices.

14.1.2.4 CVX Clients

CVX client is the agent that allows a switch to interact with a CVX server to access CVX services. Enabling the CVX client includes providing the IP address or host name of the device running CVX. The CVX client can then access services that are enabled on the CVX server.

The CVX client must be enabled to access the CVX server and the services it offers. Individual services may require additional configuration statements.

Services should be shut down or de-configured on clients before shutting down or de-configuring CVX. CVX-controlled switch features may continue to run after shutting down CVX if they are not explicitly shut down or de-configured prior to shutting down CVX.

14.1.3 CVX Services

CVX services are applications that run on top of the CVX infrastructure, and are accessed by CVX clients through the CVX server. All CVX services are maintained by version level; client switches negotiate the version they use when connecting to the server. This allows multiple switches that run different EOS versions to connect to the same CVX server.

The following sections briefly describe some of the services available to CVX clients through CVX:

- OpenStack Service
- VXLAN Control Service
- Hardware Switch Controller (HSC) Service
- Network Topology Service
- Static Topology Service

14.1.3.1 OpenStack Service

The OpenStack service on CVX allows the networking component of an OpenStack deployment (also known as Neutron) to share state with CVX.

When deployed, this integration allows CVX to send state about the logical networks created in the OpenStack cloud to the CVX clients that configure the network.

More information on OpenStack software can be found in its online documentation at http://docs.openstack.org/. .

14.1.3.2 VXLAN Control Service

The VXLAN control service allows hardware VXLAN Tunnel End Points (VTEPs) to share state with each other in order to establish VXLAN tunnels without the need for a multicast control plane. Configuration is required both on the client switches and in CVX.

14.1.3.3 Hardware Switch Controller (HSC) Service

Traffic between virtual machines which share a physical host (or between virtual machines and the rest of the network) is forwarded by virtual switches. The management and configuration of virtual switches uses the Open VSwitch DataBase (OVSDB) management protocol, as described in *RFC 7047*.

The Hardware Switch Controller (HSC) service provides an integration point between OVSDB controllers and the VXLAN control service, allowing exchange of state information among virtual and hardware switches.

14.1.3.4 Network Topology Service

The network topology service gathers information from CVX clients to provide a view of the physical topology of the network. Aggregated information gathered by the network topology service is used by other CVX services, and can be viewed on the CVX server.

14.1.3.5 Static Topology Service

Static Topology addresses cases where the deployment infrastructure in an OpenStack setup that manages Virtual Machines and Bare Metal servers does not enable LLDP on interfaces connecting hosts to switches. As a result, the topology information does not appear on CVX.

An example of this case is some deployments of OpenStack that do not enable LLDP for DPDK interfaces. Even with the manual configuration of LLDP on hypervisors, the configuration does not persist after OpenStack redeployment.

Static Topology enables the topology configuration statically using the **`service topology`** command on CVX without running LLDP on the servers connected to switches.

To view the aggregated topology information, use the **`show network physical-topology`** command on the switch running the CVX server instance.

14.1.4 Deploying CVX

CloudVision Exchange (CVX) can be deployed on KVM and ESXi. The required EOS version and Aboot version vary depending on whether you are deploying CVX on KVM or ESXi.

For the detailed steps to use to deploy CVX, see:

- Deploying CVX on Kernel-based Virtual Machine (KVM).
- Deploying CVX on VMware ESXi.

14.1.4.1 Deploying CVX on Kernel-based Virtual Machine (KVM)

Complete the following steps to install CVX on Ubuntu/KVM. Once the installation is complete, you can begin the CVX configuration process.



Note: Make sure you select versions of EOS and Aboot that meet the minimum requirements for CVX. The supported versions are:

- EOS (version ***4.16.8M*** or later).
- **`Aboot-veos-serial-8.0.0.iso`** (located in the vEOS section of the download).

Pre-requisites

Before you begin the procedure, make sure that:

- Install **`qemu-kvm`**, **`libvirt*`**, and all related dependencies using yum (RHEL7/CentOS7) and apt-get (Ubuntu).
- Two bridges are configured for use by the KVM VM, and that you have the names of the bridges. (Steps are included in the procedure to add bridges, if they are not already configured.)

Note: The bridges must be configured to persist (**brctl** commands do no persist across reboots). You can use Network Manager (or another application available to you) to complete this configuration.

- You have both **generateXmlForKvm.py** and **cvpTemplate.xml**. They are required to complete the procedure. You can find them in the CVP tarball for Ubuntu.

Complete the following steps to install CVX.

1. Download the Aboot and EOS files from: https://www.arista.com/en/support/software-download/.
2. Use **sudo su** to acquire superuser privileges, which are required to complete some of the installation steps.
3. Confirm that KVM is running on the server by entering the following command:

   virsh -c qemu:///system listAb

   The command output should match this example:

   ```
   Id    Name     State
   ------------------------------
     $
   ```

4. If the output does not look correct (previous step) go to for additional assistance: https://help.ubuntu.com/community/KVM/Installation.
5. Use the following command to convert the **vmdk** file to qcow2: **qemu-img convert EOS_4_16_8M.vmdk -O qcow2 EOS.qcow2**.

   Note: Step **6** and **7** are required if you do not already have 2 bridges defined in different subnets. If the bridges exist, go directly to step **8**.
6. Use ***brctl*** to add bridges for the KVM VM to use (br1 and br2 can be any names you choose).

   ```
   brctl addbr br1
   brctl addbr br2
   ```

   ifconfig can be used to identify Ethernet ports to be bridged. Once you identify the ports, add them to the bridges.

   Example

   brctl addif br1 enx803f5d086eae
7. Confirm that the bridges are up using **brctl show**.
   - Enter: **ifconfig br1 up**
   - And: **ifconfig br2 up**

     Note: The following step uses a number of input parameters (the number required vary depending on your server setup). To ensure the command executes successfully, we recommend that you type it into a scratch pad and edit as needed before typing it into the Linux Terminal.
   - Use the following command to generate **cvx.xml**, which will be used to setup the CVX VM.

     generateXmlForKvm.py

     Example

     ```
     python generateXmlForKvm.py -n cvx --device-bridge br1 --cluster-bridge
      br2 -e /usr/bin/kvm -i cvpTemplate.xml -c /home/myname/Downloads/Aboot-
     veos-serial-8.0.0.iso -x /home/myname/Downloads/EOS.qcow2 -b 8192 -p 2 -t

      -n cvx: VM name.
      --device-bridge br1: This is the name you gave the bridge - br1 or
      anything else.
      --cluster-bridge br2: Cluster bridge if clustering servers.
     ```

```
-i cvpTemplate.xml: Path to XML file input template.
-k: VM ID number used by virsh. If not entered, a random number is
assigned.
-b 8192: 8G of RAM.
-p 2: # of CPU cores.
-c: Path to Aboot file.
-x: Path to qcow2 file created in step 3.
-t: This parameter indicates the file defined by -x is for CVX.
-e â€˜/usr/bin/kvm: Ubuntu path to KVM.
(for RHEL KVM this is: -e â€˜usr/libexec/qemu-kvm)
-o: XML file used by virsh to define the KVM VM.
```

8. Run the following commands:

```
virsh define cvx.xml
virsh start cvx
virsh console cvx
```

9. (Optional) To configure CVX to start automatically, enter:

```
virsh autostart cvx
```

You are now ready to begin the CVX configuration (see CVX Configuration).

14.1.4.2 Deploying CVX on VMware ESXi

Complete the following steps to install CVX on ESXi. Once the installation is complete, you can begin the CVX configuration process.

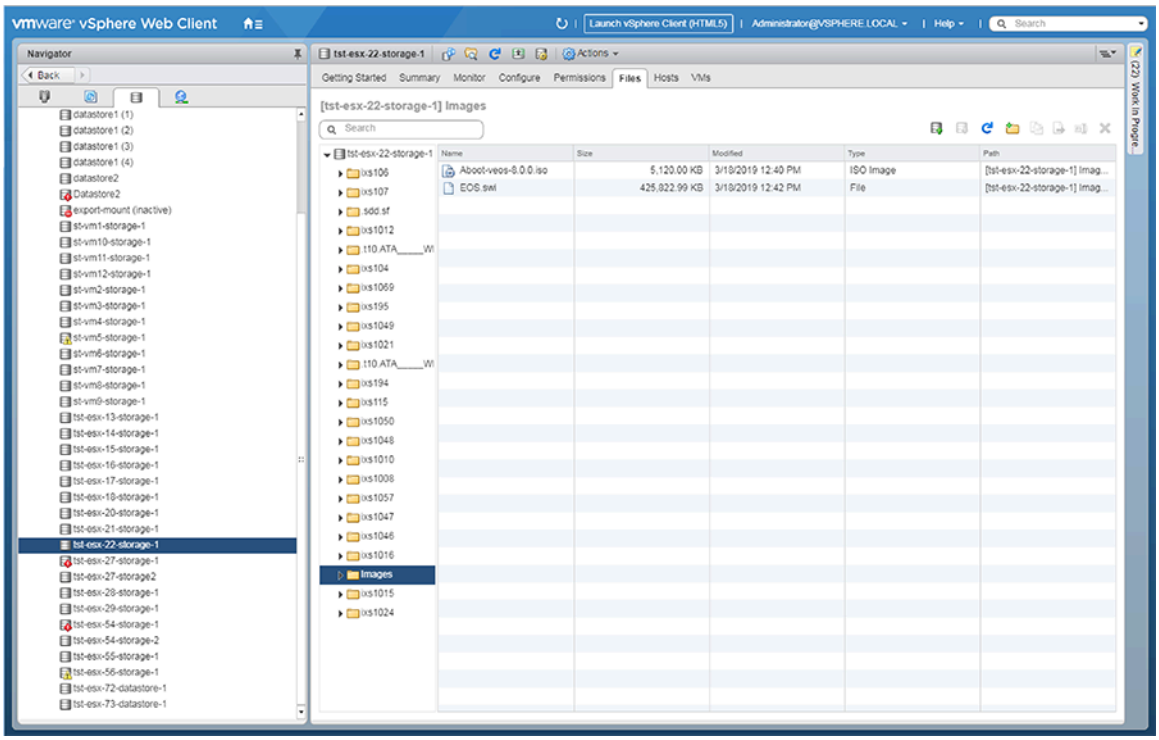
Note: Make sure you select versions of EOS that meet the minimum requirements for CVX. The supported version is EOS (version ***4.21.0*** or later).

Complete the following steps to install CVX.

1. Go to: https://www.arista.com.
2. Select **Support** > **Software Download**.
3. From the software download page, expand **Active Releases** > **4.21** > **EOS-4.21.0F** to download **`EOS-4.21.0F.vmdk`**.

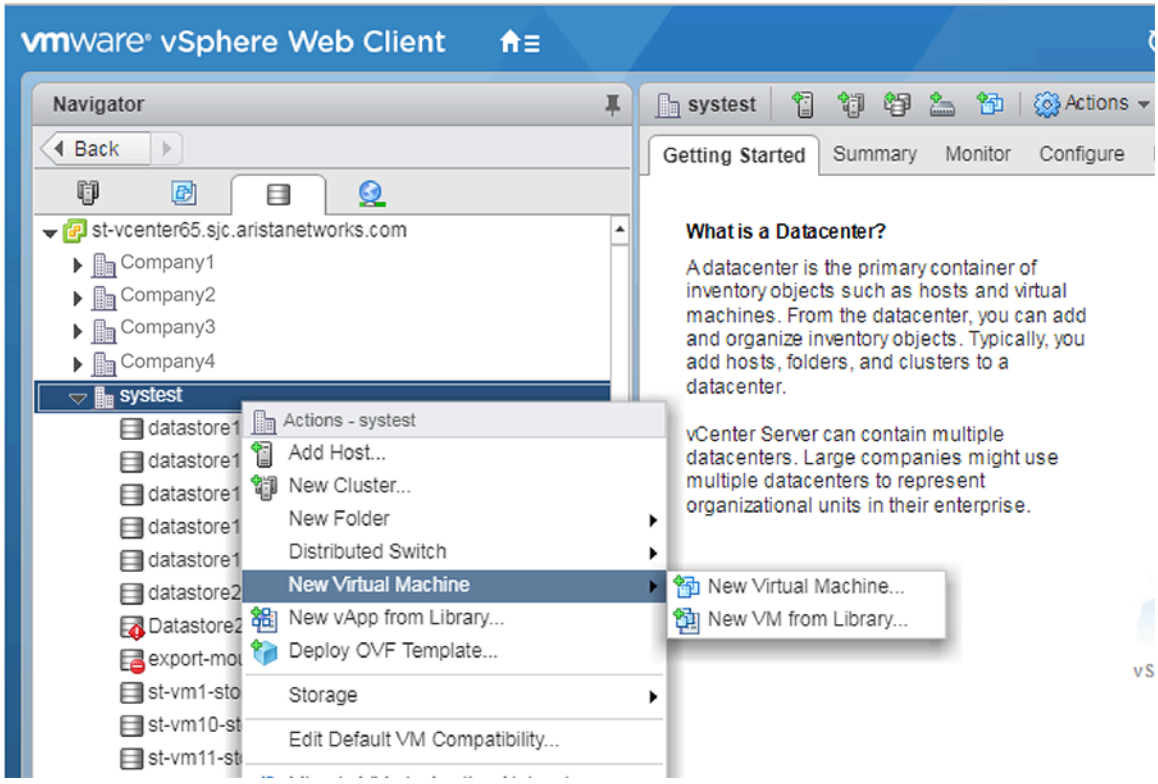
4. Load the files you downloaded into a filestore location within the VMware vSphere environment.

Figure 14-1: Loading the Files into the VMware vSphere Environment



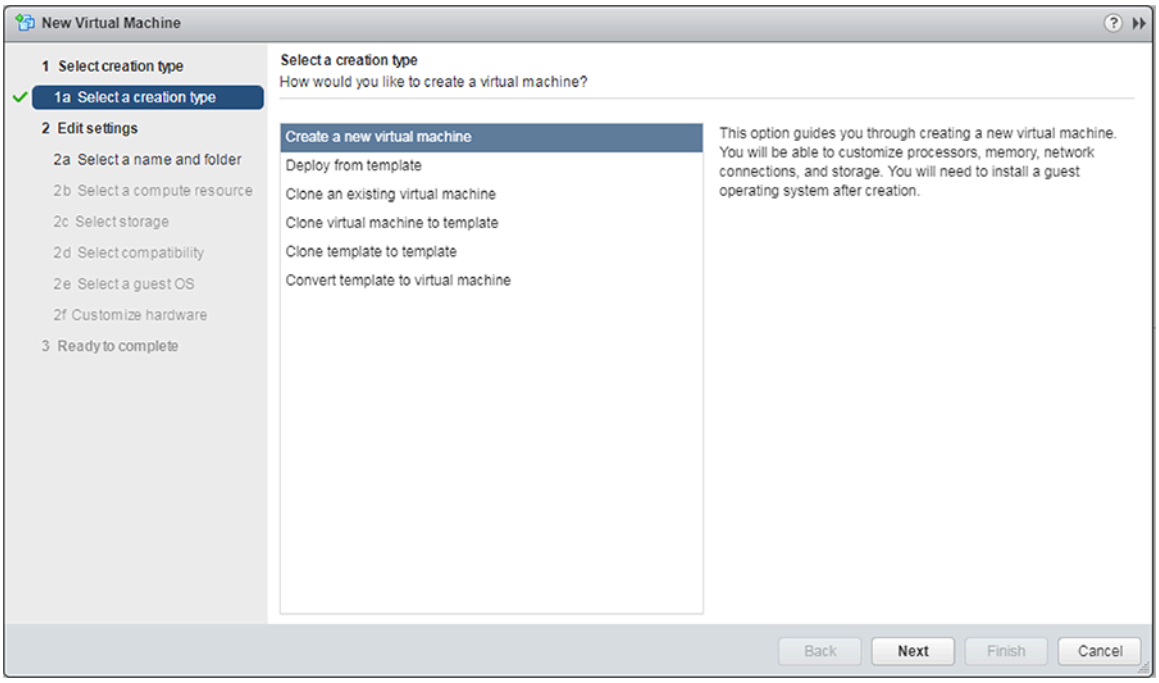
5. Right-click the **filestore location** you selected, and choose **New Virtual Machine**.

Figure 14-2: Selecting New Virtual Machine



The New Virtual Machine dialog appears.

Figure 14-3: New Virtual Machine Dialog

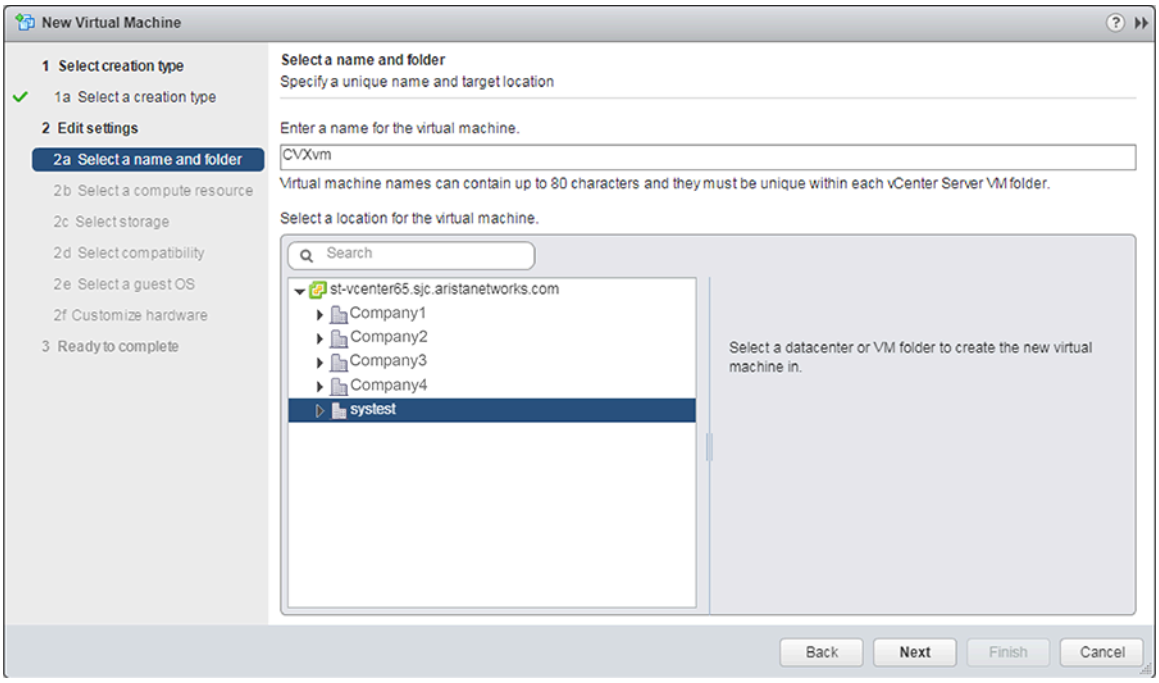


6. In the New Virtual Machine dialog, select **Create a new virtual machine**, then click **Next**.

   The dialog refreshes, showing options for the new Virtual Machine.

   New Virtual Machine dialog (naming and selecting the location).

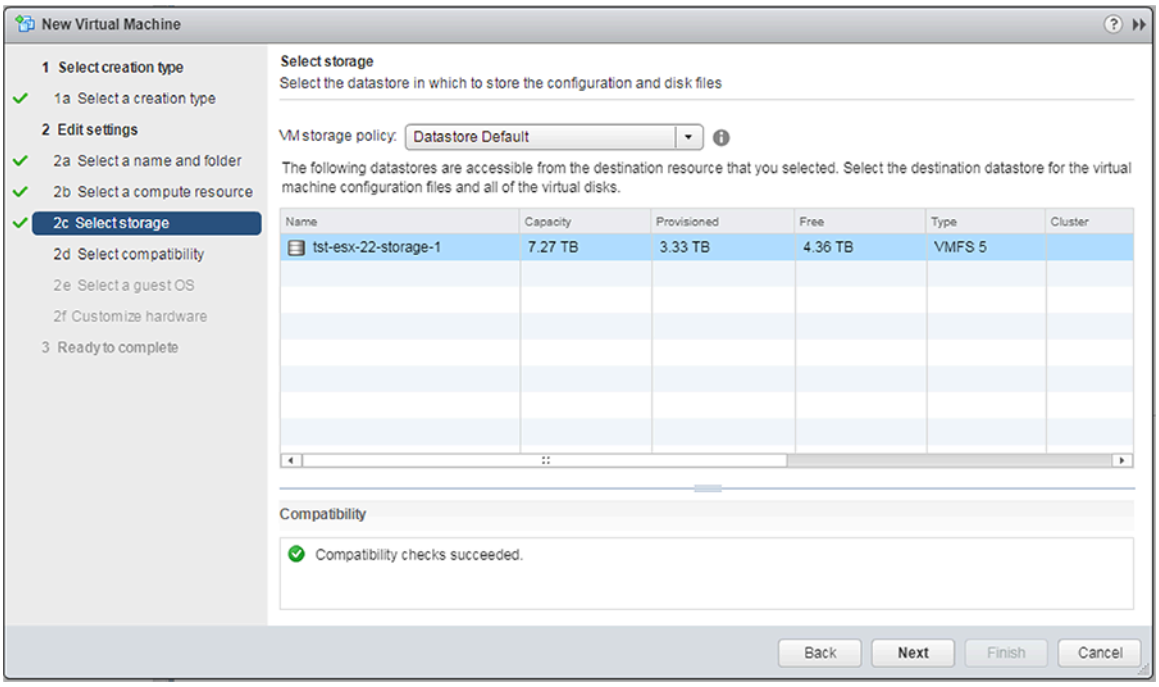
Figure 14-4: New Virtual Machine Dialog (Naming and Selecting the Location)



7. The dialog refreshes, showing options for selecting the datastore.
8. Enter a **name** for the new Virtual Machine.

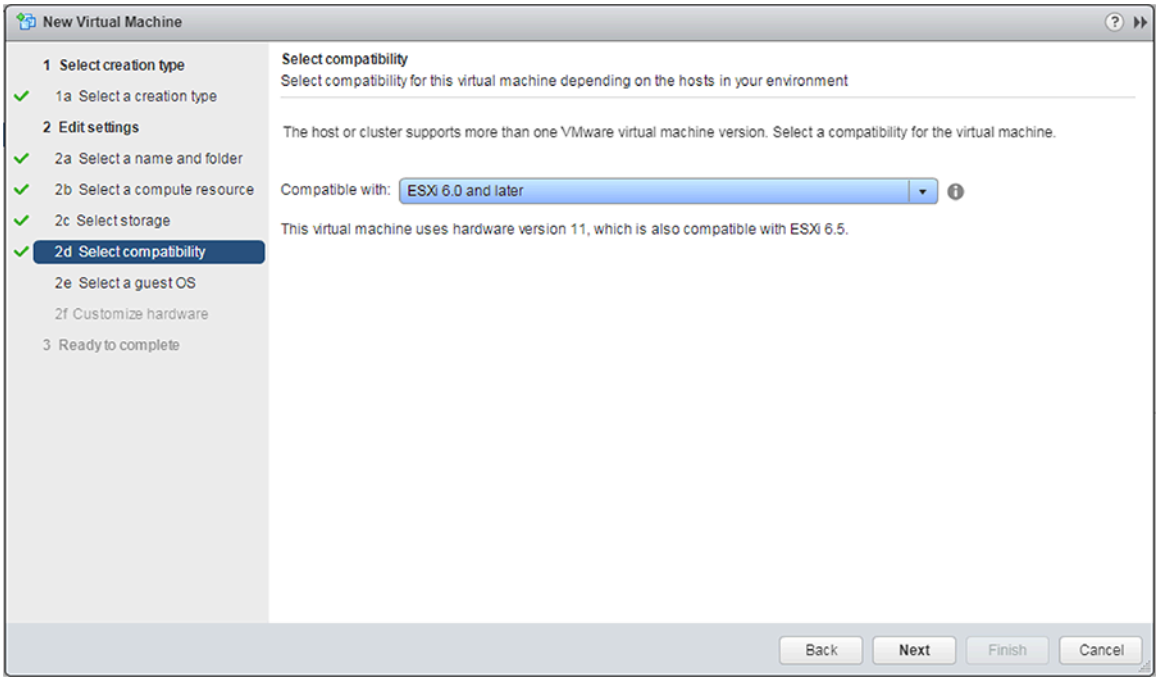
9. Select a **location** for the new Virtual Machine, then click **Next**.

Figure 14-5: New Virtual Machine Dialog (Selecting the Datastore)



10. Select the **datastore** for the new Virtual Machine configuration files and all of the virtual disks. Click **Next**. The dialog refreshes, showing operating system selection options.
11. Click **Next**. The dialog refreshes, showing compatibility options.

Figure 14-6: New Virtual Machine Dialog (Compatibility Options)



12. Using the Compatible with menu, select the **ESXi compatibility** for the new Virtual Machine.

Note: When adding the VMDK to ESX6, it treats this as sparse by default, whereas in ESX 5 it is thick. Converting the vEOS VMDK file from thin to thick would allow it to boot properly in ESX6:

`vmkfstools -i vEOS-lab-4.18.5M.vmdk -d eagerzeroedthick vEOS-lab-4.18.5M-thick.vmdk`.

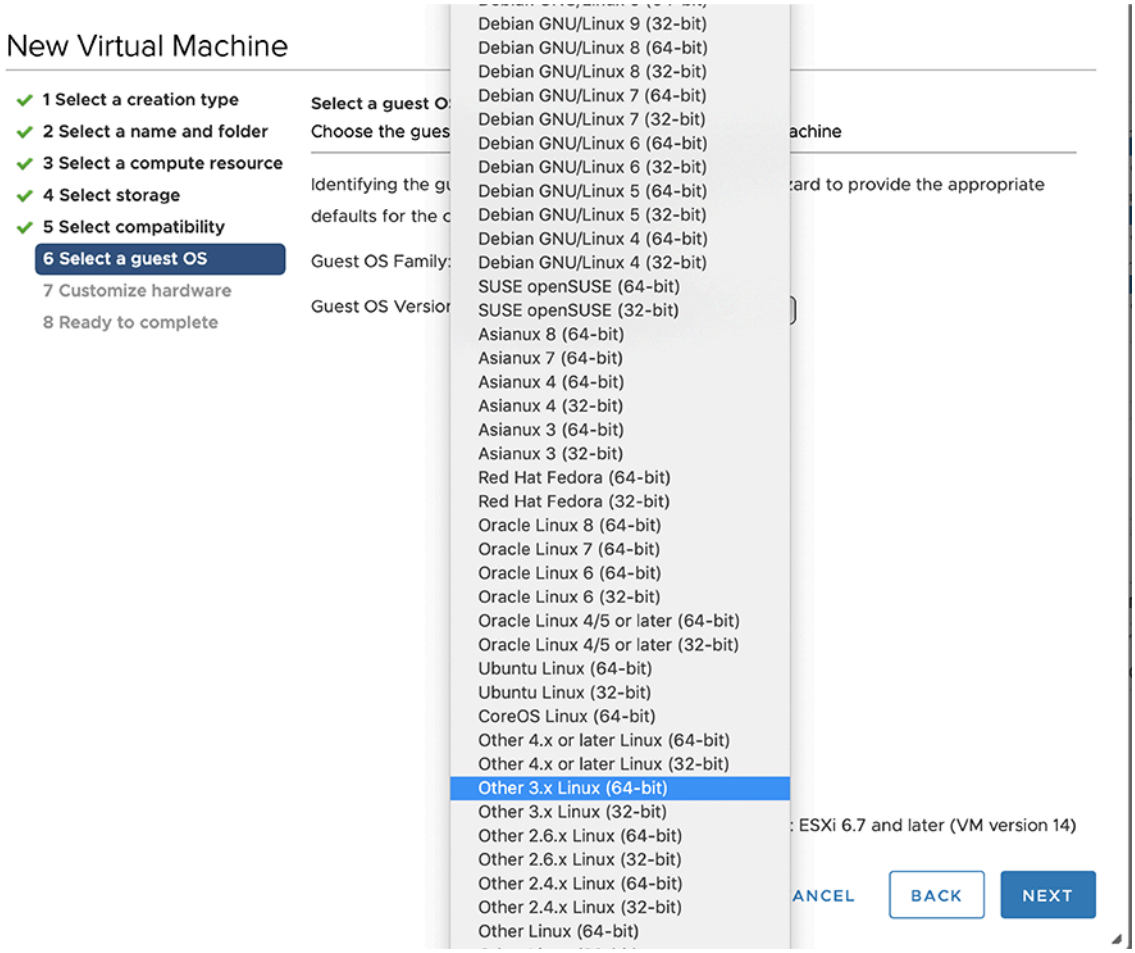
Go to https://arista.my.site.com/AristaCommunity/s/ and refer to the following topics for the issue and solution:

- Tip for Arista vEOS on VMware ESX 6..
- Common Issues When Deploying CVX.
- ***4.18.2F*** on vCenter 6 or 6.5.

Note: If the VM keeps rebooting and showing â€œThis is not a bootable disk. Insert a bootable floppy and press any key to try again", then go to https://arista.my.site.com/AristaCommunity/s/ and refer to the **Common Issues When Deploying CVX 4.18.2F on vCenter 6 or 6.5** topic.

13. Click **Next**. The dialog refreshes, showing operating system selection options.

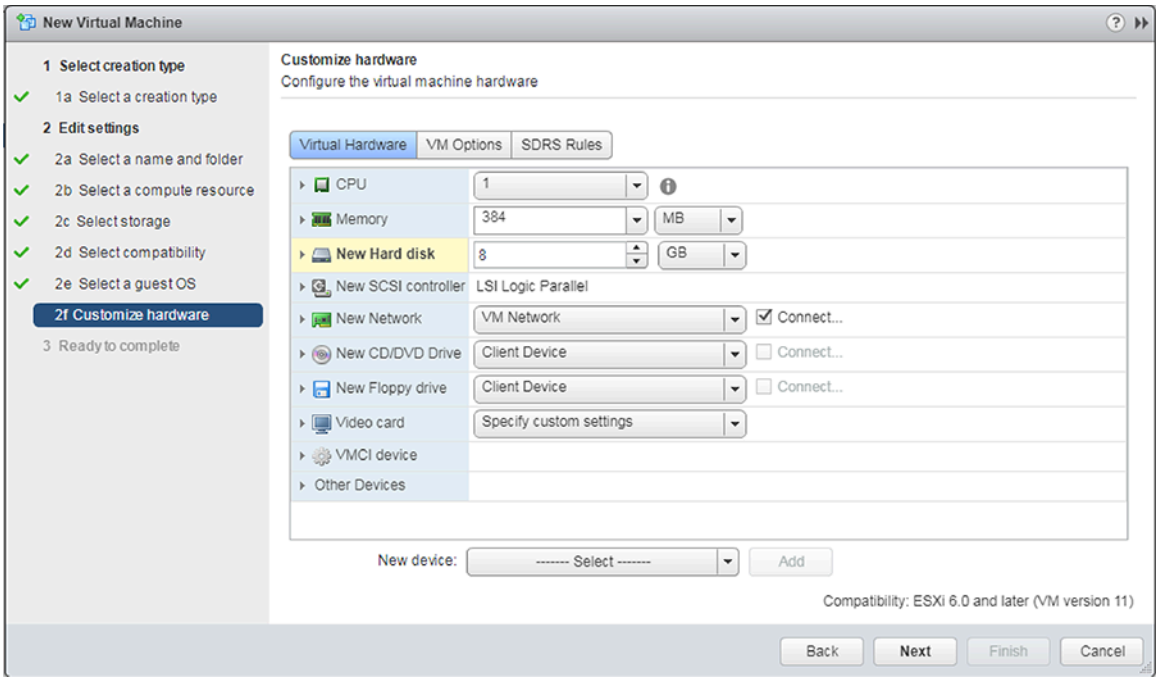
Figure 14-7: New Virtual Machine Dialog (Operating System Options)



14. Using the Guest OS Family menu, choose **Linux**.
15. Using the Guest OS Version menu, choose **Other Linux (64-Âbit)**.
16. Click **Next**.

The dialog refreshes, showing options for customizing hardware.

Figure 14-8: New Virtual Machine Dialog (Hardware Configuration Options)



17. Change the default settings for the following options:

CPU	Set to **4** (number of CPUs)
Memory	Set to **8 GB**
New Hard Disk	Delete the current setting (leave this option empty).
New Network	Specify connection to Network LAN segment with connectivity to CVX client devices (the Management LAN). Choose VMXNET3 network adapter type. This connection is used for CVX client / server communications.
Existing Hard Disk	Specify the **EOS-4.21.0F.vmdk** you downloaded in step 3.

18. (Optional) Delete the floppy drive and SCSI controller.
19. Click **Next**. You are now ready to begin the CVX configuration (seeCVX Configuration).

14.1.5 CVX Configuration

CVX, its clients, and its services, are independently configured. These sections describe configuration processes for each:

- Ports Used by CVX.
- CVX Server Configuration.
- CVX Client Configuration.
- CVX Client Services Configuration.

14.1.5.1 Ports Used by CVX

CVX uses the following ports:

- Controller database (Controllerdb): ***Port 9979.***
- Client-server out-of-band connection: ***Port 50003***.

- CVX cluster peer out-of-band connection: ***Port 50004.***

Note: All of these connections are TCP.

14.1.5.2 CVX Server Configuration

Enabling CVX on the CVX Server

CVX parameters for the server infrastructure are configured in ***CVX*** configuration mode. CVX configuration mode is not a group-change mode; ***running-config*** is changed when commands are entered, and exiting the mode does not modify ***running-config***. The **cvx** command places the switch in ***CVX*** configuration mode.

CVX is disabled by default. The **no shutdown (CVX)** command enables CVX on the switch.

Example

These commands enter CVX-configuration mode and enable CVX.

```
switch(config)# cvx
switch(config-cvx)# no shutdown
switch(config-cvx)#
```

CVX Heartbeat Configuration

CVX synchronizes with its client devices by exchanging heartbeat signals. The heartbeat transmission frequency and timeout period determine when a client's access to the server is disrupted.

The interval between heartbeat messages that the server transmits is specified by the **heartbeat-interval (CVX)** command. The CVX timeout period is specified by the **heartbeat-timeout (CVX)** command. When CVX does not receive a subsequent heartbeat message from a CVX client before the timeout expiry, the server discontinues CVX services to that client.

Best practices dictate that CVX and its client applications configure identical heartbeat interval and heartbeat timeout values.

Example

These commands configure a CVX heartbeat interval of 30 seconds and a server heartbeat timeout period of 90 seconds.

```
switch(config-cvx)# heartbeat-interval 30
switch(config-cvx)# heartbeat-timeout 90
switch(config-cvx)#
```

Disabling CVX on the CVX Server

Note: Before disabling or de-configuring CVX on the CVX server, CVX client services should be explicitly disabled or shut down. Failure to disable or de-configure services prior to disabling or de-configuring CVS may result in CVX features continuing to run after CVX shutdown.

When disabling the CVX service, service VXLAN configuration may be retained or erased. Be sure to disable or shut down client services prior to disabling the CVX service.

Examples

- These commands shut down the CVX service while retaining the CLI configuration for service VXLAN.

```
localhost(config)# cvx
localhost(config-cvx)# service vxlan
localhost(config-cvx-vxlan)# shutdown
```

- These commands shut down the CVX service and also erase service VXLAN CLI configuration.

```
localhost(config-cvx-vxlan)#
localhost(config)# cvx
localhost(config-cvx)# no service vxlan
```

14.1.5.3 CVX Client Configuration

This section describes the CVX client configuration and commands that enable CVX services. Most commands for the configuration of the CVX client infrastructure are accessed in Management-CVX configuration mode.

- **Enabling CVX on the CVX Client**

  CVX client parameters are configured in ***Management-CVX*** configuration mode. Management-CVX configuration mode is not a group-change mode; running-config is changed when commands are entered, and exiting the mode does not modify running-config. The **`management cvx`** command places the switch in ***Management-CVX*** configuration mode.

  CVX client is disabled by default. The **`no shutdown`** **(Management-CVX)** command enables CVX client on the switch.

  For the CVX network topology service to create an inventory of all CVX clients, ensure that LLDP is enabled on each client switch using the **`lldp run`** command.

  Example

  These commands enter ***Management-CVX-configuration*** mode and enable the CVX client.

```
switch(config)#lldp run
switch(config)#management cvx
switch(config-mgmt-cvx)#no shutdown
switch(config-mgmt-cvx)#
```

- **CVX Client Heartbeat Configuration**

  A CVX client synchronizes and maintains contact with CVX by exchanging heartbeat signals. The heartbeat transmission frequency and timeout period define when communication with CVX will be considered down.

  The interval between heartbeat messages that the CVX client transmits is configured by the **`heartbeat-interval`** **(Management-CVX)** command.

  The CVX client timeout period is specified by the **`heartbeat-timeout`** **(Management-CVX)** command. When a CVX client does not receive a subsequent heartbeat message from CVX within this timeout period, the client assumes that services provided by CVX are no longer available.

  Best practices dictate that a CVX client's heartbeat interval and heartbeat timeout values are identical to those of the CVX server to which it connects.

  Example

  This command configures a CVX client heartbeat interval of ***30*** seconds and client timeout period of ***90*** seconds.

```
switch(config-mgmt-cvx)# heartbeat-interval 30
switch(config-mgmt-cvx)# heartbeat-timeout 90
switch(config-mgmt-cvx)#
```

- **Connecting the CVX Client to a Server**

  The **`server host`** **(Management-CVX)** command identifies the location of the CVX server that the client accesses. The **`source-interface (Management-CVX)`** command specifies the interface from

which the client derives the IP address it uses as the source in CVX packets that it transmits. And the **no shutdown (Management-CVX)** command enables CVX on the client switch.

Example

These commands configure the switch as a CVX client, connecting to a CVX server at IP address ***10.1.1.14*** and using IP address ***10.24.24.1*** as the source address for its outbound packets.

```
switch(config)# interface loopback 5
switch(config-if-Lo5)# ip address 10.24.24.1/24
switch(config-if-Lo5)# management cvx
switch(config-mgmt-cvx)# server host 10.1.1.14
switch(config-mgmt-cvx)# source-interface loopback 5
switch(config-mgmt-cvx)# no shutdown
switch(config-mgmt-cvx)#
```

14.1.5.4 CVX Client Services Configuration

Switches running EOS must be configured as CVX clients to access the network services running on CVX. Individual services may require additional configuration.

Refer to the following for information regarding the services available to a CVX client.

- Configuring OpenStack Service.
- Configuring VXLAN Control Service.
- Configuring Hardware Switch Controller Service (HSC).
- Configuring Network Topology Service.
- Configuring Static Topology Service.

14.1.5.4.1 Configuring OpenStack Service

The OpenStack service is enabled from CVX-OpenStack configuration mode, which is accessed by the **service openstack** command. The **no shutdown (CVX-OpenStack)** command enables CVX OpenStack services on the CVX server. Additional configuration is necessary to deploy OpenStack (http://docs.openstack.org/).

Example

These commands enable the CVX-OpenStack service.

```
switch(config-cvx)# service openstack
switch(config-cvx-openstack)# no shutdown
switch(config-cvx-openstack)#
```

14.1.5.4.2 Configuring VXLAN Control Service

The VXLAN control service is enabled on CVX by the **no shutdown (CVX-VXLAN)** command and on the client switches by enabling CVX and configuring the VXLAN as a controller client. When VXLAN control service is enabled, CVX functions as a VXLAN controller for its clients.

For information about configuring VXLAN on the client switch, see the VXLAN chapter of the *User Manual* .

Examples

- These commands enable VXLAN control service on the CVX server.

```
switch(config-cvx)# service vxlan
switch(config-cvx-vxlan)# no shutdown
switch(config-cvx-vxlan)#
```

- These commands enable VXLAN Control Service on the CVX client. (This example assumes that the VXLAN has already been configured on the client switch. For information about configuring VXLAN, see the VXLAN chapter of the *User Manual*).

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan controller-client
```

14.1.5.4.3 Configuring Hardware Switch Controller Service (HSC)

Certificate Requirements for CVX Interoperability with VMware NSX 6.2.2 and Higher

The HSC service is enabled on the CVX server by the **`no shutdown` (CVX-HSC)** command.

The certificate type needs to be changed from MD5 to SHA512 for use with VMware NSX ***6.2.2***. Complete the following steps to make the change.

1. At the EOS prompt of CVX, use the following commands.

```
switch(config)# cvx
switch(config-cvx)# service hsc
switch(config-cvx-hsc)# shut
```

2. Acquire superuser privileges and edit the default.

```
switch(config)# bash
switch(config)# sudo su
switch(config)# vi /usr/bin/ovs-pki
```

3. Find and replace **`default_md`** with ***sha512*** (from **md5**).

```
default_md =md5
default_md =sha512
```

4. Delete all files and folders from **`/persist/secure/openvswitch/`**.

```
cd /persist/secure/openvswitch/bash-4.1#sudo rm -r *
```

5. Generate the new certificate.

```
[admin@CVX ~]$ exit
logout
CVX(config-cvx-hsc)#no shutdown
CVX(config-cvx-hsc)#end
```

6. Verify the change using the command.

```
CVX# show nsx status
```

Example

These commands enable the CVX-HSC service.

```
switch(config)# cvx
switch(config-cvx)# no shutdown
switch(config-cvx)# service hsc
switch(config-cvx-hsc)# no shutdown
```

The HSC service sends flood lists to each VTEP through CVX. Some controllers (such as VMware NSX's Service Nodes) implement replication nodes for head-end replication of unknown packets. For these controllers, BUM packets should be sent to a single replication node (send-to-any replication), and the flood list sent by the HSC service is a list of replication nodes. Other controllers (such as Nuage VSP)

require each VTEP to perform its own head-end replication. For these, BUM packets should be sent to every known VTEP, and the flood list sent by the HSC service is the list of VTEPs.

The default behavior is to use a send-to-any replication list of VTEPs. If the required behavior is send-to-all replication of, use the all option of the VTEP (CVX-HSC) command.

Example

This command configures the CVX-HSC service to connect to an OVSDB controller at IP address ***192.168.2.5***, using the default port ***6632***.

```
switch(config-cvx-hsc)# manager 192.163.2.5
switch(config-cvx-hsc)#
```

Example

This command configures the CVX-HSC service to use send-to-any replication.

```
switch(config-cvx-hsc)# vtep flood list type all
switch(config-cvx-hsc)#
```

Having established a connection to the OVSDB controller, the HSC service will publish the inventory of switches managed by CVX to OVSDB. For the inventory to succeed, LLDP must be enabled on each CVX client switch with the **`lldp run`** command.

Note: HSC also makes use of the VXLAN control service; ensure that VXLAN control service is enabled and properly configured (see VXLAN Control Service for details).

Note: LLDP is enabled by default on Arista switches.

Example

This command enables LLDP.

```
switch(config)# lldp run
switch(config)#
```

14.1.5.4.4 Configuring Network Topology Service

A network topology agent runs on each Arista switch whether or not the switch is connected to a CVX server. It requires no configuration. The network topology service on the CVX server is also enabled by default and requires no configuration.

To view the aggregated topology information, use the **`show network physical-topology`** command on the switch running the CVX server instance.

Examples

- This command displays all visible hosts.

```
switch# show network physical-topology hosts
Unique Id            Hostname
-------------------- ------------------------------
001c.7385.be69       cvx287.sjc.aristanetworks.com
0000.6401.0000       cvc1
0000.6402.0000       cvc2
0000.6403.0000       cvc3
0000.6404.0000       cvc4
bcf6.85bd.8050       dsj14-rack14-tor1
```

- This command displays all connections in the topology.

```
switch# show network physical-topology neighbors
cvx287.sjc.aristanetworks.com
Interface          Neighbor Intf      Neighbor Host
------------------ ------------------ --------------------
Ethernet1          Ethernet7          cvc4
Ethernet2          Ethernet7          cvc2
Ethernet9          Ethernet7          cvc1
Ethernet10         Ethernet7          cvc3
Management1        27                 dsj14-rack14-tor1

OUTPUT OMITTED FROM EXAMPLE
  dsj14-rack14-tor1

Interface     Neighbor Intf      Neighbor Host
------------- ------------------ ----------------------
27            Management1        cvx287.sjc.aristanetwork
```

14.1.5.4.5 Configuring Static Topology Service

Use the **service topology** command to configure the topology statically on CVX without running LLDP on the servers connected to switches. It is configured under CVX configuration mode.

Example

- The following command configures topology statically on the switch.

```
switch# config
switch(config)# cvx
switch(config-cvx)# service topology
```

- The topology information can be specified using the following command:

```
switch(config-cvx-topology)# network physical-topology switch SWITCH
 interface INTERFACE neighbor NEIGHBOR-HOST neighbor-interface NEIGHBOR-INTE
RFACE
```

The format of the hostname in this command could depend on services running on the host. As an example, in OpenStack use cases, it should match the hostname used by openstack services (For example, neutron server and agents) which is an FQDN format. The hostname in the command is case sensitive.

Optional Parameter

The neighbor interface is an optional parameter in the above configuration however setting it helps to understand the physical network connectivity between switches and hosts. It also helps in troubleshooting any issue that may arise in the network.

Limitations

To avoid mis-configuration in a topology consisting of a switch with a connected host, where LLDP is enabled and static topology is used to configure the physical topology it is recommended to use only one source of configuration, not both. As a mismatch in the configuration can cause wrong configuration on the switch by a feature consuming the topology information.

14.1.6 CVX Secure out-of-band Connection

This feature adds support for securing out-of-band connection between CVX server and CVX clients by SSL/TLS transport protocol. SSL/TLS is an application-layer protocol that provides secure transport between client and server through a combination of authentication, encryption and data integrity. SSL/TLS uses certificates and private-public key pairs to provide this security. We will use the term SSL to mean SSL/TLS.

By default, CVX server and CVX clients communicate over insecure transport (there is no authentication and encryption between CVX server and CVX clients). This poses the possibility of security risks, such as communicating with untrusted CVX server and CVX clients, or eavesdropping CVX server/client communications. This feature can be used to secure the out-of-band connection between CVX server and CVX clients.

Note: The CVX client-server out-of-band connection uses port ***50003***. The CVX cluster peer out-of-band connection uses port ***50004***. These are TCP ports.

For more information, see:Show Commands.

14.1.6.1 Configuring the CVX Secure out-of-band Connection

This feature uses SSL certificate and key management infrastructure for managing certificates, keys and SSL profiles. For more information regarding this infrastructure see *SSL Certificate and Key Management* in the *Arista User's Guide.*

1. On CVX server, copy the server certificate and key and also the CA certificate to verify CVX clients.

```
switch(config)# !Copy the PEM encoded certificate and RSA key files for CVX
 server
switch(config)# !Lets call them server.crt and server.key
switch(config)# copy <url> certificate:server.crt
switch(config)# copy <url> sslkey:server.key
switch(config)# !Copy the PEM encoded CA certificate to verify the
 certificate of CVX clients.Lets call it ca.crt
switch(config)# copy <url> certificate:ca.crt
```

2. On CVX server, configure SSL profile with the certificates and key as below. Lets call the SSL profile as ***serverssl***.

```
switch(config)# management security
switch(config-mgmt-security)# ssl profile serverssl
switch(config-mgmt-sec-ssl-profile-serverssl)# certificate server.crt key
 server.key
switch(config-mgmt-sec-ssl-profile-serverssl)# !You can trust multiple CA
 certificates
switch(config-mgmt-sec-ssl-profile-serverssl)# trust certificate ca.crt
```

Note: If you are using intermediate certificates to build a 'Chain of Trust' (such as **`server.crt -> intermediate1.crt -> intermediate2.crt -> ca.crt`**), then you need to configure the intermediate certificates as part of the SSL profile using the following commands:

```
switch(config-mgmt-sec-ssl-profile-serverssl)# chain certificate
 intermediate1.crt
switch(config-mgmt-sec-ssl-profile-serverssl)# chain certificate
 intermediate2.crt
```

3. On CVX server, configure to use the serverssl SSL profile. With this configuration, the CVX server starts listening on a secure port. The CVX server will continue to listen on the default port. i.e., the CVX server will accept connections from CVX clients over both SSL and default non-SSL transports. During a SSL negotiation, the CVX server will authenticate itself to the CVX clients by presenting server.crt and it verifies the authenticity of the CVX client by checking if the CVX client certificate is signed by the trusted certificate ca.crt.

```
switch(config)# cvx
switch(config-cvx)# ssl profile serverssl
```

4. On CVX client, copy the client certificate and key and also the CA certificate to verify CVX server.

```
switch(config)# !Copy PEM encoded certificate and RSA key files for CVX
 client
switch(config)# !Lets call them client.crt and client.key
switch(config)# copy <url> certificate:client.crt
switch(config)# copy <url> sslkey:client.key
switch(config)# !Copy PEM encoded CA certificate used to verify the
switch(config)# !certificate of CVX server. Lets call it ca.crt
switch(config)# copy <url> certificate:ca.crt
```



Note: If you are using intermediate certificates to build a 'Chain of Trust' (such as **client.crt -> intermediate1.crt -> intermediate2.crt -> ca.crt**), then you need to configure the intermediate certificates as part of the SSL profile using the following commands:

```
switch(config-mgmt-sec-ssl-profile-clientssl)# chain certificate
 intermediate1.crt
switch(config-mgmt-sec-ssl-profile-clientssl)# chain certificate
 intermediate2.crt
```

5. On CVX client, configure SSL profile with the certificates and key as below. Lets call the SSL profile as clientssl.

```
switch(config)# management security
switch(config-mgmt-security)# ssl profile clientssl
switch(config-mgmt-sec-ssl-profile-clientssl)# certificate client.crt key
 client.key
switch(config-mgmt-sec-ssl-profile-clientssl)# !You can trust multiple CA
 certificates
switch(config-mgmt-sec-ssl-profile-clientssl)# trust certificate ca.crt
```

6. On CVX client, configure to use the SSL profile clientssl. With this configuration, the CVX client will connect to the secure port of the CVX server over SSL transport. During SSL negotiation, the CVX client will authenticate itself to the CVX server by presenting client.crt and it verifies the authenticity of the CVX server by checking if the CVX server certificate is signed by the trusted certificate ca.crt.

```
switch(config)# management cvx
switch(config-mgmt-cvx)# ssl profile clientssl
```

14.1.6.2 Show Commands

For information regarding show commands of SSL certificate, key and profile, please refer to *SSL Certificate and Key Management*.

To show the SSL profile status on CVX server, use the **show cvx** command.

```
switch# show cvx

CVX Server
 Status: Enabled
 UUID: beb19142-dfaa-11e4-b996-001c73105347
 Heartbeat interval: 20.0
 Heartbeat timeout: 60.0
 SSL profile: serverssl
  Status: Enabled
```

The Enabled SSL status means that the SSL profile is enabled for CVX server and the CVX clients can connect to CVX server over SSL transport. If there are any errors, then the status will show Disabled and the reason will be listed. In Disabled state, the CVX clients wont be able to connect to CVX server over SSL transport.

To show the SSL connection status of CVX clients on CVX server, use the show cvx connections command.

```
switch# show cvx connections

Switch 00:1c:73:10:53:48
 Hostname: sq302
 Status: up
Last heartbeat sent: 0:00:04 ago
Last heartbeat received: 0:00:10 ago
Clock offset: -0.00201620385865
Out-of-band connection: SSL secured
In-band connection: Not secured (SSL not supported)
```

The out-of-band connection shows as SSL secured, which means that the CVX client has connected to CVX server over SSL transport. The in-band connection is another connection between CVX server and CVX client. The SSL is not yet supported for this connection and hence it shows as SSL not supported. There is already some level of protection for the in-band connection. The CVX server and CVX client opens up the access to in-band connection only if the out-of-band connection is successful. Since the out-of-band connection is configured to use SSL, the in-band connection access is granted only for authentic CVX client and CVX server.

To show SSL profile status and connection status on CVX client, use the show management cvx command.

```
switch# show management cvx

CVX Client
 Status: Enabled
 Last connected time: 2015-04-14 11:16:19
 Connection status: Connected
  Out-of-band connection: SSL secured
  In-band connection: Not secured (SSL not supported)
 Negotiated version: 2
 Controller UUID: 0e7dee2e-e2cf-11e4-880f-001c73105347
 Controller: 127.0.0.1
  Last heartbeat sent: 0:00:00 ago
  Last heartbeat received: never
  Clock offset: 0.0
 SSL profile: clientssl
  Status: Enabled
```

The Enabled SSL status means that the SSL profile is enabled and the CVX client can connect to CVX server over SSL transport. If there are any errors, then the status will show as Disabled and the reason will be listed. In Disabled state, the CVX client wont be able to connect to the CVX server.

Similar to the CVX server, the out-of-band connection shows as SSL secured and the SSL is not yet supported for in-band connection.

The possible reasons for Disabled SSL status on CVX server and CVX client are:

- **SSL profile does not exist:** If the SSL profile configured under CVX server/client is not configured under management security, you will see this message. Configure the SSL profile with required certificates and key under management security.
- Invalid SSL profile: If the SSL profile configured under CVX server/client is in invalid state, you will see this message. Check **`show management security ssl profile <name>`** command to see the errors on the SSL profile and fix them.
- **Trusted certificates not configured in SSL profile:** If the SSL profile configured under CVX server/client does not have trusted certificates configured, you will see this message. Configure trusted CA certificates in the SSL profile.
- **Certificate not configured in SSL profile:** If the SSL profile configured under CVX server/client does not have certificate key pair configured, you will see this message. Please configure certificate and key pair in the SSL profile.

Diffie-Hellman parameters not yet ready: When EOS is booted, a Diffie-Hellman parameters file is auto generated by the system if one does not exist. This Diffie-Hellman parameters file is used for symmetric key exchange during SSL negotiation. Only the CVX server uses this file and hence this message can be seen only on **`show cvx`** command output. If the file is not yet generated, you will see this message. When the file is ready, this message automatically goes away and the SSL profile will become enabled.

14.1.7 CVX High Availability

CVX provides high availability by enabling you to use multiple (redundant) CVX Controllers in the same cluster. Each Controller in the cluster has its own dedicated machine so that if a Controller fails, the failure is isolated to a single machine.

Within a cluster, one of the Controllers is a primary (leader), and the other Controllers are backup (follower) Controllers. If the primary Controller fails, one of the backup Controllers automatically assumes the role of the primary Controller.

CVX high availability does not prevent or compromise the detection of software failures or link failures that may cause Controllers to be unreachable on the network.

The configuration that is required to ensure CVX is set up for high availability involves:

- Configuring the CVX cluster.
- Configuring the CVX clients.

For more information, see:

- CVX Clusters.
- Handling of CVX Controller Failures.
- CVX Support for EOS Failure Modes.
- Client Interaction.
- Service Agents Interaction.
- Leader Election.
- Configuring CVX Clusters for High Availability.
- Configuring CVX Clients for High Availability.

14.1.7.1 CVX Clusters

CVX clusters are sets of CVX Controllers (usually 3 Controllers). Within a cluster, each Controller runs on its own dedicated machine, and all of the Controllers run the same version of CVX. Each Controller in the cluster functions as either the primary (leader) Controller, or a backup (follower) Controller.

One of the CVX Controllers is elected by the group of Controllers to be the primary Controller. Once a Controller is elected to be the primary, the other Controllers in the cluster are automatically assigned the role of backup Controllers. Cluster members maintain an out-of-band connection amongst themselves, which is used for the leader election protocol.

CVX Controllers in a cluster that are not the primary Controller always function as backup Controllers. Within the same cluster, only one CVX Controller can assume the role of a primary at any time.

For more information, see:

- Required Number of Controllers to Support High Availability
- Cluster Configuration Options

14.1.7.1.1 Required Number of Controllers to Support High Availability

A cluster must have enough Controllers so that in the case of a failure of the primary Controller, there are enough remaining Controllers for the election process to be completed. The election process is used by clusters to select a new primary Controller in the case of failure.

Note: The number of Controllers for a cluster is **3** (one primary and two backup Controllers).

Examples

In a cluster with only **two** Controllers (one primary and one backup), a simple majority of backup Controllers does not exist after a failure of the primary Controller. A simple majority of two backup Controllers is required for the leader election process.

14.1.7.1.2 Cluster Configuration Options

You can configure the cluster for high availability using either of the following modes:

- Cold followers mode - Only the Controllerdb of the primary (leader) CVX Controller mounts from the client switches.
- Warm followers mode - The Controllerdb of every (all) CVX Controllers in the cluster mount from the client switches.

Advantages and Disadvantages of the Modes

The advantage of the warm follower mode is that if the primary CVX Controller fails, the switchover to the new primary is faster than a switchover in cold follower mode. The reason for this is that the state of the new primary does not have to be rebuilt from scratch. The disadvantage of the warm follower mode is that serialization from the switch is slower compared to cold follower mode.

14.1.7.2 Handling of CVX Controller Failures

CVX Controllers can fail because of hardware or software faults. Because EOS agents are designed to be software fault-tolerant, an agent that fails is automatically restarted and resumes operation statefully. The most recent saved state in Sysdb for the agent is used to restore the state of the agent.

Unlike software failures, hardware failures are not handled by EOS. CVX handles hardware failures through the use of redundant backup (follower) CVX Controllers that run on their own dedicated machine. Within a cluster, any backup Controller can assume the role of the primary (leader) Controller.

Note: In the event of a network partition, the partition with a majority of the Controllers elects a leader from its Controllers, and the minority partition relinquishes any leadership it might have had.

14.1.7.3 CVX Support for EOS Failure Modes

CVX supports both EOS failure modes that apply when a CVX Controller fails. The EOS failure modes are:

- Fail-stop
- Fail-recover

Because CVX supports both EOS failure modes, a failed CVX Controller can rejoin the cluster if the following failures occur:

- A crash of the agent or machine running CVX.
- The CVX controller or dedicated machine it runs on is removed (partitioned) from the cluster.

14.1.7.4 Client Interaction

Client switches maintain an out-of-band connection to all members of the cluster. The connection is used to determine liveness and for communications. The connection is also used to signal a change in leadership (switchover) to the client switches. Switchovers that are changes in leadership within a cluster are executed similarly to CVX Graceful Reboot switchovers.

The ControllerClient agent on the switch is responsible for maintaining liveness with the Controllers and for exchanging metadata. The ControllerClient agent registers with all cluster members. Each Controller's ControllerStatus has an additional flag to record whether the Controller is a leader within the cluster.

If there is more than one leader, the switch automatically waits until only one Controller is designated as the leader in the cluster. Once a single Controller is designated as the leader, the switch executes a graceful switchover to the new leader Controller.

14.1.7.5 Service Agents Interaction

One change to Service Agents is required to support CVX high availability. Service Agents must be modified to include the leader flag (this flag identifies the leader CVX Controller in the cluster). On a leader switchover, Service Agents are deactivated on the old leader Controller and activated on the new leader Controller. The client switches will perform a graceful switchover to the new leader Controller.

14.1.7.6 Leader Election

Leader election is an internal, system-run process that is essential to CVX high availability. The leader election process is used to safely elect a new leader Controller within a cluster following the failure of the current leader Controller, or a network configuration change that results in the loss of the current leader Controller in the cluster.

The leader election process is designed to ensure stability of leader Controllers within clusters. The process is based on an algorithm that provides the mechanism for the backup (follower) Controllers to elect (by consensus), the new leader Controller in the cluster.

14.1.7.7 Configuring CVX Clusters for High Availability

Configuring CVX clusters for high availability is a simple process that involves pointing each cluster member to the other cluster members using the peer host command. The objective of this task is to successfully register each cluster member with the other cluster members. Successful registration of the cluster members with each other ensures that the members can communicate with each other to elect a new leader member if the original leader member fails.

Once you complete the process, the cluster members will be successfully registered with each other. In addition, the cluster members will automatically elect a leader member and assign the leader to that member. The non-leader members are automatically assigned the role of follower.

Requirements

The requirements for setting up clusters for high availability are:

- The number of CVX Controllers in a cluster is ***3***.
- An **odd number** of CVX instances (CVX Controllers) are required to form a cluster.

  Note: If an even number of CVX Controllers are configured in a cluster, a CVX instance will automatically refuse to participate in the cluster.
- All cluster members must point to each other. This is essential for clusters to operate normally. (The steps required to complete this task are included in the following procedure.)

Procedure

Note: This procedure provides configuration examples for each step. The example cluster used throughout the procedure contains 3 cluster members (named ***cvs1***, ***cvs2***, and ***cvs3***). The IP addresses of the cluster members are:

- ***cvs1*** (***10.0.0.1***)
- ***cvs2*** (***10.0.0.2***)
- ***cvs3*** (***10.0.0.3***).

Complete the following steps to configure clusters for high availability.

1. Using the peer host command, configure one of the cluster members to point to every other cluster member. This example shows the configuration of cluster member ***cvs1*** to point to the other cluster members (***cvs2*** and ***cvs3***).

```
cvs1(config-cvx)# peer host 10.0.0.2 (connects cvs1 to cvs2)
cvs1(config-cvx)#peer host 10.0.0.3 (connects cvs1 to cvs3)
```

2. Use the **show cvx** command to check the **Mode** and **Peer registration state** status values for cluster member ***cvs1***. The status values should be:
   - **Mode** = *Cluster*
   - **Peer registration state** = *Connecting*

   Note: Mode automatically changes from Standalone to Cluster when configuring a CVX cluster. This is because the presence of multiple CVX peers causes the Mode to change to Cluster. **Peer registration state** remains in Connecting status after you configure the first cluster member. This is because the two peers must register with each other for the registration of the two members to be successful.
3. Using the peer host command, configure peer cluster member ***cvs2*** to point to every other cluster member. This example shows the configuration of cluster member ***cvs2*** to point to the other cluster members (***cvs1*** and ***cvs3***).

```
cvs2(config-cvx)# peer host 10.0.0.1 (connects cvs2 to cvs1)
cvs2(config-cvx)# peer host 10.0.0.3 (connects cvs2 to cvs3)
```

4. Use the **show cvx** command to check the **Peer registration state** settings for ***cvs1***. This is done to verify that peers ***cvs1*** and ***cvs2*** are successfully registered with each other.

```
cvs1(config-cvx)# show cvx
```

Example

This example shows the output of the **show cvx** command for ***cvs1***. The **Peer registration state** setting of Registration Complete for peer ***cvs2*** *indicates a successful registration between* ***cvs1*** *and* ***cvs2***.

```
cvs1(config-cvx)#show cvx

CVX Server
 Status: Enabled
 UUID: 6c208fba-7324-11e5-8fef-1d98cdd3b27a
 Mode: Cluster
 Heartbeat interval: 20.0
 Heartbeat timeout: 60.0
 Cluster Status
  Name: default
  Role: Standby
  Leader: 10.0.0.2
  Peer timeout: 10.0
  Last leader switchover timestamp: 0:00:03 ago
  Peer Status for 10.0.0.3
   Peer registration state: Connecting
   Peer service version compatibility : Version mismatch
  Peer Status for 10.0.0.2
   Peer Id : 02-01-63-02-00-00
   Peer registration state: Registration complete
   Peer service version compatibility : Version ok
```

5. Using the peer host command, configure peer cluster member ***cvs3*** to point to every other cluster member. This example shows the configuration of cluster member ***cvs3*** to point to the other cluster members (***cvs1*** and ***cvs2***).

```
cvs3(config-cvx)# peer host 10.0.0.1 (connects cvs3 to cvs1)
cvs3(config-cvx)# peer host 10.0.0.2 (connects cvs3 to cvs2)
```

6. Use the **show cvx** command to check the **Peer registration state** settings for ***cvs1***. This is done to verify that peers ***cvs1*** and ***cvs3*** are successfully registered with each other.

```
cvs1(config-cvx)# show cvx
```

Example

This example shows the output of the **show cvx** command for *cvs1*. The **Peer registration state** setting of Registration Complete for peer ***cvs3*** *indicates a successful registration between* ***cvs1*** and ***cvs3***.

```
cvs1(config-cvx)# sh cvx

CVX Server
 Status: Enabled
 UUID: 6c208fba-7324-11e5-8fef-1d98cdd3b27a
 Mode: Cluster
 Heartbeat interval: 20.0
 Heartbeat timeout: 60.0
 Cluster Status
  Name: default
  Role: Standby
  Leader: 10.0.0.2
  Peer timeout: 10.0
  Last leader switchover timestamp: 0:05:37 ago
  Peer Status for 10.0.0.3
   Peer Id : 02-01-63-03-00-00
   Peer registration state: Registration complete
   Peer service version compatibility : Version ok
  Peer Status for 10.0.0.2
   Peer Id : 02-01-63-02-00-00
   Peer registration state: Registration complete
   Peer service version compatibility : Version ok
```

Next Step

You are now ready to configure the CVX clients for high availability (see Configuring CVX Clients for High Availability).

14.1.7.8 Configuring CVX Clients for High Availability

Configuring CVX clients for high availability is a simple process that involves pointing each CVX client to every CVX cluster member using the server host command. The objective of this task is to successfully establish connections between each CVX client and every CVX cluster member. The connections are essential to ensure that the CVX clients are aware of the current status of each cluster member.

Note: If a CVX client is not pointing to every cluster member, or if it is pointing to a CVX instance (Controller) that is not part of the cluster, the client may not be aware of leadership changes in the cluster, or may become confused about which cluster member is currently the leader. Either of these scenarios can result in unexpected errors.

Once you complete the process, the CVX clients will have established connections with each cluster member (the Connection status for each Controller should be Established). In addition, the clients will be aware of which CVX instance (Controller) is currently the leader in the cluster.

Procedure

Note: This procedure provides configuration examples for each step. The example CVX client used throughout the procedure is named cvc1. The IP addresses of the cluster members are: ***10.0.0.1*** (***cvs1***), ***10.0.0.2*** (***cvs2***), and ***10.0.0.3*** (***cvs3***).

Complete the following steps to configure CVX clients for high availability.

1. Using the server host command, configure each of the CVX clients to point to every cluster member. This example shows the configuration of client ***cvc1*** to point to all of the cluster members (the addresses of the cluster members are ***10.0.0.1***, ***10.0.0.2***, and ***10.0.0.3***).

```
cvc1(config-mgmt-cvx)# server host 10.0.0.1 (connects cvc1 to cluster member
 10.0.0.1)
cvc1(config-mgmt-cvx)# server host 10.0.0.2 (connects cvc1 to cluster member
 10.0.0.2)
cvc1(config-mgmt-cvx)# server host 10.0.0.3 (connects cvc1 to cluster member
 10.0.0.3)
```

2. Use the **show man cvx** command to check the status of client ***cvc1***. The Connection status for each cluster member should be Established. In addition, the client is also aware that cluster member ***10.0.0.3*** is the current Master.

```
cvc1(config-mgmt-cvx)#show man cvx

CVX Client
 Status: Enabled
 Source interface: Inactive (Not configured)
 Controller cluster name: default
  Controller status for 10.0.0.1
   Connection status: established
    Out-of-band connection: Not secured
    In-band connection: Not secured (SSL not supported)
   Negotiated version: 2
   Controller UUID: 6c208fba-7324-11e5-8fef-1d98cdd3b27a
   Last heartbeat sent: 0:00:07 ago
   Last heartbeat received: 0:00:07 ago
 Controller status for 10.0.0.3
  Master since 0:03:34 ago
  Connection status: established
   Out-of-band connection: Not secured
   In-band connection: Not secured (SSL not supported)
 Negotiated version: 2
 Controller UUID: c64954b8-7324-11e5-9f33-51f8b016cae8
 Last heartbeat sent: 0:00:14 ago
 Last heartbeat received: 0:00:14 ago
Controller status for 10.0.0.2
 Connection status: established
  Out-of-band connection: Not secured
  In-band connection: Not secured (SSL not supported)
 Negotiated version: 2
 Controller UUID: 6a0dbf2c-7324-11e5-94f3-ff17a8a1cdc8
 Last heartbeat sent: 0:00:05 ago
 Last heartbeat received: 0:00:05 ago
```

14.1.8 CVX VIP

CVX VIP provides the virtual IP address that actively follows the master controller of the CVX cluster.

The virtual IP address of the CVX HA Cluster is configured on a macvlan interface setup on top of a physical management interface of the master controller. The virtual IP and virtual MAC needs to be provided by the customer as part of the controller configuration. This information is available to all controllers as each cluster member has to be configured manually by the user on all controllers.

The macvlan interface created should be designated as `***Management0***`. `***Management0***` is currently used for the ManagementActive interface on modular switches. Without explicit configuration of VIP and VMAC, CVX VIP functionality will not work in the CVX HA cluster.

Customers can pick the VMAC from a pool of MAC addresses reserved for use with CVX clusters. The OUI pool, 00:1C:73:00:00:AA " 00:1C:73:00:00:FF has been reserved for this purpose.

The macvlan interface is setup if all of the following conditions are met:

- VMAC is configured by the user
- The controller instance is a leader
- There are more than one controller instances
- The controller is not being run on a modular system

- CVX VIP
- Data Replication
- SSH Host Key Tagging

14.1.8.1 Configuring VIP

All CLI commands applicable to the management interface of the controller will be allowed on ***Management0***, with the exception of Layer 1 / phy level commands. So auto-negotiation or flow control cannot be configured on the ***Management0*** interface. Instead these commands can only be run on the physical management interfaces. This makes sense as the phy-level configuration really depends on what the interface is physically wire.

To configure VMAC/VIP :

```
CVX(config)# interface management 0
CVX(config-if-Ma0)# mac-address 00:1C:72:00:00:FF
CVX(config-if-Ma0)# ip address 10.0.0.2
```

14.1.8.2 Data Replication

At EOS boot time, SSH host keys and Diffie-Hellman parameters are automatically generated and persistently stored on each controller. Multiple SSL profiles / keys / certificates might also be created and used by various agents on the controllers. Since these information contribute to the identity of the master, they will need to follow the master controller for all time.

In case of a controller switchover, the newly elected master controller will need to use the same SSH host keys & SSL profiles / keys / certificates to retain its identity and prevent any kind of network security alarms from being tripped. For example, if an SSH client notices that the host key has changed, it will normally flag an error warning the user of a possible man-in-the-middle type attack. Hence, this data will be replicated from the master to slaves.

14.1.8.3 SSH Host Key Tagging

SSH host keys are tagged with the chassis MAC address to deal with key regeneration issues when a supervisor module is moved from one chassis to another. This behavior will cause regeneration issues if we replicate the SSH host keys across the cluster resulting in the key fingerprint seen by management tools to be different.

To mitigate this, in addition to the chassis MAC address, the host keys would now be tagged with VMAC of the CVX HA cluster. If CVX VIP and VMAC are configured, SshHostKeysAgent will not regenerate keys if tagged VMAC and configured VMAC are the same, even if there is a mismatch between the chassis MAC and tagged MAC.

14.1.9 CVX Commands

CVX Server Commands

- cvx
- heartbeat-interval (CVX)
- heartbeat-timeout (CVX)
- port (CVX)
- show cvx
- shutdown (CVX)

CVX Client Commands

- management cvx
- heartbeat-interval (Management-CVX)
- heartbeat-timeout (Management-CVX)
- server host (Management-CVX)
- source-interface (Management-CVX)
- shutdown (Management-CVX)

CVX OpenStack Commands

- name-resolution force (CVX-OpenStack)
- name-resolution interval (CVX-OpenStack)
- service openstack
- shutdown (CVX-OpenStack)

CVX VXLAN Control Service Commands

- resync-period
- service vxlan
- shutdown (CVX-VXLAN)
- vtep (CVX-VXLAN)

CVX Hardware Switch Controller (HSC) Commands

- manager
- ovsdb-shutdown
- service hsc
- shutdown (CVX-HSC)
- vtep (CVX-HSC)

CVX Network Topology Service Commands

- lldp run

CVX Static Topology Service Commands

- service topology
- show network physical-topology

14.1.9.1 cvx

CVX (CloudVision eXtension) aggregates and shares status across a network of physical switches running EOS. CVX services provide visibility and coordinate activities across a network of switches that are configured as CVX clients.

The **cvx** command enters CVX configuration mode. CVX configuration mode is not a group-change mode; ***running-config*** is changed immediately upon entering commands. Exiting CVX configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

The **no cvx** and **default cvx** commands restore all CVX server defaults by deleting all CVX configuration mode statements from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

cvx

no cvx

default cvx

Commands Available in CVX Configuration Mode

- **port (CVX)**
- **service openstack**
- **service vxlan**
- **shutdown (CVX)**
- **heartbeat-interval (CVX)**
- **heartbeat-timeout (CVX)**

Example

These commands enter the CVX-configuration mode and displays the CVX configuration.

```
switch(config)# cvx
switch(config-cvx)# show active all

 cvx
  shutdown
  port 9979
  heartbeat-interval 20
  heartbeat-timeout 60
  no service vxlan
  service openstack
   shutdown
   name-resolution interval 21600
switch(config-cvx)#
```

14.1.9.2 heartbeat-interval (CVX)

The **heartbeat-interval** command configures the interval between heartbeat messages that the switch sends as a CVX server. Heartbeat messages are part of the keepalive mechanism between CVX and the CVX clients to which it connects.

The **no heartbeat-interval** and **default heartbeat-interval** commands restore the heartbeat interval to the default setting by removing the **heartbeat-interval** command from *running-config*.

Command Mode

CVX Configuration

Command Syntax

heartbeat-interval period

no heartbeat-interval

default heartbeat-interval

Parameters

period Interval duration (seconds). Value ranges from ***5*** through ***60***. Default value is ***20***.

Related Commands

- **cvx**.
- **heartbeat-timeout (CVX)**

Guidelines

Heartbeat messages flow independently in both directions between CVX and clients. When a client stops receiving heartbeat messages from the server within a specified period, the client assumes that the CVX server is no longer functioning.

Best practices dictate that CVX and its client applications configure identical heartbeat interval values.

Example

This command configures a CVX server heartbeat interval of ***30*** seconds:

```
switch(config)# cvx
switch(config-cvx)# heartbeat-interval 30
switch(config-cvx)#
```

14.1.9.3 heartbeat-interval (Management-CVX)

The **heartbeat-interval** command configures the interval between heartbeat messages that the switch sends as a CVX client. Heartbeat messages are part of the keepalive mechanism between the CVX client and the CVX server to which it connects.

The **no heartbeat-interval** and **default heartbeat-interval** commands revert the heartbeat interval to the default setting by removing the **heartbeat-interval** command from ***running-config***.

Command Mode

Mgmt-CVX Configuration

Command Syntax

heartbeat-interval period

no heartbeat-interval

default heartbeat-interval

Parameters

period: Interval duration (seconds). Value ranges from **5** through **60**. Default value is **20**.

GuidelinesHeartbeat messages flow independently in both directions between CVX and clients. When the server stops receiving heartbeat messages from a client within a specified period, the server assumes that the device it is no longer functioning as a CVX client.

Best practices dictate that the CVX client's heartbeat interval value is identical to that of its CVX server.

Related Commands

heartbeat-timeout (Management-CVX) specifies the CVX client timeout interval.

Example

These commands configure a CVX client heartbeat interval of ***30*** seconds:

```
switch(config)# management cvx
switch(config-mgmt-cvx)# heartbeat-interval 30
switch(config-mgmt-cvx)#
```

14.1.9.4 heartbeat-timeout (CVX)

The **heartbeat-timeout** command specifies the CVX timeout period. When a CVX server does not receive consecutive heartbeat messages from a CVX client within the heartbeat timeout period, the server discontinues providing CVX services to the client device. The default timeout period is ***60*** seconds.

The **no heartbeat-timeout** and **default heartbeat-timeout-timeout** commands restore the heartbeat timeout to the default setting by removing the heartbeat-timeout command from ***running-config***.

Command Mode

CVX Configuration

Command Syntax

heartbeat-timeout period

no heartbeat-timeout

default heartbeat-timeout

Related Commands

- cvx places the switch in CVX configuration mode.
- heartbeat-interval (CVX) specifies the CVX heartbeat interval.

Parameters

period heartbeat timeout interval (seconds). Value ranges from ***15*** to ***10800***. Default value is ***60***.

Guidelines

Best practices dictate that CVX and its client applications configure identical heartbeat timeout values.

Examples

These commands set the CVX timeout period to ***90*** seconds.

```
switch(config)# cvx
switch(config-cvx)# heartbeat-timeout 90
switch(config-cvx)#
```

14.1.9.5 heartbeat-timeout (Management-CVX)

The **heartbeat-timeout** command specifies the CVX client timeout period. When a CVX client does not receive consecutive heartbeat messages from a CVX server within the period specified by this command, the client assumes that its connection to CVX is disrupted. The default timeout period is ***60*** seconds.

The **no heartbeat-timeout** and **default heartbeat-timeout** commands restore the CVX client heartbeat timeout to the default setting by removing the **heartbeat-timeout** command from ***running-config***.

Command Mode

Mgmt-CVX Configuration

Command Syntax

heartbeat-timeout period

no heartbeat-timeout

default heartbeat-timeout

Parameters

period heartbeat timeout interval (seconds). Value ranges from ***15*** to ***10800***. Default value is ***60***.

Guidelines

Best practices dictate that the CVX client's heartbeat timeout value is identical to that of its CVX server.

Related Command

heartbeat-interval (Management-CVX) specifies the CVX client heartbeat interval.

Example

These commands set the CVX client timeout period to ***90*** seconds.

```
switch(config)# management cvx
switch(config-mgmt-cvx)# heartbeat-timeout 90
switch(config-mgmt-cvx)#
```

14.1.9.6 lldp run

The **lldp run** command enables LLDP on the Arista switch.

Command Mode

Global Configuration

Command Syntax

lldp run

no lldp run

default lldp run

Examples

- This command enables LLDP globally on the Arista switch.

```
switch(config)# lldp run
switch(config)#
```

- This command disables LLDP globally on the Arista switch.

```
switch(config)# no lldp run
switch(config)#
```

14.1.9.7 management cvx

The **management cvx** command places the switch in ***mgmt-CVX*** configuration mode to configure CVX client parameters.

Mgmt-CVX configuration mode is not a group-change mode; ***running-config*** is changed immediately upon entering commands. Exiting ***mgmt-CVX*** configuration mode does not affect the ***running-config***. The **exit** command returns the switch to global configuration mode.

The **no management cvx** and **default management cvx** commands delete all mgmt-CVX configuration mode statements from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

management cvx

no management cvx

default management cvx

exit

Commands Available in Mgmt-CVX Configuration Mode

- **heartbeat-interval (Management-CVX)**
- **heartbeat-timeout (Management-CVX)**
- **server host (Management-CVX)**
- **source-interface (Management-CVX)**
- **shutdown (Management-CVX)**

Examples

- This command places the switch in mgmt-CVX configuration mode.

```
switch(config)# management cvx
switch(s1)(config-mgmt-cvx)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-cvx)# exit
switch(config)#
```

14.1.9.8 manager

The **manager** command configures the IP address of the OVSDB controller for the HSC service, allowing CVX to connect to the controller.

The **no manager** and **default manager** commands remove the HSC manager configuration from ***running-config***.

Command Mode

CVX-HSC Configuration

Command Syntax

manager ip_address [port]

Parameters

- ***ip_address*** IP address of the HSC manager.
- **port connection port**. Values range from ***1*** to ***65535***; default value is ***6632***.

Related Commands

service hsc places the switch in CVX-HSC configuration mode.

Example

These commands point the HSC service to a controller at IP address ***192.168.2.5*** using the default port ***6632***.

```
switch(config)# cvx
switch(config-cvx)# service hsc
switch(config-cvx-hsc)# manager 192.163.2.5
switch(config-cvx-hsc)#
```

14.1.9.9 name-resolution force (CVX-OpenStack)

The **name-resolution force** command initiates an OpenStack controller function that communicates with the OpenStack Keystone and Nova services to update names of VMs and tenants mapped by the local OpenStack instance.

The OpenStack controller accesses the Keystone and Nova services in response to various triggering events (such as the creation of a new tenant, network or VM), and also at a regular interval configured by the **name-**

resolution interval (CVX-OpenStack) command (default interval 6 hours). The **name-resolution force** command is used to force an immediate update without waiting for a triggering event.

Command Mode

CVX-OpenStack Configuration

Command Syntax

name-resolution force

Related Commands

- **service openstack** places the switch in CVX-OpenStack configuration mode.
- **name-resolution interval (CVX-OpenStack)** sets the interval for automatic Keystone updates.

Example

These commands update the OpenStack instance immediately with data from the Keystone service.

```
switch(config)# cvx
switch(config-cvx)# service openstack
switch(config-cvx-openstack)# name-resolution force
switch(config-cvx-openstack)#
```

14.1.9.10 name-resolution interval (CVX-OpenStack)

The **name-resolution interval** command specifies the period between consecutive requests that the OpenStack controller sends to the Keystone service for VM and tenant name updates. Keystone is OpenStack's authentication and authorization service.

The default period is ***21600*** seconds (six hours).

The **name-resolution force (CVX-OpenStack)** command performs an immediate update, as opposed to waiting for the periodic update.

Command Mode

CVX-OpenStack Configuration

Command Syntax

name-resolution interval period

Parameters

period Keystone identity service polling interval (seconds).

Related Command

service openstack places the switch in **CVX-OpenStack** configuration mode.

Example

These commands set the name resolution interval period at ***18000*** (five hours).

```
switch(config)# cvx
switch(config-cvx)# service openstack
switch(config-cvx-openstack)# name-resolution interval 18000
switch(config-cvx-openstack)#
```

14.1.9.11 ovsdb-shutdown

The **ovsdb-shutdown** command shuts down the OVSDB server.

The **no ovsdb-shutdown** and **default ovsdb-shutdown** commands enable the OVSDB server by removing the **ovsdb-shutdown** command from the ***running-config***.

Command Mode

CVX-HSC Configuration

Command Syntax

ovsdb-shutdown

no ovsdb-shutdown

default ovsdb-shutdown

Related Command

The **service hsc** command places the switch in the ***CVX-HSC*** configuration mode.

Example

These commands shut down the OVSDB server used by the HSC service.

```
switch(config)# cvx
switch(config-cvx)# service hsc
switch(config-cvx-hsc)# ovsdb-shutdown
switch(config-cvx-hsc)#
```

14.1.9.12 port (CVX)

The **port** command specifies the TCP port number the CVX server listens on. The default port number is ***9979***.

The **no port** and **default port** commands restore the default port number by removing the port statement from ***running-config***.

Command Mode

CVX Configuration

Command Syntax

port ***port_number***

no port

default port

Parameter

port_number TCP port number. Value ranges from ***1*** to ***65535***.

Related Command

cvx places the switch in the ***CVX*** configuration mode.

Examples

- These commands configure ***9500*** as the CVX server port.

```
switch# config
switch(config)# cvx
switch(config-cvx)# port 9500
switch(config-cvx)#
```

- These commands restore the default port (***9979***) as the CVX server port.

```
switch(config-cvx)# no port
```

```
switch(config-cvx)#
```

14.1.9.13 resync-period

The **resync-period** command configures the grace period for completion of synchronization between the VXLAN control service and clients after a CVX restart. Arista recommends leaving the grace period set to its default of ***300*** seconds.

The **no resync-period** command disables VXLAN control service graceful restart. The **default resync-period** command resets the grace period to its default of ***300*** seconds.

Command Mode

CVX-VXLAN Configuration

Command Syntax

resync-period ***seconds***

no resync-period

default resync-period

Parameter

seconds synchronization grace period in seconds. Values range from ***30*** to ***4800***; default is ***300***.

Example

These commands reset the VXLAN control service synchronization grace period to ***300*** seconds.

```
switch(config)# cvx
switch(config-cvx)# service vxlan
switch(config-cvx-vxlan)# default resync-period
switch(config-cvx-vxlan)#
```

14.1.9.14 server host (Management-CVX)

The **server host** command configures the IP address or host name of the CVX server to which the CVX client device connects. The configuration of this address is required for the switch to function as a CVX client. By default, no CVX host address is specified.

The **no server host** and **default server host** commands remove the CVX host address assignment by removing the server host statement from the ***running-config***.

Command Mode

Mgmt-CVX Configuration

Command Syntax

server host host

no server host

default server host

Parameter

host IPv4 address (in dotted decimal notation) or FQDN host name of the CVX server.

Example

This command specifies ***10.1.1.14*** as the address of the server to which the CVX client connects.

```
switch(config)# management cvx
switch(config-mgmt-cvx)# server host 10.1.1.14
```

```
switch(config-mgmt-cvx)#
```

14.1.9.15 service hsc

The **service hsc** command enters the ***CVX-HSC*** configuration mode where the HSC service is enabled and configured.

CVX-HSC configuration mode is not a group change mode; the ***running-config*** is changed immediately upon entering commands. Exiting the ***CVX-HSC*** configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

CVX Configuration

Command Syntax

service hsc

Commands Available in CVX-HSC Configuration Mode

- **manager**
- **ovsdb-shutdown**
- **shutdown (CVX-HSC)**

Related Command

cvx places the switch into the CVX configuration mode.

Example

These commands enter the ***CVX-HSC*** configuration mode.

```
switch(config)# cvx
switch(config-cvx)# service hsc
switch(config-cvx-hsc)#
```

14.1.9.16 service openstack

The service openstack command places the switch in CVX-OpenStack configuration mode.

In order to integrate Arista switches into an OpenStack managed cloud network, OpenStack needs to interact with CVX to configure and maintain VLANs on appropriate physical switch ports that connect to hosts where the VMs reside.

CVX-OpenStack configuration mode is not a group change mode; the ***running-config*** is changed immediately upon entering commands. Exiting the ***CVX-OpenStack*** configuration mode does not affect the ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

CVX Configuration

Command Syntax

service openstack

Commands Available in CVX-OpenStack Configuration Mode

- **name-resolution force (CVX-OpenStack)**
- **name-resolution interval (CVX-OpenStack)**
- **shutdown (CVX-OpenStack)**

Related Command

cvx places the switch into the CVX configuration mode.

Example

These commands places the switch into the ***CVX-OpenStack*** configuration mode.

```
switch(config)# cvx
switch(config-cvx)# service openstack
switch(config-cvx-openstack)#
```

14.1.9.17 service topology

The **service topology** command configures the topology statically on CVX without running LLDP on the servers connected to switches.

The **no service topology** command removes the static topology configuration from the ***running-config***.

Command Mode

CVX Configuration Mode

Command Syntax

service topology

no service topology

Example

- The following command configures topology statically on the switch.

```
switch# config
switch(config)# cvx
switch(config-cvx)# service topology
switch(config-cvx-topology)#
```

14.1.9.18 service vxlan

The **service vxlan** command enters the ***CVX-VXLAN*** configuration mode where the VXLAN control service is enabled and configured.

The CVX-VXLAN configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***CVX-VXLAN*** configuration mode does not affect the ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

CVX Configuration

Command Syntax

service vxlan

Commands Available in CVX-VXLAN Configuration Mode

- **resync-period**
- **shutdown (CVX-VXLAN)**
- **vtep (CVX-VXLAN)**

Related Command

The **cvx** command places the switch into the CVX configuration mode.

Example

These commands enters the ***CVX-VXLAN*** configuration mode.

```
switch(config)# cvx
switch(config-cvx)# service vxlan
switch(config-cvx-vxlan)#
```

14.1.9.19 show cvx

The **`show cvx`** command displays the enable status and current configuration of CVX.

Command Mode

EXEC

Command Syntax

`show cvx`

Example

This command displays the status and configuration of CVX.

```
switch(config)# cvx
 cvx
 no shutdown
 heartbeat-interval 30
 heartbeat-timeout 90
switch(config-cvx)# dis
switch> show cvx
CVX Server
 Status: Enabled
 UUID: 75ce27ce-cc04-11e4-a404-233646319a2c
 Heartbeat interval: 30.0
 Heartbeat timeout: 90.0 switch>
```

14.1.9.20 show network physical-topology

The **`show network physical-topology`** command displays the network topology discovered through CVX.

Command Mode

EXEC

Command Syntax

`show network physical-topology` ***hosts***|***neighbors***

Parameters

- ***hosts*** Displays all hosts visible in the topology.
- ***neighbors*** Displays all connections in the network topology. Table is sorted by host name, and can be optionally filtered by host.

Example

- This command displays all visible hosts.

```
switch# show network physical-topology hosts

Unique Id            Hostname
-------------------- ------------------------------
001c.7385.be69       cvx287.sjc.aristanetworks.com
0000.6401.0000       cvc1
```

```
0000.6402.0000       cvc2
0000.6403.0000       cvc3
0000.6404.0000       cvc4
bcf6.85bd.8050       dsj14-rack14-tor1
```

- This command displays all connections in the topology.

```
switch# show network physical-topology neighbors

cvx287.sjc.aristanetworks.com

Interface          Neighbor Intf      Neighbor Host
------------------ ------------------ ----------------------
Ethernet1          Ethernet7          cvc4
Ethernet2          Ethernet7          cvc2
Ethernet9          Ethernet7          cvc1
Ethernet10         Ethernet7          cvc3
Management1        27                 dsj14-rack14-tor1

OUTPUT OMITTED FROM EXAMPLE
dsj14-rack14-tor1

Interface          Neighbor Intf      Neighbor Host
------------------ ------------------ ----------------------
27                 Management1        cvx287.sjc.aristanetwork
```

14.1.9.21 shutdown (CVX)

The **shutdown** command, in cvx mode, disables or enables the switch as a CVX server. By default, CVX is disabled on the switch.

The **no shutdown** command enables the switch as a CVX server. The **shutdown** and **default shutdown** commands disable the switch as a CVX server by removing the **no shutdown** command from ***running-config***.

Note: Be sure to de-configure or shut down all CVX client services before disabling CVX; failure to do so may result in CVX client services continuing to run after CVX has been disabled.

Command Mode

CVX Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Related Command

The **cvx** command places the switch in CVX configuration mode.

Examples

- These commands enable the switch as a CVX server.

```
switch# config
switch(config)# cvx
switch(config-cvx)# no shutdown
switch(config-cvx)#
```

- This command disables CVX on the switch.

```
switch(config-cvx)# shutdown
switch(config-cvx)#
```

14.1.9.22 shutdown (CVX-HSC)

The **shutdown** command, in ***CVX-HSC*** configuration mode, disables or enables the CVX service on the switch. HSC is disabled by default.

When a CVX server enables HSC, its clients (hardware VTEPs) are able to share state to establish VXLAN tunnels without the need for a multicast control plane. Configuration is also required on the client switches.

The **no shutdown** command enables the HSC service; the **shutdown** and **default shutdown** commands disable the HSC service.

Command Mode

CVX-VXLAN Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Related Command

The **service hsc** command places the switch into the CVX-HSC configuration mode.

Examples

- These commands enable the HSC service.

```
switch(config)# cvx
switch(config-cvx)# service hsx
switch(config-cvx-hsc)# no shutdown
switch(config-cvx-hsc)#
```

- These commands disable the HSC service.

```
switch(config)# cvx
switch(config-cvx)# service hsx
switch(config-cvx-hsc)# shutdown
switch(config-cvx-hsc)#
```

14.1.9.23 shutdown (Management-CVX)

The **shutdown** command, in the ***mgmt-cvx*** mode, disables or enables CVX client services on the switch. CVX services are disabled by default.

The **no shutdown** command enables CVX client services. The **shutdown** and **default shutdown** commands disable CVX client services by removing the corresponding no shutdown command from the **running-config**.

Command Mode

Mgmt-CVX Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- These commands enable CVX client services.

```
switch(config)# management cvx
switch(config-mgmt-cvx)# no shutdown
switch(config-mgmt-cvx)#
```

- This command disables CVX client services.

```
switch(config-mgmt-cvx)# shutdown
switch(config-mgmt-cvx)#
```

14.1.9.24 shutdown (CVX-OpenStack)

The **shutdown** command, in the ***cvx-openstack*** configuration mode, disables or enables CVX-OpenStack on the switch. CVX-OpenStack is disabled by default.

When a CVX server enables OpenStack services, its clients are accessible to the OpenStack network controller (Neutron). Integrating Arista switches into an OpenStack-managed cloud network requires OpenStack to interact with CVX to configure and maintain VLANs on appropriate physical switch ports that connect to the hosts where the VMs reside.

The **no shutdown** command enables CVX-OpenStack. The **shutdown** and **default shutdown** commands disable CVX-OpenStack by removing the corresponding no shutdown command from the ***running-config***.

Command Mode

CVX-OpenStack Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Related Command

service openstack places the switch in ***CVX-OpenStack*** configuration mode.

Examples

- These commands enable CVX-OpenStack.

```
switch(config)# cvx
switch(config-cvx)# service openstack
switch(config-cvx-openstack)# no shutdown
switch(config-cvx-openstack)#
```

- These commands disable CVX-OpenStack.

```
switch(config-cvx-openstack)#
switch(config-cvx-openstack)# shutdown
switch(config-cvx-openstack)#
```

14.1.9.25 shutdown (CVX-VXLAN)

The **shutdown** command, in ***CVX-VXLAN*** configuration mode, disables or enables the CVX VXLAN control service on the switch. VXLAN control service is disabled by default.

When a CVX server enables VXLAN control service, its clients (hardware VTEPs) are able to share state to establish VXLAN tunnels without the need for a multicast control plane. Configuration is also required on the client switches.

The **no shutdown** command enables the VXLAN control service. The **shutdown** and **default shutdown** commands disable the VXLAN control service.

Command Mode

CVX-VXLAN Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Related Command

The **service vxlan** command places the switch in CVX-VXLAN configuration mode.

Examples

- These commands enable VXLAN control service.

```
switch(config)# cvx
switch(config-cvx)# service vxlan
switch(config-cvx-vxlan)# no shutdown
switch(config-cvx-vxlan)#
```

- These commands disable VXLAN control service.

```
switch(config)# cvx
switch(config-cvx)# service vxlan
switch(config-cvx-vxlan)# shutdown
switch(config-cvx-vxlan)#
```

14.1.9.26 source-interface (Management-CVX)

The **source-interface** command specifies the interface from where the IPv4 address is derived for use as the source for outbound CVX packets that the switch sends as a CVX client. There is no default source interface assignment.

The **no source-interface** and **default source-interface** commands remove the source interface assignment for the CVX client by deleting the source-interface statement from the ***running-config***.

Command Mode

Mgmt-CVX Configuration

Command Syntax

source-interface INT_NAME

no source-interface

default source-interface

Parameters

INT_NAME: Interface type and number. Options include:

- **ethernet *e_num***: Ethernet interface specified by ***e_num***.
- **loopback *l_num***: Loopback interface specified by ***l_num***.
- **management *m_num***: Management interface specified by ***m_num***.
- **port-channel *p_num***: Port-Channel Interface specified by ***p_num***.
- **vlan *v_num***: VLAN interface specified by ***v_num***.

Example

These commands configure the CVX client to use the IP address ***10.24.24.1*** as the source address for its outbound packets.

```
switch# config
switch(config)# interface loopback 5
switch(config-if-Lo5)# ip address 10.24.24.1/24
switch(config-if-Lo5)# exit
switch(config)# management cvx
switch(config-mgmt-cvx)# source-interface loopback 5
switch(config-mgmt-cvx)#
```

14.1.9.27 vtep (CVX-HSC)

The HSC service sends flood lists to each VTEP through CVX. Some controllers (such as VMware NSX's Service Nodes) implement replication nodes for head-end replication of unknown packets. For these controllers, BUM packets should be sent to a single replication node (send-to-any replication), and the flood list sent by the HSC service is a list of replication nodes. Other controllers (such as Nuage VSP) require each VTEP to perform its own head-end replication. For these, BUM packets should be sent to every known VTEP, and the flood list sent by the HSC service is the list of VTEPs.

The default behavior is to use a send-to-any replication list of VTEPs. If the required behavior is send-to-all replication of, use the all option of the **vtep** command in the ***CVX-HSC*** configuration mode.

Command Mode

CVX-HSC Configuration

Command Syntax

vtep flood ***list type*** all | any

no vtep flood ***list type***

default vtep flood ***list type***

Parameters

- **all**: send-to-all replication; flood list is the list of VTEPs.
- **any**: send-to-any replication; flood list is a list of replication nodes. This is the default setting.

Example

These commands configure the HSC to use send-to-all replication.

```
switch(config)# cvx
switch(config-cvx)# service hsc
switch(config-cvx-hsc)# vtep flood list type all
switch(config-cvx-hsc)#
```

14.1.9.28 vtep (CVX-VXLAN)

The OVSDB management protocol includes provisions for control-plane MAC learning, which allows MAC addresses to be distributed among VTEPs without using the data plane. Some controllers (such as VMware NSX) take advantage of this facility; others (such as Nuage VSP) do not. By default, CVX uses control-plane MAC learning.

To switch to data plane MAC learning, use the **vtep** command in the CVX-VXLAN configuration mode, as shown below.

Command Mode

CVX-VXLAN Configuration

Command Syntax

vtep mac-learning [control-plane|data-plane]

Related Command

The **service vxlan** command places the switch into the ***CVX-VXLAN*** configuration mode.

Example

These commands configure CVX to use data-plane MAC address learning.

```
switch(config)# cvx
switch(config-cvx)# service vxlan
switch(config-cvx-vxlan)# vtep mac-learning data-plane
switch(config-cvx)#
```

Chapter 15

Routing Protocols

The Routing Protocols chapter contains the following sections:

- Routing Information Protocol (RIP)
- Open Shortest Path First – Version 2
- Open Shortest Path First – Version 3
- IS-IS
- Border Gateway Protocol (BGP)
- Maintenance Mode
- Bidirectional Forwarding Detection

15.1 Routing Information Protocol (RIP)

This chapter contains the following sections.

- RIP Conceptual Overview
- Running RIP on the Switch
- Configuring RIP on Multiple VRFs
- RIP Commands

15.1.1 RIP Conceptual Overview

Routing Information Protocol (RIP) is a routing protocol typically used as an Interior Gateway Protocol (IGP). RIP uses hop counts only to determine the shortest path to a destination. To avoid loops, RIP limits its paths to a maximum of ***15*** hops, making it an ineffective protocol for large networks. RIP Version 2 supports Classless Inter-Domain Routing (CIDR) and uses IP multicast at address ***224.0.0.9*** to share the routing table with adjacent routers.

RIP sends updates whenever there is a change in the network topology and periodic updates when there are no changes. Receiving switches update their routing table whenever the update includes topology changes. Because RIP transmits the entire routing table every ***30*** seconds, RIP updates can generate heavy traffic loads in large or complicated networks.

Each switch also sends a list of distance-vectors to each of its neighbors periodically. The distance-vector is the metric RIP uses to express the cost of a route, and it describes the number of hops required to reach a destination. Each hop is typically assigned a hop count value of ***1***, and the router adds ***1*** to the metric when it receives a routing update and adds the network to its routing table.

To remove dead routes from its routing table, RIP marks a route for deletion if the router does not receive an advertisement for it within the expiration interval, then removes it from the routing table after the deletion interval.

15.1.2 Running RIP on the Switch

15.1.2.1 Accessing RIP Configuration Mode and Enabling RIP

15.1.2.1.1 RIP Configuration Mode

The router rip command places the switch in router-RIP configuration mode to configure Routing Information Protocol (RIP) routing.

Example

This command places the switch in router-RIP configuration mode.

```
switch(config)#router rip
switch(config-router-rip)#
```

Using the router rip command puts the switch in router-RIP configuration mode, but does not enable RIP on the switch.

15.1.2.1.2 Enabling RIP

Routing Information Protocol (RIP) is disabled on the switch by default. The **no** shutdown (RIP) command in router-RIP configuration mode will enable RIP.

Example

This command enables RIP on the switch.

```
switch(config-router-rip)#no shutdown
switch(config-router-rip)#
```

Issuing this command enables RIP, but to send and receive RIP route updates and to route packets via RIP you must also specify interfaces on which RIP will run by using the network (RIP) command.

15.1.2.1.3 Disabling RIP

You can disable RIP in two ways. The shutdown (RIP) command disables RIP on the switch but maintains all user-entered router-RIP configuration statements in the ***running-config***. The **no** router rip command disables RIP and removes all user-entered router-RIP configuration statements from the ***running-config***.

Examples

- This command disables RIP on the switch and removes all user-entered router-RIP configuration.

  ```
  switch(config)#no router rip
  switch(config)#
  ```

- This command disables RIP on the switch, but preserves all user-entered router-RIP configuration.

  ```
  switch(config-router-rip)#shutdown
  switch(config-router-rip)#
  ```

15.1.2.2 Configuring RIP

Issuing the **no** shutdown (RIP) command in router-RIP configuration mode enables RIP, but to run RIP on an interface you must specify a RIP network by using the network (RIP) command.

You can also configure the redistribution of routes learned from other protocols, set the default metric and administrative distance for redistributed routes, configure the timing of various RIP events, and configure specific interfaces to send RIP update packets by broadcast instead of multicast.

15.1.2.2.1 Specifying RIP Networks

The network (RIP) command identifies networks on which RIP will run and also specifies which routes RIP will accept into its routing table. You can issue the network (RIP) command multiple times to build up a list of RIP networks. No RIP networks are configured by default, so in order to route packets and send and receive RIP updates you must specify one or more RIP networks.

To disable RIP on a specific network, use the **`no network (RIP)`** command.

Examples

- This command enables RIP on ***10.168.1.1/24***.

```
switch(config-router-rip)network 10.168.1.1/24
switch(config-router-rip)#
```

- This command disables RIP on ***10.168.1.1/24***.

```
switch(config-router-rip)#no network 10.168.1.1/24
switch(config-router-rip)#
```

15.1.2.2.2 Redistributing Routes Learned from Other Protocols into RIP

To enable route import from a specified protocol into RIP, use the redistribute (RIP) command. Additionally, you can apply a route map to the incoming routes to filter which routes are added to the RIP routing table. All connected routes are redistributed into RIP by default.

Example

This command redistributes all routes learned from OSPF into RIP.

```
switch(config-router-rip)#redistribute OSPF
switch(config-router-rip)#
```

15.1.2.2.3 Configuring RIP Timers

When RIP is running on the switch, it sends unsolicited route updates and deletes expired routes at regular intervals. To configure the timing of those events, use the timers (RIP) command. The command takes three parameters: the update interval, the route expiration time, and the route deletion time.

The update interval is the amount of time in seconds that the switch waits between sending unsolicited RIP route updates to its neighbors. The route expiration time is how long the switch waits before marking an unadvertised route for deletion (the counter resets whenever an advertisement for the route is received). And the route deletion time is how long the switch waits between marking a route for deletion and removing it from the routing table. During the deletion interval, the switch continues to forward packets on the route.

Example

This command sets the update interval to ***60*** seconds, expiration time to ***90*** seconds, and deletion time to ***150*** seconds.

```
switch(config-router-rip)#timers 60 90 150
switch(config-router-rip)#
```

15.1.2.2.4 Configuring an Interface to Transmit Broadcast RIP Updates

By default, the switch uses RIP version 2 and multicasts RIP update packets from all participating interfaces. To reconfigure a specific interface to send updates as broadcast packets, use the rip v2 multicast disable command in the configuration mode for the interface.

Example

The following commands configure RIP version 2 broadcasting on ***interface ethernet5***.

```
switch(config)#interface ethernet5
switch(config-if-Et5)#rip v2 multicast disable
switch(config-if-Et5)#exit
switch(config)#
```

15.1.2.3 Displaying RIP Information

15.1.2.3.1 Displaying RIP Routes

To see a listing of the RIP routes in the switch's routing table, use the show ip rip database command. (You can also display similar information using the RIP option in the show ip route command.)

Examples

- This command displays all active rip routes.

```
switch>show ip rip database
10.168.11.0/24 directly connected, Et4
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et4
[2] via 10.168.15.2, 00:00:20, Et1
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et3
```

-

```
switch>show ip rip database 10.168.13.0/16
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et4
[2] via 10.168.15.2, 00:00:20, Et1
```

  This command submits a query for RIP route information for a network.

15.1.2.3.2 Displaying RIP Route Gateways

To see information about the switch's RIP route gateways, use the show ip rip neighbors command. The output displays the IPv4 address, the last heard time of the gateway, and characteristic flags applying to the gateway.

Example

This command displays information about all the gateways of RIP routes.

```
switch>show ip rip neighbors
Gateway     Last-Heard   Bad-Packets   Bad-Routes   Flags
10.2.12.33  00:00:15                                SRC, TRSTED,
                                                    ACCPTED, RJCTED,
                                                    Q_RJCTED, AUTHFAIL
```

15.1.3 Configuring RIP on Multiple VRFs

VRF support for Routing Information Protocol (RIP) allows instances of RIP on multiple non-default VRFs on the same router. By default, all interfaces belong to the default VRF until VRF forwarding is executed.

The **vrf instance** and **vrf (Interface mode)** commands configure a non-default VRF, enable routing in it, and configure the network command under the configuration router RIP for the prefix to which the interface belongs.

The router rip vrf command places the switch in router-RIP configuration mode to configure a RIP routing instance in a non-default VRF.

Examples

- These commands configure a non-default VRF and enable unicast routing in it.

```
switch(config)# vrf instance test
switch(config-vrf-test)# exit
switch(config)# ip routing vrf test
switch(config)#
```

- This command configures a RIP instance in a non-default VRF.

```
switch(config)# router rip vrf test
switch(config-router-rip-router-rip-vrf-test)# no shutdown
switch(config-router-rip)# exit
switch(config)#
```

- This command configures an interface as part of a non-default VRF by configuring the network command under the configuration router RIP for the prefix to which the interface belongs.

```
switch(config)# interface Ethernet 3 / 1
switch(config-if-Et3/1)# no switchport
switch(config-if-Et3/1)# ip address 1.0.0.1/24
switch(config-if-Et3/1)# vrf test
switch(config-if-Et3/1)# network 1.0.0.1
switch(config-if-Et3/1)# exit
switch(config)#
```

15.1.4 RIP Commands

Global Configuration Commands

- router rip
- router rip vrf

Interface Configuration Commands

- rip v2 multicast disable

Router-RIP Configuration Mode

- distance (RIP)
- distribute-list (RIP)
- metric default
- network (RIP)
- redistribute (RIP)
- shutdown (RIP)
- timers (RIP)

Display Commands – EXEC Mode

- show ip rip database
- show ip rip neighbors

15.1.4.1 distance (RIP)

The **distance** command assigns an administrative distance to routes that the switch learns through RIP. Routers use administrative distances to select a route when two protocols provide routing information to the same destination. Distance values range from ***1 to 255***; lower distance values correspond to higher reliability. The default RIP distance value is ***120***.

The **no distance** and **default distance** commands restore the administrative distance default value of ***120*** by removing the **distance** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

distance ***distance_value***

no distance

default distance

Parameter

distance_value distance assigned to RIP routes. Values range from ***1*** to ***255***.

Example

These commands assign an administrative distance of ***75*** to RIP routes.

```
switch(config)# router rip
switch(config-router-rip)# distance 75
switch(config-router-rip)#
```

15.1.4.2 distribute-list (RIP)

The **distribute-list** command allows users to filter out routes that are received or sent out. The distribute-list command influences which routes the router installs into its routing table and advertises to its neighbors.

Configuration Notes:

- Only one inbound distribute-list is allowed per interface.
- Only one outbound distribute-list is allowed per interface.
- Only one globally-defined inbound distribute-list is allowed.
- Only one globally-defined outbound distribute-list is allowed.
- Not all match clauses in a route-map are supported using RIP routes filtering. These match clauses for distribute-lists are supported:
  - **match ip address access-list**
  - **match ip address prefix-list**
- The **distribute-list** command does not enforce the specified route-map to contain only supported match clauses.
- Permit or deny can be specified in both prefix/access list and route-map configurations. The following rules apply when filtering routes:
  - Routes permitted by the prefix/access lists are treated as matched.
  - Matched routes are filtered based on the permit or deny option configured for the route-map clause.

- Unmatched routes are further evaluated by the next route-map clause.
- If a route does not match any clause in a route-map, it is denied.
- If the route-map given in the **distribute-list** command is not configured, then all routes are permitted.
- When multiple inbound (or outbound) distribute-lists are configured, only the most specific one is applied.

The **no distribute-list** and **default distribute-list** commands remove the corresponding **distribute-list** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

distribute-list DIRECTION MAP [**INTF**]

no distribute-list DIRECTION MAP [**INTF**]

default distribute-list DIRECTION MAP [**INTF**]

Parameters

- **DIRECTION** direction specifies if distribute-list is applied on inbound or outbound traffic. Valid options include:
  - **in** specifies inbound as the direction the distribute-list is applied.
  - **out** specifies outbound as the direction the distribute-list is applied.
- **MAP** specifies route map that assigns attribute values to the network. Options include:
  - ***no parameter*** attributes are not assigned through a route map.
  - **route-map *map_name*** attributes listed by specified route map are assigned to the network.
- **INTF** interface to be configured. Options include:
  - **ethernet *e_num*** Ethernet interface.
  - **loopback *l_num*** Loopback interface.
  - **port-channel *p_num*** Port channel interface.
  - **vlan *v_num*** VLAN interface.

Examples

- The following commands demonstrate that an **access-list** or **prefix-list** can be used within a ***route-map*** for use in a **distribute-list**.

```
switch(config)# ip prefix-list 8to24 seq 5 permit 0.0.0.0/0 ge 8 le 24
switch(config)# route-map myRouteMap permit 10
switch(config-route-map-myRouteMap)# match ip address prefix-list 8to24
switch(config-route-map-myRouteMap)# exit
switch(config)#
switch(config)# router rip
switch(config-router-rip)# distribute-list in route-map myRouteMap
switch(config-router-rip)#
```

- These commands suppress routes advertised on a particular interface.

```
switch(config)# ip prefix-list 2 seq 10 deny 30.1.1.0/24
switch(config)# route-map myRmOut permit 10
switch(config-route-map-myRmOut)# match ip address prefix-list 2
switch(config-route-map-myRouteMap)# exit
switch(config)# router rip
switch(config-router-rip)# distribute-list out route-map myRmOut
```

15.1.4.3 metric default

The **metric default** command specifies the metric value assigned to RIP routes learned from other protocols. All routes imported into RIP receive the default metric unless a matching route-map exists for the route. The route metric of ***0*** is assigned to redistributed connected and static routes. The default metric values range from ***0*** to ***16*** with a default value of ***1***.

The **no metric default** and **default metric default** commands remove the **metric default** command from ***running-config*** and returns the metric value to its default value of ***1***.

Command Mode

Router-RIP Configuration

Command Syntax

metric default ***metric_value***

no metric default

default metric default

Parameter

metric_value default metric value assigned. Values range from ***0 to 16***; default is ***1***.

Example

This command sets the default metric value to ***5***.

```
switch(config)# router rip
switch(config-router-rip)# metric default 5
switch(config-router-rip)#
```

15.1.4.4 network (RIP)

The **network** command specifies which network the switch runs Routing Information Protocol (RIP), and also specifies which routes will be accepted into the RIP routing table. Multiple network commands can be issued to create a network list on which RIP runs.

The switch enables RIP on all interfaces in the specified network.

The **no network** and **default network** commands disable RIP on the specified network by removing the corresponding **network** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

network NETWORK_ADDRESS

no network NETWORK_ADDRESS

default network NETWORK_ADDRESS

Parameters

NETWORK_ADDRESS network IP address. Entry formats include the following:

- ***ipv4_subnet*** IPv4 subnet (CIDR notation).
- ***ipv4_addr*** **mask** ***wildcard_mask*** IP address and wildcard-mask.

Examples

- This command enables RIP on ***10.168.1.1/24***.

```
switch(config)# router rip
switch(config-router-rip)# network 10.168.1.1/24
switch(config-router-rip)#
```

- This command also enables RIP on ***10.168.1.1/24***.

```
switch(config-router-rip)# network 10.168.1.1 mask 0.0.0.255
switch(config-router-rip)#
```

15.1.4.5 redistribute (RIP)

The **redistribute** command enables the importing of routes from a specified routing domain to RIP.

- **connected** by default, RIP redistributes all connected routes that are established when IP is enabled on an interface. The route-map parameter facilitates the exclusion of connected routes from redistribution by specifying a route map that denies the excluded routes.
- **BGP**, **OSPF**, and **IP static routes** by default, routes are not redistributed. The **redistribute** command without the route-map parameter facilitates the redistribution of all routes from the specified source.

The **no redistribute** and **default redistribute** commands reset the default route redistribution setting by removing the **redistribute** statement from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

redistribute connected **ROUTE_MAP**

redistribute **ROUTE_TYPE** [**ROUTE_MAP**]

no redistribute connected **ROUTE_MAP**

no redistribute **ROUTE_TYPE**

default redistribute connected **ROUTE_MAP**

default redistribute **ROUTE_TYPE**

Parameters

- **ROUTE_TYPE** source from which routes are redistributed. Options include:
  - **BGP** routes from a BGP domain.
  - **OSPF** routes from an OSPF domain.
  - **OSPF match external** routes external to RIP, but imported from OSPF.
  - **OSPF match internal** OSPF routes that are internal to the AS.
  - **static** IP static routes.
- **ROUTE_MAP** route map that determines the routes that are redistributed. Options include:
  - ***no parameter*** all routes are redistributed.
  - **route-map** ***map_name*** only routes in the specified route map are redistributed.

Example

These commands redistribute OSPF routes into RIP.

```
switch(config)# router rip
switch(config-router-rip)# redistribute OSPF
switch(config-router-rip)#
```

15.1.4.6 rip v2 multicast disable

The **rip v2 multicast disable** command specifies the transmission of Routing Information Protocol (RIP) Version 2 update packets from the configuration mode interface as broadcast to ***255.255.255.255***.

The **no rip v2 multicast disable** and **default rip v2 multicast disable** commands specify the transmission of update packets as multicast to ***224.0.0.9*** if the configuration mode interface is multicast capable. Updates are broadcast if the interface is not multicast capable.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

rip v2 multicast disable

no rip v2 multicast disable

default rip v2 multicast disable

Example

The following example configures version 2 broadcasting on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# rip v2 multicast disable
switch(config-if-Et5)# exit
switch(config)#
```

15.1.4.7 router rip vrf

The **router rip** command places the switch in router-RIP configuration mode to configure an RIP routing instance in the non-default VRF.

The **no router rip vrf** and **default router rip vrf** commands disable an RIP routing instance in the non-default VRF, and remove all user-entered **router-rip** configuration statements from ***running-config***. To disable RIP without removing configuration statements, use the shutdown (RIP) command.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

router rip vrf [**RIP_INSTANCE**]

no router rip vrf [**RIP_INSTANCE**]

default router rip vrf [**RIP_INSTANCE**]

Parameter

RIP_INSTANCE configure a RIP VRF instance in the non-default VRF.

Examples

- This command configures a RIP instance in the non-default VRF.

```
switch(config)# router rip vrf test
switch(config-router-rip-router-rip-vrf-test)# no shutdown
switch(config-router-rip)# exit
```

```
switch(config)#
```

- This command disables a RIP instance in the non-default VRF.

```
switch(config)# no router rip vrf test
switch(config)#
```

15.1.4.8 router rip

The **router rip** command places the switch in ***router-rip*** configuration mode to configure the Routing Information Protocol (RIP) routing process. Router-rip configuration mode is not a group change mode; ***running-config*** is changed immediately upon command entry. The **exit** command does not affect ***running-config***.

The **no router rip** and **default router rip** commands disable RIP and remove all user-entered **router-rip** configuration statements from ***running-config***. To disable RIP without removing configuration statements, use the **shutdown (RIP)** command.

The **exit** command returns the switch to the ***global*** configuration mode.

Command Mode

Global Configuration

Command Syntax

router rip

no router rip

default router rip

Commands Available in router-rip Configuration Mode

- distance (RIP)
- network (RIP)
- redistribute (RIP)
- shutdown (RIP)
- timers (RIP)

Example

This command places the switch in the ***router-rip*** configuration mode.

```
switch(config)# router rip
switch(config-router-rip)#
```

15.1.4.9 show ip rip database

The **show ip rip database** command displays information about routes in the Routing Information Base. The default command displays active routes and learned routes not used in deference to higher priority routes from other protocols.

This command has the following forms:

- **default (no arguments):** information about all RIP routes.
- **IPv4 address and mask:** information about the referenced addresses.
- **active:** information about routes not superseded by routes from other protocols.

Command Mode

EXEC

Command Syntax

show ip rip database [**FILTER**]

Parameters

FILTER routing table entries that the command displays. Values include :.

- ***no parameter*** displays all routing table entries.
- **active** displays all active routing table entries.
- ***net_addr*** subnet address (CIDR or address-mask). Command displays entries in this subnet.

Examples

- This command displays all active rip routes.

```
switch> show ip rip database active
10.168.11.0/24 directly connected, Et4
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et4
[2] via 10.168.15.2, 00:00:20, Et1
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et3
```

- This command submits a query for RIP route information for a network.

```
switch> show ip rip database 10.168.13.0/16
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et4
[2] via 10.168.15.2, 00:00:20, Et1
```

- This command returns information for all RIP routes.

```
switch> show ip rip database
10.1.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, holddown
10.2.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, holddown
10.3.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, inactive
10.212.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, active
10.214.0.0/255.255.255.0
[1] via 10.8.12.17, 00:00:30, Et4, active
```

15.1.4.10 show ip rip neighbors

The **show ip rip neighbors** command displays information about all RIP route gateways. The output displays the IPv4 address, the last heard time of the gateway, and characteristic flags applying to the gateway.

Command Mode

EXEC

Command Syntax

show ip rip neighbors

Example

The **show ip rip neighbors** command displays information about all gateways of RIP routes.

```
switch> show ip rip neighbors
Gateway     Last-Heard      Bad-Packets     Bad-Routes    Flags
10.2.12.33  00:00:15                                      SRC, TRSTED,
                                                          ACCPTED, RJCTED,
```

```
                                                    Q_RJCTED, AUTHFAIL
```

15.1.4.11 shutdown (RIP)

The **shutdown** command disables RIP on the switch without modifying the RIP configuration. RIP is disabled by default.

The **no shutdown** command enables RIP. The **default shutdown** command disables RIP.

Command Mode

Router-RIP Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- This command disables RIP on the switch.

```
switch(config)# router rip
switch(config-router-rip)# shutdown
switch(config-router-rip)#
```

- This command enables RIP on the switch.

```
switch(config-router-rip)# no shutdown
switch(config-router-rip)#
```

15.1.4.12 timers (RIP)

The **timers** command configures the update interval, the expiration time, and the deletion time for routes received and sent through RIP. The command requires value declaration of all values.

- The update time is the interval between unsolicited route responses.
- The expiration time is initialized when a route is established and any time an update is received for the route.
- The deletion time is initialized when the expiration time elapses and the route is invalid. It is retained in the routing table until deletion time expiry.

The **no timers** and **default timers** commands return the timer values to their default values by removing the **timers** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

timers ***update_time expire_time deletion_time***

no timers

default timers

Parameters

- ***update_time*** Default is ***30*** seconds.
- ***expire_time*** Default is ***180*** seconds.
- ***deletion_time*** Default is ***120*** seconds.

Parameter values are in seconds and range from ***5 to 2147483647***.

Example

This command sets the update (***60*** seconds), expiration (***90*** seconds), and deletion (***150*** seconds) times.

```
switch(config)# router rip
switch(config-router-rip)# timers 60 90 150
switch(config-router-rip)#
```

15.2 Open Shortest Path First – Version 2

Open Shortest Path First (OSPF) is a link-state routing protocol that operates within a single autonomous system. OSPF version 2 is defined by ***RFC 2328***.

This section contains the following topics:

- OSPFv2 Introduction
- OSPFv2 Conceptual Overview
- Configuring OSPFv2
- OSPFv2 Configuration Examples
- OSPFv2 Commands

15.2.1 OSPFv2 Introduction

This section contains the following topics:

- Supported Features
- Features Not Supported

15.2.1.1 Supported Features

Arista switches support the following OSPFv2 functions:

- A single OSPFv2 instance.
- Intra- and inter-area routing.
- Type 1 and 2 external routing.
- Broadcast and P2P interfaces.
- Stub areas.
- Not so stubby areas (NSSA) (***RFC 3101***).
- MD5 Authentication.
- Redistribution of static, IP, and BGP routes into OSPFv2 with route map filtering.
- Opaque LSAs (***RFC 2370***).
- Graceful restart (***RFC 3623***).
- OSPF Routes over GRE Tunnels

15.2.1.2 Features Not Supported

The following OSPFv2 functions are not supported in the current version:

- NBMA, demand circuit, and P2MP interfaces
- OSPFv2 MIB support

15.2.2 OSPFv2 Conceptual Overview

This section contains the following topics:

- Storing Link States
- Topology
- Link Updates
- OSPFv2 Route Redistribution Instance
- OSPFv2 and BFD Sessions for Adjacencies in any State
- OSPFv2 Multiple Instances Support
- OSPF Routes over GRE Tunnels

15.2.2.1 Storing Link States

OSPFv2 is a dynamic, link-state routing protocol, where links represent interfaces or routable paths. Dynamic routing protocols calculate the most efficient path between locations based on bandwidth and device status.

A Link State Advertisement (LSA) is an OSPFv2 packet that communicates a router's topology to other routers. The Link State DataBase (LSDB) stores an areas topology database and is composed of LSAs received from other routers. Routers update the LSDB by storing LSAs from other routers.

15.2.2.2 Topology

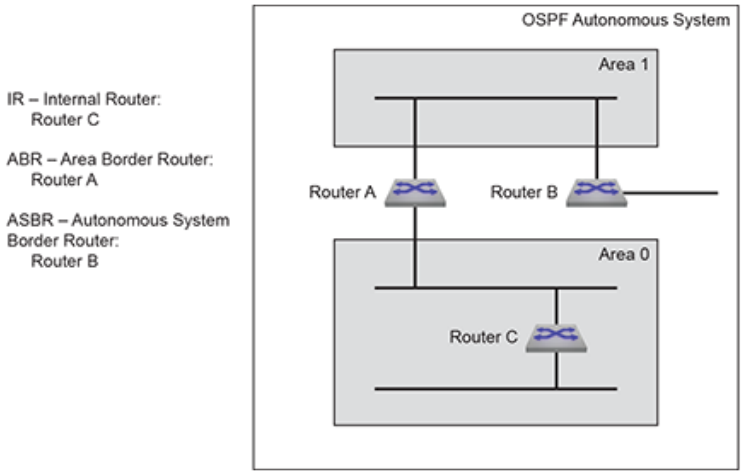
An Autonomous System (AS) is the IP domain within which a dynamic protocol controls the routing of traffic. In OSPFv2, an AS is composed of areas, which define the LSDB computation boundaries. All routers in an area store identical LSDBs. Routers in different areas exchange updates without storing the entire database, reducing information maintenance on large, dynamic networks.

An AS shares internal routing information from its areas and external routing information from other processes to inform routers outside the AS about routes the network can access. Routers that advertise routes on other ASs commit to carry data to the IP space on the route.

OSPFv2 defines these routers:

- Internal Router (IR) a router whose interfaces are contained in a single area. All IRs in an area maintain identical LSDBs.
- Area Border Router (ABR) a router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area.
- Autonomous System Boundary Router (ASBR) a gateway router connecting the OSPFv2 domain to external routes, including static routes and routes from other autonomous systems.
- **OSPFv2 Router Types** displays the OSPFv2 router types.

Figure 15-1: OSPFv2 Router Types



OSPFv2 areas are assigned a number between ***0*** and ***4,294,967,295*** (2321). Area numbers are often expressed in dotted decimal notation, similar to IP addresses.

Each AS has a backbone area, designated as area 0, that connects to all other areas. The backbone receives routing information from all areas, then distributes it to the other areas as required.

OSPFv2 area types include:

- Normal area accepts intra-area, inter-area, and external routes. The backbone is a normal area.
- Stub area does not receive router advertisements external to the AS. Stub area routing is based on a default route.
- Not-So-Stubby-Area (NSSA) may import external routes from an ASBR, does not receive external routes from the backbone, and does not propagate external routes to other areas.

15.2.2.3 Link Updates

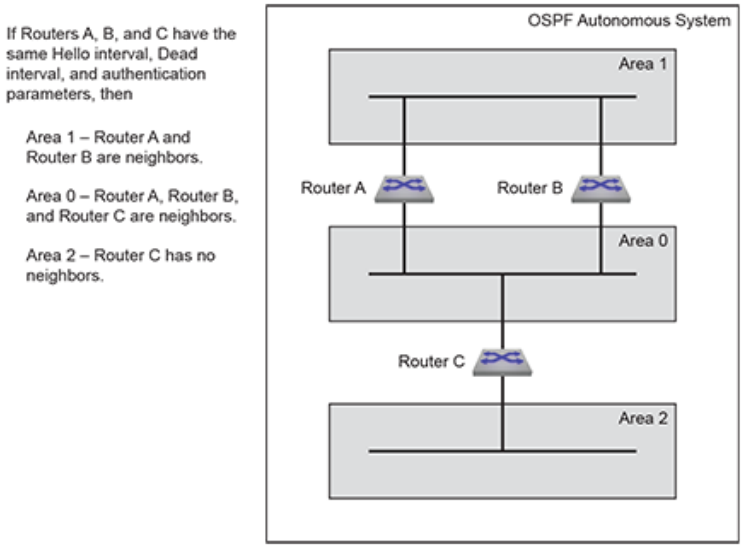
Routers periodically send hello packets to advertise status and establish neighbors. A routers hello packet includes IP addresses of other routers from which it received a hello packet within the time specified by the router dead interval. Routers become neighbors when they detect each other in their hello packets if they:

- share a common network segment.
- are in the same area.
- have the same hello interval, dead interval, and authentication parameters.

Neighbors form adjacencies to exchange LSDB information. A neighbor group uses hello packets to elect a Designated Router (DR) and Backup Designated Router (BDR). The DR and BDR become adjacent to all other neighbors, including each other. Only adjacent neighbors share database information.

OSPFv2 Neighbors illustrates OSPFv2 neighbors.

Figure 15-2: OSPFv2 Neighbors



The DR is the central contact for database exchanges. Switches send database information to their DR, which relays the information to the other neighbors. All routers in an area maintain identical LSDBs. Switches also send database information to their BDR, which stores this data without distributing it. If the DR fails, the BDR distributes LSDB information to its neighbors.

OSPFv2 routers distribute LSAs by sending them on all of their active interfaces. The router generates an LSA for a network defined and active on a passive interface but will not transmit this LSA on the passive interface as no adjacencies are formed.

When a routers LSDB is changed by an LSA, it sends the changes to the DR and BDR for distribution to the other neighbors. Routing information is updated only when the topology changes.

Routers use Dijkstras algorithm to calculate the shortest path to all known destinations, based on cumulative route cost. The cost of an interface indicates the transmission overhead and is usually inversely proportional to its bandwidth.

15.2.2.4 OSPFv2 Route Redistribution Instance

OSPFv2 Route Redistribution is used for redistributing OSPFv2 leaked and non-leaked routes from one instance to another when multiple OSPFv2 instances are configured. The OSPFv2 Route Redistribution is supported on all platforms in the multi-agent routing mode.

15.2.2.5 OSPFv2 and BFD Sessions for Adjacencies in any State

- BFD sessions are only established for OSPFv2 adjacencies that are in the FULL state. In a LAN environment this results in BFD sessions not being established for OSPFv2 adjacencies with DR Other neighbors.
- This feature provides configuration that enables the establishment of BFD sessions for OSPFv2 adjacencies that are in any state. This results in the BFD sessions being established for OSPFv2 adjacencies with DR Other neighbors.

15.2.2.6 OSPFv2 Multiple Instances Support

EOS Release 4.22.1F adds support for multiple OSPFv2 instances to be configured in the default VRF. OSPFv2 Multiple Instances Support provides isolation and allows segregating and dividing the link state database based on the interface.

Basic OSPFv2 functionality along with redistribution of OSPFv2 routes (all instances) into BGP and default information originate always is available forward from the ***EOS Release 4.22.1F***.

Support for graceful restart and BFD with multiple OSPFv2 instances was added in the ***EOS Release 4.23.1***.

15.2.2.6.1 OSPFv2 Multiple Instances Support Platform Compatibility

OSPFv2 Multiple Instances Support is supported on all platforms.

15.2.2.7 OSPF Routes over GRE Tunnels

This feature introduces the support for OSPF routes over GRE tunnels under default as well as non-default VRFs. The feature is disabled by default.

15.2.2.7.1 Limitations

The platform does not support any arbitrarily created TCAM profile. When the TCAM profile cannot be programmed, the show command prints **ERROR** in the status column.

15.2.3 Configuring OSPFv2

These sections describe basic OSPFv2 configuration steps:

- Configuring the OSPFv2 Instance
- Configuring OSPFv2 Areas
- Support for OSPFv2 dn-bit-ignore
- OSPFv2 Area Filter by Prefix-List
- IPv4 Unnumbered Interfaces
- Configuring Interfaces for OSPFv2
- Enabling OSPFv2
- OSPFv2 Multiple Instances Support Configuration
- OSPF Routes over GRE Tunnels Configuration

- Displaying OSPFv2 Status

15.2.3.1 Configuring the OSPFv2 Instance

15.2.3.1.1 Entering OSPFv2 Configuration Mode

The **router ospf** command places the switch in router-ospf configuration mode and creates an OSPFv2 instance if one was not previously created. The switch only supports one OSPFv2 instance and all OSPFv2 configuration commands apply to this instance.

When an OSPFv2 instance is already configured, the command must specify its process ID. Any attempt to define additional instances will fail and generate errors.

The process ID is local to the router and is used to identify the running OSPFv2 process. Neighbor OSPFv2 routers can have different process ID's.

Example

This command places the switch ***in router-ospf*** configuration mode and, if not previously created, creates an OSPFv2 instance with a process ID of ***100***.

```
switch(config)# router ospf 100
switch(config-router-ospf)#
```

15.2.3.1.2 Defining the Router ID

The router ID is a 32-bit number assigned to a router running OSPFv2. This number uniquely labels the router within an Autonomous System. Status commands identify the switch through the router ID.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IP address, if a loopback interface is active on the switch.
3. The highest IP address on the router.



Note: When configuring VXLAN on an MLAG, always manually configure the OSPFv2 router ID to prevent the switch from using the common VTEP IP address as the router ID.

The router-id (OSPFv2) command configures the router ID for an OSPFv2 instance.

Example

This command assigns ***10.1.1.1*** as the OSPFv2 router ID.

```
switch(config-router-ospf)# router-id 10.1.1.1
switch(config-router-ospf)#
```

15.2.3.1.3 Global OSPFv2 Parameters

These router-ospf configuration mode commands define OSPFv2 behavior.

LSA Overload

The **max-lsa (OSPFv2)** command specifies the maximum number of LSAs allowed in an LSDB database and configures the switch behavior when the limit is approached or exceeded. An LSA overload condition triggers these actions:

- **Warning**: the switch logs OSPF MAXLSAWARNING if the LSDB contains a specified percentage of the LSA maximum.

- **Temporary shutdown**: when the LSDB exceeds the LSA maximum, OSPFv2 is disabled and does not accept or acknowledge new LSAs. The switch re-starts OSPFv2 after a specified period (the default is ***5*** minutes).
- **Permanent shutdown**: the switch permanently disables OSPFv2 after performing a specified number of temporary shutdowns (the default is ***5***). This state usually indicates the need to resolve a network condition that consistently generates excessive LSA packets.

OSPFv2 is re-enabled with a **`router ospf`** command.

The LSDB size restriction is removed by setting the LSA limit to zero.



Note: if OSPFv2 has entered permanent shutdown, it can also be restarted by increasing the LSA limit to a value larger than the number of LSAs in the database. Setting the max-LSA value to zero will also restart OSPFv2, and will disable overload protection.

Example

- This command configures the OSPFv2 maximum LSA count to ***20000*** and triggers these actions:
  - The switch logs an **`OSPF MAXLSAWARNING`** if the LSDB has ***8000*** LSAs (***40%*** of ***20000***).
  - The switch temporarily disables OSPFv2 for ***10*** minutes if the LSDB contains ***20000*** LSAs.
  - The switch permanently disables OSPFv2 after four temporary OSPFv2 shutdowns.
  - The shutdown counter resets if the LSDB contains less than 20,000 LSAs for ***20*** minutes.

```
switch(config-router-ospf)# max-lsa 20000 40 ignore-time 10 ignore-count 4
 reset-time 20
switch(config-router-ospf)#
```

Logging Adjacency Changes

The **`log-adjacency-changes (OSPFv2`** command configures the switch to log OSPFv2 link-state changes and transitions of OSPFv2 neighbors into the up or down state.

Examples

- This command configures the switch to log transitions of OSPFv2 neighbors into the up or down state.

```
switch(config-router-ospf)# log-adjacency-changes
switch(config-router-ospf)#
```

- This command configures the switch to log all OSPFv2 link-state changes.

```
switch(config-router-ospf)# log-adjacency-changes detail
switch(config-router-ospf)#
```

OSPF RFC Compatibility

RFC 2328 and ***RFC 1583*** specify different methods for calculating summary route metrics. The compatible (OSPFv2) command allows the selective disabling of compatibility with ***RFC 2328***.

Example

This command sets the OSPF compatibility list with ***RFC 1583***.

```
switch(config)# router ospf 6
switch(config-router-ospf)# compatible rfc1583
switch(config-router-ospf)#
```

Administrative Distance

The **distance ospf (OSPFv2)** command configures the administrative distance for intra-area, inter-area, or external OSPF routes. To configure the administrative distance for multiple route types, the command must be entered multiple times. Administrative distances compare dynamic routes configured by different protocols. The default administrative distance for all routes is ***110***.

Note: OSPF links will flap if the administrative distance value is adjusted while OSPF is running, whether it is adjusted by entering the **distance ospf** command directly through the CLI or by applying a configuration file that contains the command.

Example

This command configures an administrative distance of **95** for OSPFv2 intra-area routes, and will cause links to flap if issued while OSPF is running.

```
switch(config-router-ospf)# distance ospf intra-area 95
switch(config-router-ospf)#
```

Passive Interfaces

The **passive-interface (OSPFv2)** command prevents the transmission of hello packets on the specified interface. Passive interfaces drop all adjacencies and do not form new adjacencies. Passive interfaces send LSAs but do not receive them. The router does not send or process OSPFv2 packets received on passive interfaces. The router advertises the passive interface in the router LSA.

The **no passive-interface** command re-enables OSPFv2 processing on the specified interface.

Examples

- This command configures ***vlan 2*** as a passive interface.

```
switch(config-router-ospf)# passive-interface vlan 2
switch(config-router-ospf)#
```

- This command configures ***vlan 2*** as an active interface.

```
switch(config-router-ospf)# no passive-interface vlan 2
switch(config-router-ospf)#
```

Redistributing Connected Routes

Redistributing connected routes causes the OSPFv2 instance to advertise all connected routes on the switch as external OSPFv2 routes. Connected routes are routes that are established when IPv4 is enabled on an interface.

Example

The **redistribute (OSPFv2) connected** command converts connected routes to OSPFv2 external routes.

```
switch(config-router-ospf)# redistribute connected
switch(config-router-ospf)#
```

Redistributing Static Routes

Redistributing static routes causes the OSPFv2 instance to advertise all static routes on the switch as external OSPFv2 routes. The switch does not support redistributing individual static routes.

Examples

- The **redistribute (OSPFv2)** static command converts the static routes to OSPFv2 external routes.

```
switch(config-router-ospf)# redistribute static
switch(config-router-ospf)#
```

- The **no redistribute (OSPFv2)** command stops the advertising of the static routes as OSPFv2 external routes.

```
switch(config-router-ospf)# no redistribute static
switch(config-router-ospf)#
```

Filtering Routes with Distribute Lists

An OSPF distribute list uses a route map or prefix list to filter specific routes from incoming OSPF LSAs; this filtering occurs after SPF calculation. The filtered routes are not installed on the switch, but are still included in LSAs sent by the switch. An OSPF router instance can have one distribute list configured.

If a prefix list is used, destination prefixes that do not match the prefix list will not be installed. If a route map is used, routes may be filtered based on address, next hop, or metric. OSPF external routes may also be filtered by metric type or tag.

The **distribute-list in** command specifies the filter to be used and applies it to the OSPF instance.

Example

These commands configure a prefix-list named ***dist_list1*** in OSPF instance ***5*** to filter certain routes from incoming OSPF LSAs.

```
switch(config)# router ospf 5
switch(config-router-ospf)# distribute-list prefix-list dist_list1 in
switch(config-router-ospf)#
```

15.2.3.1.4 Configuring OSPFv2 Route Redistribution

Use the **redistribute ospf instance** command to redistribute either the non-leaked routes, or both leaked and non-leaked routes. This command is configured under the ***router-ospf*** mode.

Examples

- The **leaked** clause includes both internal annd external routes.

```
switch(config-router-ospf)# redistribute ospf instance include leaked
<cr>
Options:
  include    Include leaked routes
  match      Routes learned by the OSPF protocol
  route-map  Specify which route map to use
```

- The **match** clause allows matching on the different OSPFv2 route types.

```
switch(config-router-ospf)# redistribute ospf instance match external
<cr>
Options:
  external       OSPF routes learned from external sources
  internal       OSPF routes learned from internal sources
  nssa-external  OSPF routes learned from external NSSA sources
```

- The following command redistributes the OSPFv2 external routes from all other OSPFv2 instances in the same VRF into the given instance.

```
switch(config-router-ospf)# redistribute ospf instance match external
```

- The following command redistributes the OSPFv2 internal leaked and non-leaked routes from all other instances in all VRFs into the given instance.

```
switch(config-router-ospf)# redistribute ospf instance include leaked match
 internal
```

- Matching based on the OSPFv2 instance ID is supported in the route-map.

```
switch(config)# route-map rm1 permit 10
switch(config-route-map-rm1)# match ospf instance 3
```

- The following command redistributes the OSPFv2 external routes from the OSPFv2 instance with ***ID 3*** in the same VRF into the given instance.

```
switch(config-router-ospf)# redistribute ospf instance match external route-
map rm1
```

Show Commands

- The **`show ip ospf database external`** command is used to verify if the AS-External LSAs are created in the instance for the redistributed route and advertised into the OSPFv2 domain.
- The **`show route-map`** command is used to display the details of a configured route-map.

15.2.3.2 Configuring OSPFv2 Areas

OSPFv2 areas are configured through area commands. The switch must be in ***router-ospf*** configuration mode, as described in **Entering OSPFv2 Configuration Mode**, to run area commands.

Areas are assigned a 32-bit number that is expressed in decimal or dotted-decimal notation. When an OSPFv2 instance configuration contains multiple areas, the switch only configures areas associated with its interfaces.

15.2.3.2.1 Configuring the Area Type

The **`area (OSPFv2)`** commands specifies the area type, refer **OSPFv2 Commands** section for the **`area`** commands. The switch supports three area types:

- **Normal area:** Area that accepts intra-area, inter-area, and external routes. The backbone area (area ***0***) is a normal area.
- **Stub area:** Area that does not advertise external routes. External routes are reached through a default summary route (***0.0.0.0***). Networks with no external routes do not require stub areas.
- **Not So Stubby Area (NSSA):** ASBRs advertise external LSAs directly connected to the area. External routes from other areas are not advertised and are reached through a default summary route.

The default area type is normal.

Examples

- This command configures area ***45*** as a stub area.

```
switch(config-router-ospf)# area 45 stub
switch(config-router-ospf)#
```

- This command configures area ***10.92.148.17*** as an NSSA.

```
switch(config-router-ospf)# area 10.92.148.17 NSSA
switch(config-router-ospf)#
```

15.2.3.2.2 Blocking All Summary Routes from Flooding the NSSA

The area nssa no-summary (OSPFv2) command configures the router to not import type-3 summary LSAs into the Not-So-Stubby Area (NSSA) and injects a default summary route (***0.0.0.0/0***) into the NSSA to reach the inter-area prefixes.

Example

This command directs the device not to import type-3 summary LSAs into the NSSA area and injects a default summary route (***0.0.0.0/0***) into the NSSA area.

```
switch(config)# router ospf 6
switch(config-router-ospf)# area 1.1.1.1 nssa no-summary
switch(config-router-ospf)#
```

15.2.3.2.3 Assigning Network Segments to the Area

Assigning Routes to an Area

The **network area (OSPFv2)** command assigns the specified network segment to an OSPFv2 area. The network can be entered in CIDR notation or by an address and wildcard mask.

The switch zeroes the host portion of the specified network address e.g. ***1.2.3.4/24*** converts to ***1.2.3.0/24*** and ***1.2.3.4/16*** converts to ***1.2.0.0/16***.

Example

Each of these equivalent commands assign the network segment ***10.1.10.0/24*** to area ***0***.

```
switch(config-router-ospf)# network 10.1.10.0 0.0.0.255 area 0
switch(config-router-ospf)# network 10.1.10.0/24 area 0
```

In each case, ***running-config*** stores the command in CIDR (prefix) notation.

Summarizing Routes

By default, ABRs create a summary LSA for each route in an area and advertise them to adjacent routers. The **area range (OSPFv2)** command aggregates routing information, allowing the ABR to advertise multiple routes with one LSA. The **area range (OSPFv2)** command can be used to suppress route advertisements.

Examples

- Two **network area** command assigns subnets to an area. The **area range (OSPFv2)** command summarizes the addresses, which the ABR advertises in a single LSA.

  ```
  switch(config-router-ospf)# network 10.1.25.80 0.0.0.240 area 5
  switch(config-router-ospf)# network 10.1.25.112 0.0.0.240 area 5
  switch(config-router-ospf)# area 5 range 10.1.25.64 0.0.0.192
  switch(config-router-ospf)#
  ```

- The **network area** command assigns a subnet to an area, followed by an **area range (OSPFv2)** command that suppresses the advertisement of that subnet.

  ```
  switch(config-router-ospf)# network 10.12.31.0 0.0.0.255 area 5
  switch(config-router-ospf)# area 5 range 10.12.31.0 0.0.0.255 not-advertise
  switch(config-router-ospf)#
  ```

15.2.3.2.4 Configuring Area Parameters

These router-ospf configuration mode commands define OSPFv2 behavior in a specified area.

Default Summary Route Cost

The **area default-cost (OSPFv2)** command specifies the cost of the default summary route that ABRs send into a stub area or NSSA. Summary routes, also called inter-area routes, originate in areas different than their destination.

Example

This command configures a cost of ***15*** for the default summary route in area ***23***.

```
switch(config-router-ospf)# area 23 default-cost 15
switch(config-router-ospf)#
```

Filtering Type 3 LSAs

The area filter (OSPFv2) command prevents an area from receiving Type 3 (Summary) LSAs from a specified subnet. Type 3 LSAs are sent by ABRs and contain information about one of its connected areas.

Example

This command prevents the switch from entering Type 3 LSAs originating from the ***10.1.1.2/24*** subnet into its ***area 2*** LSDB.

```
switch(config-router-ospf)# area 2 filter 10.1.1.2/24
switch(config-router-ospf)#
```

15.2.3.3 Support for OSPFv2 dn-bit-ignore

The OSPFv2 **dn-bit-ignore** command allows enabling or disabling the inclusion of LSAs having "Down" (DN) bit set in SPF calculations. The DN Bit is a loop prevention mechanism that implements when using OSPF as a CE - PE IGP protocol.

OSPFv2 only honors the DN-bit in **type-3** LSAs in non-default VRFs. Starting with ***Release EOS-4.25.0F***, OSPFv2 honors the DN-bit in **type-5** and **type-7** LSAs in non-default VRFs. This means that the type-3/5/7 LSAs with DN-bit set are not in SPF calculation, and any routes that carry LSAs are not installed in the routing table. This behavior changes when using the **dn-bit-ignore lsa type-5 type-7** command.

15.2.3.3.1 Configuration

OSPFv2

Use the command **dn-bit-ignore** to ignore the DN-bit in type-3/5/7 LSAs.

Use the command **dn-bit-ignore lsa type-5 type-7** to include only type-5, and type-7 LSAs having their DN-bit set in the SPF calculation. The commands **no dn-bit-ignore lsa type-5 type-7** or **default dn-bit-ignore lsa type-5 type-7** are configured to revert the behavior back to default. This command is available in **router ospf PROCESS_ID vrf VRF_NAME** configuration mode.



Note: This command is not available in the default VRF.

This command is for backwards compatibility to revert the behavior seen prior to Release ***EOS-4.25.0*** where the type-5 and type-7 LSAs with DN-bit set would get included in the SPF calculations.

```
(config)# router ospf 1 vrf red
(config-router-ospf-vrf-red)#?
```

```
...
 dn-bit-ignore             Disable DN-bit check for Type-3, Type-5 and Type-7
 LSAs in non-default VRFs
...
(config-router-ospf-vrf-red)#dn-bit-ignore ?
  lsa   Disable DN-bit check only for Type-5 and Type-7 LSAs in non-default
 VRFs
  <cr>
(config-router-ospf-vrf-red)#dn-bit-ignore lsa type-5 type-7
```

OSPFv3

Use the command **dn-bit-ignore** to include type-3/5/7 LSAs having their DN-bit set in the SPF calculation.

Use the commands **dn-bit-ignore** or **default dn-bit-ignore** to revert the behavior back to default. This command is available in **ipv6 router ospf PROCESS_ID vrf VRF_NAME** configuration mode and **router ospfv3 vrf <VRF_NAME>** configuration mode. Note that this command is not available in the default VRF, and that both configuration styles are captured below.

router ospfv3 Configuration Style

The **dn-bit-ignore** command is available under the **router ospfv3 vrf VRF_NAME** configuration mode. This disables the dn-bit check for Type-3/5/7 LSAs in non-default VRFs.

```
(config)# router ospfv3 vrf red
(config-router-ospfv3-vrf-red)# dn-bit-ignore
```

ipv6 router ospf Configuration Style

The **dn-bit-ignore** command is also available under the **ipv6 router ospf PROCESS_ID vrf VRF_NAME** configuration mode. This disables the dn-bit check for Type-3/5/7 LSAs in non-default VRFs.

```
(config)# ipv6 router ospf 1 vrf red
(config-router-ospfv3-vrf-red)# dn-bit-ignore
```

15.2.3.3.2 Show Commands

Use the **show running-config** command to verify whether the **dn-bit-ignore** command is configured.

15.2.3.4 OSPFv2 Area Filter by Prefix-List

The **ospf area <area_id> filter** command configures the set of prefixes to be filtered for multi-agent routing and the ribd routing protocols. Area filters are used to prevent specific prefixes from being announced by an area as Type 3 summary LSAs in an OSPFv2 Area Border Router (ABR).

Examples

The following configures a prefix-list filter to permit two prefixes and deny all others.

```
switch(config)# ip prefix-list type3Permit
switch(config-ip-pfx)# ip seq 10 permit 10.0.1.0/24
switch(config-ip-pfx)# ip seq 20 permit 10.0.2.0/24
switch(config-ip-pfx)# ip seq 30 deny 10.0.0.0/0
switch(config-ip-pfx)# exit
```

The following applies the filter to the backbone area.

```
switch(config)# router ospf 1
```

```
switch(config-router-ospf)# area 0 filter prefix-list type3Permit
```

The following configures a prefix-list to deny a list of prefixes and permit all others.

```
switch(config)# ip prefix-list type3Deny
switch(config-ip-pfx)# ip seq 10 deny 10.0.1.0/24
switch(config-ip-pfx)# ip seq 20 deny 10.0.2.0/24
switch(config-ip-pfx)# exit
```

The following applies the filter.

```
switch(config)# router ospf 1
switch(config-router-ospf)# area 1.1.1.1 filter prefix-list type3Deny
```

Show commands

The following displays the output of **`show ip ospf`** with the area filter listed.

```
switch# show ip ospf
Area 3.3.3.3
 Number of interface in this area is 2
   It is a normal area
   Traffic engineering is disabled
   Area has None authentication
   SPF algorithm executed 1 times
   Number of LSA 1. Checksum Sum 53568
   Number of opaque link LSA 0. Checksum Sum 0
   Number of opaque area LSA 0. Checksum Sum 0
   Area ranges are
      3.3.0.0/16 Cost 0 Advertise
      3.30.0.0/16 Cost 0 Advertise
   Area filter prefix-list type3Permit
```

15.2.3.5 IPv4 Unnumbered Interfaces

The **`ip address unnumbered`** command specifies a lending interface from which many interfaces may borrow the same address, reducing the number of unique IPv4 addresses needed. A lending interface is a loopback interface. Only one borrowing interface is referenced to one lender at a time even though multiple loopbacks may be used as lending interfaces. Unnumbered interfaces may reference the same or different lending interfaces. Any IPv4 routed interface is configurable as unnumbered interface and is referenced to one lending interface.

The following configures an unnumbered borrowing interface.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# ip address unnumbered Loopback1
```

OSPF configuration

To enable OSPF on an unnumbered interface, configure the area and set the network type to point-to-point under the ***interface*** config mode.

```
switch(config-if-Et1)# ip ospf area 1
switch(config-if-Et1)# ip ospf network point-to-point
```



Note: The **`network`** command under ***router ospf configuration mode*** is not supported for the configuration of unnumbered interfaces. You must specify the area and **`network point-to-point`** command in the configuration context of the unnumbered interface.

Enabling OSPF on the lending interface in the same area as the borrowing interfaces is recommended. For different unnumbered interfaces in different areas, configure them to use different loopbacks.

```
switch(config)# interface loopback 1
switch(config-if-Lo1)# ip address 1.1.1.1/32
switch(config-if-Lo1)# ip ospf area 1
```

ISIS configuration

To enable ISIS on an unnumbered interface, configure the area and set the network type to point-to-point under the ***interface*** config mode.

```
switch(config-if-Et1)# isis enable inst1
switch(config-if-Et1)# isis network point-to-point
```

Enabling ISIS on the lending interface in the same area as the borrowing interfaces is recommended.

```
switch(config)# interface loopback 1
switch(config-if-Lo1)# ip address 1.1.1.1/32
switch(config-if-Et1)# isis enable inst1
switch(config-if-Et1)# isis network point-to-point
```

Show commands

The same IP address that may be in use on multiple interfaces at the same time, and is displayed as shown below.

The following displays the output of show ip interface brief. In this example, Ethernet ***2-5*** are all unnumbered and borrowing from ***loopback1***.

```
switch(config-if-Et2)# show ip interface brief
Address
Interface    IP Address   Status   Protocol   MTU    Owner
-----------  -----------  -------  ---------  -------  -------
Ethernet1    1.1.2.1/24   up        up        1500
Ethernet2    1.1.1.1/32   up        up        1500     Lo1
Ethernet3    1.1.1.1/32   up        up        1500     Lo1
Ethernet4    1.1.1.1/32   up        up        1500     Lo1
Ethernet5    1.1.1.1/32   up        up        1500     Lo1
Loopback1    1.1.1.1/32   up        up        65535
```

The following displays OSPF with two adjacencies with the same peer via ***Ethernet 2*** and ***Ethernet 3***. The same **`Neighbor ID`** is listed for both interfaces. IS-IS behaves similarly.

```
switch(config-if-Et2)#show ip ospf neighbor
Neighbor ID   Instance VRF      Pri State   Dead Time  Address  Interface
2.2.1.1       1        default  0   FULL    00:00:36   2.2.1.1  Ethernet3
2.2.1.1       1        default  0   FULL    00:00:34   2.2.1.1  Ethernet2
```

Limitations

- Configuring the addresses on the lending loopbacks as ***/32*** is recommended. In order to resolve routes via an unnumbered peer, the***/32*** address is required. Configuring a lending loopback as ***/32*** and enabling OSPF/ISIS on it propagates that prefix.
- Use only loopback interfaces as a lending interface.
- Enable only one IGP on a lending loopback interface. For multiple IGPs enable each on a different loopback.
- Configure only one BFD multi-hop session per loopback.

- SSO is not supported for BFD multihop sessions over unnumbered interfaces.
- OSPFv3 does not support unnumbered interface addressing.

15.2.3.6 Configuring Interfaces for OSPFv2

OSPFv2 interface configuration commands specify transmission parameters for routed ports and SVIs that handle OSPFv2 packets.

15.2.3.6.1 Configuring Authentication

OSPFv2 authenticates packets through passwords configured on VLAN interfaces. Interfaces connecting to the same area can authenticate packets if they have the same key. By default, OSPFv2 does not authenticate packets.

OSPFv2 supports simple password and message digest authentication:

- Simple password authentication: A password is assigned to an area. Interfaces connected to the area can authenticate packets by enabling authentication and specifying the area password.
- Message digest authentication: Each interface is configured with a key (password) and key-id pair. When transmitting a packet, the interface generates a string, using the MD5 algorithm, based on the OSPFv2 packet, key, and key ID, then appends that string to the packet.

Message digest authentication supports uninterrupted transmissions during key changes by allowing each interface to have two keys with different key IDs. When a new key is configured on an interface, the router transmits OSPFv2 packets for both keys. Once the router detects that all neighbors are using the new key, it stops sending the old one.

Implementing authentication on an interface is a two step process:

1. Enabling authentication.
2. Configuring a key (password).

To configure simple authentication on a VLAN interface:

1. Enable simple authentication with the **ip ospf authentication** command.

```
switch(config-if-vl12)# ip ospf authentication
```

2. Configure the password with the **ip ospf authentication-key** command.

```
switch(config-if-vl12)# ip ospf authentication-key 0 code123
```

The ***running-config*** stores the password as an encrypted string, using a proprietary algorithm.To configure Message-Digest authentication on a VLAN interface:

1. Enable Message-Digest authentication with the **ip ospf authentication** command.

```
switch(config-if-vl12)# ip ospf authentication message-digest
```

2. Configure the key ID and password with the **ip ospf message-digest-key** command.

```
switch(config-if-vl12)# ip ospf message-digest-key 23 md5 0 code123
```

The ***running-config*** stores the password as an encrypted string, using a proprietary algorithm. The key ID (***23***) is between keywords **message-digest-key** and **md5**.

15.2.3.6.2 Configuring Intervals

Interval configuration commands determine OSPFv2 packet transmission characteristics for the specified VLAN interface and are entered in ***interface-vlan*** configuration mode.

Hello Interval

The hello interval specifies the period between consecutive hello packet transmissions from an interface. Each OSPFv2 neighbor should specify the same hello interval, which should not be longer than any neighbors dead interval.

The ip ospf hello-interval command configures the hello interval for the configuration mode interface. The default is ***10*** seconds.

Example

This command configures a hello interval of ***30*** seconds for ***VLAN 2***.

```
switch(config-if-Vl2)# ip ospf hello-interval 30
switch(config-if-Vl2)#
```

Dead Interval

The dead interval specifies the period that an interface waits for an OSPFv2 packet from a neighbor before it disables the adjacency under the assumption that the neighbor is down. The dead interval should be configured identically on all OSPFv2 neighbors and be longer than the hello interval of any neighbor.

The **ip ospf dead-interval** command configures the dead interval for the configuration mode interface. The default is ***40*** seconds.

Example

This command configures a dead interval of ***120*** seconds for ***vlan 4***.

```
switch(config-if-Vl4)# ip ospf dead-interval 120
switch(config-if-Vl4)#
```

Retransmit Interval

Routers that send OSPFv2 advertisements to an adjacent router expect to receive an acknowledgment from that neighbor. Routers that do not receive an acknowledgment will retransmit the advertisement. The retransmit interval specifies the period between retransmissions.

The **ip ospf retransmit-interval** command configures the LSA retransmission interval for the configuration mode interface. The default retransmit interval is ***5*** seconds.

Example

This command configures a retransmit interval of ***15*** seconds for ***vlan 3***.

```
switch(config-if-Vl3)# ip ospf retransmit-interval 15
switch(config-if-Vl3)#
```

Transmission Delay

The transmission delay is an estimate of the time that an interface requires to transmit a link-state update packet. OSPFv2 adds this delay to the age of outbound packets to more accurately reflect the age of the LSA when received by a neighbor. The default transmission delay is one second.

The **ip ospf transmit-delay** command configures the transmission delay for the configuration mode interface.

Example

This command configures a transmission delay of ***5*** seconds for ***vlan 6***.

```
switch(config-if-Vl6)# ip ospf transmit-delay 5
```

```
switch(config-if-Vl6)#
```

15.2.3.6.3 Configuring Interface Parameters

Interface Cost

The OSPFv2 interface cost (or metric) reflects the overhead of sending packets across the interface. The cost is typically inversely proportional to the bandwidth of the interface. The default cost is ***10***.

The **`ip ospf cost`** command configures the OSPFv2 cost for the configuration mode interface.

Example

This command configures a cost of ***15*** for ***vlan 2***.

```
switch(config-if-Vl2)# ip ospf cost 15
switch(config-if-Vl2)#
```

Router Priority

Router priority determines preference during Designated Router (DR) and Backup Designated Router (BDR) elections. Routers with higher priority numbers have preference over other routers. Routers with a priority of zero cannot be elected as a DR or BDR.

The ip ospf priority command configures router priority for the configuration mode interface. The default priority is 1.

Examples

- This command configures a router priority of ***15*** for ***vlan 8***.

```
switch(config-if-Vl8)# ip ospf priority 15
switch(config-if-Vl8)#
```

- This command restores the router priority of ***1*** for ***vlan 7***.

```
switch(config-if-Vl7)# no ip ospf priority
switch(config-if-Vl7)#
```

15.2.3.7 Enabling OSPFv2

15.2.3.7.1 Disabling OSPFv2

The switch can disable OSPFv2 operations without disrupting the OSPFv2 configuration.

- **`shutdown (OSPFv2)`** disables all OSPFv2 activity.
- **`ip ospf disabled`** disables OSPFv2 activity on a VLAN interface.

The **`no shutdown`** and **`no ip ospf disabled`** commands resume OSPFv2 activity.

Examples

- This command disables OSPFv2 activity on the switch.

```
switch(config-router-ospf)# shutdown
switch(config-router-ospf)#
```

- This command resumes OSPFv2 activity on the switch.

```
switch(config-router-ospf)# no shutdown
```

```
switch(config-router-ospf)#
```

- This command disables OSPFv2 activity on VLAN 5.

```
switch(config-if-Vl5)# ip ospf disabled
switch(config-if-Vl5)#
```

15.2.3.7.2 IPv4 Routing

OSPFv2 requires that IPv4 routing is enabled on the switch. When IP routing is not enabled, entering OSPFv2 configuration mode generates a message.

Examples

- This message is displayed if, when entering the ***router-ospf*** configuration mode, IP routing is not enabled.

```
switch(config)# router ospf 100
! IP routing not enabled
switch(config-router-ospf)#
```

- This command enables IP routing on the switch.

```
switch(config)# ip routing
switch(config)#
```

15.2.3.8 OSPFv2 Multiple Instances Support Configuration

The existing OSPFv2 configuration commands remain unchanged and are used for configuring multiple OSPFv2 instances. Each OSPFv2 instance in the default VRF is identified by a unique instance ID.

`router ospf` ***id*** [vrf | general]

15.2.3.8.1 Redistribute Configuration

Configuring the **`redistribute ospf`** command under the ***config-router-bgp*** mode with multiple OSPFv2 instances configured redistributes routes from all OSPFv2 instances into BGP.

These commands redistribute OSPFv2 routes into the BGP domain.

```
switch(config)# router bgp 1
switch(config-router-bgp)# redistribute OSPF
switch(config-router-bgp)#
```

15.2.3.8.2 Special Cases

Route Selection in case of Ties between Instances

When the same prefix happens to be learned in multiple instances with the same metric, route-type are used as the first criteria to tie break:

O > O IA > N1 > N2 > E1 > E2

```
Codes: O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2
```

When routes have identical route-type as well, the route with the lowest nexthop IP address is selected.

Note: An ECMP route is not created in this case.

Overlapping Network Statements Configured

The CLI does not guard against overlapping network statements configured in different instances. This state is a misconfiguration.

15.2.3.8.3 OSPFv2 Multiple Instances Limitations

- OSPFv2 Multiple Instances is available only with the multi-agent routing protocol model.
- Only one interface can only have one instance of OSPFv2 running at any point in time.
- All the OSPFv2 instances must be in the default VRF.
- Multiple OSPFv2 instances can not be connected to the same network or configured with interfaces in the same area. In particular, multiple OSPFv2 instances may not be connected to the same instance on another router, in the same area.
- The following features are not supported with multiple OSPFv2 instances:
  - Redistributing routes from a specific OSPFv2 instance into BGP.
  - Redistribution of routes into an OSPFv2 instance.
  - Per interface area configuration.
  - Passive interface configuration.
  - SNMP.
  - Summary address.
  - Service ACL.
  - Max Metric with **`on-startup`** configuration.

15.2.3.9 OSPF Routes over GRE Tunnels Configuration

The following commands may be employed in the router OSPF mode:

- tunnel routes
- **`no tunnel routes`**
- **`default tunnel routes`**

To enable OSPFv2 over GRE tunnels, use the following commands:

```
switch(config)# router ospf 6
switch(config-router-ospf)# tunnel routes
switch(config-router-ospf)#
```

To disable OSPFv2 routes over GRE tunnels, use the following commands:

```
switch(config)# router ospf 6
switch(config-router-ospf)# no tunnel routes
switch(config-router-ospf)#
```

To enable the default OSPFv2 routes over GRE tunnels, use the following commands:

```
switch(config)# router ospf 6
switch(config-router-ospf)# default tunnel routes
switch(config-router-ospf)#
```

15.2.3.9.1 TCAM Profile Configuration

On DCS-7020, DCS-7280R/R2, or DCS-7500R/R2 enabling OSPF routes over GRE tunnels requires the system TCAM profile to have “Tunnel IPv4” feature enabled so that control packets such as OSPF hellos received over GRE tunnel interfaces are appropriately classified. This can be achieved by creating a user defined TCAM profile as described below.

The user defined TCAM profile can be created either manually from scratch or by copying from an existing TCAM profile. The system TCAM profile must have the feature **tunnel ipv4** for the OSPFv2 over GRE tunnel interfaces to work. This is applicable regardless of whether the TCAM profile is copied from an existing profile or created from scratch.

15.2.3.9.2 User Defined PMF (or TCAM) Profiles

This section describes a set of CLI commands to create user defined PMF (or TCAM) profile. The profile is composed of a set of TCAM features, with each feature having customized lookup key, actions and packet types to hit.

All TCAM profile CLIs are under **hardware tcam** mode.

```
(config)# hardware tcam
(config-hw-tcam)#
```

There are two ways to create a TCAM profile. The recommended way is to create a profile based on an existing one. In this example, a copy of the profile ***newprofile1*** is created.

```
(config)# hardware tcam
(config-hw-tcam)# profile newprofile1 copy default
(config-hw-tcam-profile-newprofile1)#
```

The other way is to create one profile from scratch. In this example, the profile ***newprofile2*** is created.

```
(config)# hardware tcam
(config-hw-tcam)# profile newprofile2
(config-hw-tcam-profile-newprofile2)#
```

To remove a profile, use the **no profile** command similar to the following. In this example, the profile ***newprofile2*** is removed.

```
(config)# hardware tcam
(config-hw-tcam)# no profile newprofile2
```

Features can be turned on and off in the new profile. In this example, the feature **acl port ipv6** is turned on.

```
(config-hw-tcam-profile-<profile>)# feature acl port ipv6
```

Features can be turned on and off in the new profile. In this example, the feature **acl port ipv6** is turned off.

```
(config-hw-tcam-profile-<profile>)# no feature acl port ipv6
```

The features are described by hierarchical CLI tokens. For example, IPv4 port ACL is represented by **feature acl port ip** and IPv6 port ACL is represented by **feature acl port ipv6**. Under each feature mode, there are various fields that can be modified.

- **packet**

  This describes packet types that the feature will be applied on.

  packet ***packet header tokens*** **forwarding [bridged | routed | mpls][multicast][decap]**

`no packet` *packet header tokens* forwarding [bridged | routed | mpls][multicast][decap]

The packet header is described a series of CLI packet header tokens after **packet** token. It starts from the outer most header after Ethernet. For example, a regular IPv4 packet is ***packet ipv4*** and a vxlan packet is ***packet ipv4 vxlan eth ipv4***. The **forwarding** token indicates the forwarding type of the packet. **multicast** indicates if the packet is a multicast packet. Lastly, **decap** indicates if the packet is decapsulated after a tunnel.

- **`key field`**

  This describes the TCAM key format for the feature. The CLI can add or delete fields that are used to build the key.

  ```
  (config-hw-tcam-profile-<profile>-feature-<feature>)# [no]keyfield field
  ```

  All supported key fields can be found with **`key field ?`**

- **`key size`**

  This describes the TCAM key size limit. If too many key fields are added to the feature so that the key size goes beyond the limit, a Syslog will be issued. The default key size limit is ***320***.

  ```
  (config-hw-tcam-profile-<profile>-feature-<feature>)# [no]key size
   limit size
  ```

- **`action`**

  This describes the action to take if a TCAM entry is hit.

  ```
  (config-hw-tcam-profile-<profile>-feature-<feature>)# [no]action action
  ```

  The supported actions can be found through **`action ?`**.

- **`sequence`**

  This describes the programming order of each feature. Changing the order may affect the programming status of a profile. The default sequence is ***0***.

  ```
  (config-hw-tcam-profile-<profile>-feature-<feature>)# [no]sequence sequence
  ```

  The profile is saved after exiting the feature mode. To use the newly defined profile, a CLI is available to apply the profile to the system globally.

  ```
  (config)# hardware tcam
  (config-hw-tcam)# system profile newprofile1
  ```

15.2.3.10 Displaying OSPFv2 Status

This section describes OSPFv2 **`show`** commands that display OSPFv2 status. General switch methods that provide OSPFv2 information include pinging routes, viewing route status (**`show ip route`** command), and viewing the configuration (**`show running-config`** command).

15.2.3.10.1 OSPFv2 Summary

The **`show ip ospf`** command displays general OSPFv2 configuration information and operational statistics.

Example

This command displays general OSPFv2 information.

```
switch# show ip ospf
Routing Process "ospf 1" with ID 10.168.103.1
```

```
Supports opaque LSA
Maximum number of LSA allowed 12000
Threshold for warning message 75%
Ignore-time 5 minutes, reset-time 5 minutes
Ignore-count allowed 5, current 0
It is an area border router
Hold time between two consecutive SPFs 5000 msecs
SPF algorithm last executed 00:00:09 ago
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msecs
Number of external LSA 0. Checksum Sum 0x000000
Number of opaque AS LSA 0. Checksum Sum 0x000000
Number of LSA 27.
Number of areas in this router is 3. 3 normal 0 stub 0 nssa
Area BACKBONE(0.0.0.0)
Number of interfaces in this area is 2
It is a normal area
Area has no authentication
SPF algorithm executed 153 times
Number of LSA 8. Checksum Sum 0x03e13a
Number of opaque link LSA 0. Checksum Sum 0x000000
Area 0.0.0.2
Number of interfaces in this area is 1
It is a normal area
Area has no authentication
SPF algorithm executed 153 times
Number of LSA 11. Checksum Sum 0x054e57
Number of opaque link LSA 0. Checksum Sum 0x000000
Area 0.0.0.3
Number of interfaces in this area is 1
It is a normal area
Area has no authentication
SPF algorithm executed 5 times
Number of LSA 6. Checksum Sum 0x02a401
Number of opaque link LSA 0. Checksum Sum 0x000000
```

The output lists configuration parameters and operational statistics and status for the OSPFv2 instance, followed by a brief description of the areas located on the switch.

15.2.3.10.2 Viewing OSPFv2 on the Interfaces

The **show ip ospf interface** command displays OSPFv2 information for switch interfaces configured for OSPFv2. Different command options allow the display of either all interfaces or a specified interface. The command can also be configured to display complete information or a brief summary.

Examples

- This command displays complete OSPFv2 information for ***vlan 1***.

```
switch# show ip ospf interface vlan 1
Vlan1 is up, line protocol is up (connected)
Internet Address 10.168.0.1/24, Area 0.0.0.0
Process ID 1, Router ID 10.168.103.1, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router is 10.168.104.2
Backup Designated router is 10.168.103.1
Timer intervals configured, Hello 10, Dead 40, Retransmit 5
Neighbor Count is 1
MTU is 1500 switch#
```

The display indicates the switch is an ABR by displaying a neighbor count, the Designated Router (DR), and Backup Designated Router (BDR).

- This command displays a summary of interface information for the switch.

```
switch# show ip ospf interface brief
InterfacePIDAreaIP AddressCostStateNbrs
Loopback010.0.0.010.168.103.1/2410DR0
Vlan110.0.0.010.168.0.1/2410BDR1
Vlan210.0.0.210.168.2.1/2410BDR1
Vlan310.0.0.310.168.3.1/2410DR0 switch#
```

Configuration information includes the Process ID (PID), area, IP address, and cost. OSPFv2 operational information includes the Designated Router status and number of neighbors.

15.2.3.10.3 Viewing the OSPFv2 Database

The **show ip ospf database <link state list>** command displays the LSAs in the LSDB for the specified area. If no area is listed, the command displays the contents of the database for each area on the switch. The database command provides options to display subsets of the LSDB database, a summary of database contents, and the link states that comprise the database.

Examples

- This command displays LSDB contents for ***area 2***.

```
switch# show ip ospf 1 2 database

OSPF Router with ID(10.168.103.1)(Process ID 1)

Router Link States (Area 0.0.0.2)

Link IDADV RouterAgeSeq#Checksum Link count
10.168.103.110.168.103.100:29:080x80000031 0x001D5F 1
10.168.104.210.168.104.200:29:090x80000066 0x00A49B 1

Net Link States (Area 0.0.0.2)

Link IDADV RouterAgeSeq#Checksum
10.168.2.110.168.103.100:29:080x80000001 0x00B89D

Summary Net Link States (Area 0.0.0.2)

Link IDADV RouterAgeSeq#Checksum
10.168.0.010.168.103.100:13:200x80000028 0x0008C8
10.168.0.010.168.104.200:09:160x80000054 0x00A2FF
10.168.3.010.168.104.200:24:160x80000004 0x00865F
10.168.3.010.168.103.100:24:200x80000004 0x002FC2
10.168.103.010.168.103.100:14:200x80000028 0x0096D2
10.168.103.010.168.104.200:13:160x80000004 0x00364B
10.168.104.010.168.104.200:08:160x80000055 0x002415
10.168.104.010.168.103.100:13:200x80000028 0x00EF6E switch#
```

- This command displays an LSDB content summary for ***area 2***.

```
switch# show ip ospf 1 2 database database-summary

OSPF Router with ID(10.168.103.1) (Process ID 1)

Area 0.0.0.2 database summary
LSA TypeCount
Router2
Network1
Summary Net8
Summary ASBR0
```

```
Type-7 Ext0
Opaque Area0
Subtotal11

Process 1 database summary
LSA TypeCount
Router2
Network1
Summary Net8
Summary ASBR0
Type-7 Ext0
Opaque Area0
Type-5 Ext0
Opaque AS0
Total11 switch#
```

- This command displays the router Link States contained in the ***area 2*** LSDB.

```
switch# show ip ospf 1 2 database router

OSPF Router with ID(10.168.103.1) (Process ID 1)

Router Link States (Area 0.0.0.2)

LS age: 00:02:16
Options: (E DC)
LS Type: Router Links
Link State ID: 10.168.103.1
Advertising Router: 10.168.103.1
LS Seq Number: 80000032
Checksum: 0x1B60
Length: 36
Number of Links: 1

Link connected to: a Transit Network
(Link ID) Designated Router address: 10.168.2.1
(Link Data) Router Interface address: 10.168.2.1
Number of TOS metrics: 0
TOS 0 Metrics: 10


LS age: 00:02:12
Options: (E DC)
LS Type: Router Links
Link State ID: 10.168.104.2
Advertising Router: 10.168.104.2
LS Seq Number: 80000067
Checksum: 0xA29C
Length: 36
Number of Links: 1

Link connected to: a Transit Network
(Link ID) Designated Router address: 10.168.2.1
(Link Data) Router Interface address: 10.168.2.2
Number of TOS metrics: 0
TOS 0 Metrics: 10 switch#
```

15.2.3.10.4 Viewing OSPFv2 Neighbors

The **show ip ospf neighbor** command displays information about the routers that are neighbors to the switch. Command options allow the display of summary or detailed information about the neighbors

for all areas and interfaces on the switch. The command also allows the display of neighbors for individual interfaces or areas. The ***adjacency-changes*** option displays the interfaces adjacency changes.

Examples

- This command displays the switchs neighbors.

```
switch# show ip ospf neighbor
Neighbor IDPriStateDead TimeAddressInterface
10.168.104.21FULL/DR00:00:3510.168.0.2Vlan1
10.168.104.28FULL/BDR00:00:3110.168.2.2Vlan2 switch#
```

- This command displays details about the neighbors to ***vlan 2***.

```
switch# show ip ospf neighbor vlan 2 detail
 Neighbor 10.168.104.2, interface address 10.168.2.2
In the area 0.0.0.2 via interface Vlan2
Neighbor priority is 8, State is FULL, 13 state changes
Adjacency was established 000:01:25:48 ago
DR is 10.168.2.1 BDR is 10.168.2.2
Options is E
Dead timer due in 00:00:34 switch#
```

- This command displays the adjacency changes to ***vlan 2***.

```
switch# show ip ospf neighbor vlan 2 adjacency-changes
[08-04 08:55:32] 10.168.104.2, interface Vlan2 adjacency established
[08-04 09:58:51] 10.168.104.2, interface Vlan2 adjacency dropped: interface
 went
down
[08-04 09:58:58] 10.168.104.2, interface Vlan2 adjacency established
[08-04 09:59:34] 10.168.104.2, interface Vlan2 adjacency dropped: interface
 went
down
[08-04 09:59:42] 10.168.104.2, interface Vlan2 adjacency established
[08-04 10:01:40] 10.168.104.2, interface Vlan2 adjacency dropped: nbr did
 not
list our router ID
[08-04 10:01:46] 10.168.104.2, interface Vlan2 adjacency established switch#
```

The **`show ip ospf neighbor state`** command displays the state information for OSPF neighbors on a per-interface basis.

Example

This command displays OSPF information for neighboring routers that are fully adjacent.

```
switch# show ip ospf neighbor state full
Neighbor ID     VRF     Pri   State           Dead Time   Address         Interface
Test1           default     1   FULL/BDR          00:00:35    10.17.254.105  Vlan3912
Test2           default     1   FULL/BDR          00:00:36    10.17.254.29   Vlan3910
Test3 default     1   FULL/DR 00:00:35 10.25.0.1      Vlan101
Test4           default     1   FULL/DROTHER      00:00:36    10.17.254.67   Vlan3908
Test5           default     1   FULL/DROTHER      00:00:36    10.17.254.68   Vlan3908
Test6           default     1   FULL/BDR          00:00:32    10.17.254.66   Vlan3908
Test7           default     1   FULL/DROTHER      00:00:34    10.17.36.4     Vlan3036
Test8           default     1   FULL/BDR          00:00:35    10.17.36.3     Vlan3036
Test9           default     1   FULL/DROTHER      00:00:31    10.17.254.13   Vlan3902
Test10          default     1   FULL/BDR          00:00:37    10.17.254.11   Vlan3902
Test11          default     1   FULL/DROTHER      00:00:33    10.17.254.163  Vlan3925
Test12          default     1   FULL/DR           00:00:37    10.17.254.161  Vlan3925
Test13          default     1   FULL/DROTHER      00:00:31    10.17.254.154  Vlan3923
Test14          default     1   FULL/BDR          00:00:39    10.17.254.156  Vlan3923
Test15          default     1   FULL/DROTHER      00:00:33    10.17.254.35   Vlan3911
Test16          default     1   FULL/DR           00:00:34    10.17.254.33   Vlan3911
Test17 default     1   FULL/DR          00:00:36    10.17.254.138  Ethernet12
Test18          default     1   FULL/DR           00:00:37    10.17.254.2    Vlan3901 switch>
```

The **show ip ospf neighbor summary** command displays a single line of summary information for each OSPFv2 neighbor.

Example

This command displays the summary information for the OSPFv2 neighbors.

```
switch# show ip ospf neighbor summary
OSPF Router with (Process ID 1) (VRF default)
0 neighbors are in state DOWN
0 neighbors are in state GRACEFUL RESTART
2 neighbors are in state INIT
0 neighbors are in state LOADING
0 neighbors are in state ATTEMPT
18 neighbors are in state FULL
0 neighbors are in state EXCHANGE
0 neighbors are in state 2 WAYS
0 neighbors are in state EXCH START switch>
```

15.2.3.10.5 Viewing OSPFv2 Routes

The **show ip routes** command provides an OSPFv2 option.

Examples

- This command displays all of a switchs routes.

```
switch# show ip route
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP

Gateway of last resort:
S0.0.0.0/0 [1/0] via 10.255.255.1
C10.255.255.0/24 is directly connected, Management1
C10.168.0.0/24 is directly connected, Vlan1
C10.168.2.0/24 is directly connected, Vlan2
O10.168.3.0/24 [110/20] via 10.168.0.1
O10.168.103.0/24 [110/20] via 10.168.0.1
C10.168.104.0/24 is directly connected, Loopback0 switch#
```

- This command displays the switchs OSPFv2 routes.

```
switch# show ip route ospf
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP

O10.168.3.0/24 [110/20] via 10.168.0.1
O10.168.103.0/24 [110/20] via 10.168.0.1 switch#
```

Use the **ping** command to determine the accessibility of a route.

Example

This command pings an OSPFv2 route.

```
switch# ping 10.168.0.1
PING 10.168.0.1 (10.168.0.1) 72(100) bytes of data.
80 bytes from 10.168.0.1: icmp_seq=1 ttl=64 time=0.148 ms
80 bytes from 10.168.0.1: icmp_seq=2 ttl=64 time=0.132 ms
80 bytes from 10.168.0.1: icmp_seq=3 ttl=64 time=0.136 ms
80 bytes from 10.168.0.1: icmp_seq=4 ttl=64 time=0.137 ms
80 bytes from 10.168.0.1: icmp_seq=5 ttl=64 time=0.136 ms

--- 10.168.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 7999ms
```

```
rtt min/avg/max/mdev = 0.132/0.137/0.148/0.015 ms switch#
```

15.2.3.10.6 Viewing OSPFv2 SPF Logs

The **show ip ospf spf-log** command displays when and how long the switch took to run a full SPF calculation for OSPF.

Example

This command displays the SPF information for OSPF.

```
switch# show ip ospf spf-log
OSPF Process 172.26.0.22
When      Duration(msec)
13:01:34  1.482
13:01:29  1.547
13:01:24  1.893
13:00:50  1.459
13:00:45  1.473
13:00:40  2.603
11:01:49  1.561
11:01:40  1.463
11:01:35  1.467
11:01:30  1.434
11:00:54  1.456
11:00:49  1.472
11:00:44  1.582
15:01:49  1.575
15:01:44  1.470
15:01:39  1.679
15:01:34  1.601
15:00:57  1.454
15:00:52  1.446
15:00:47  1.603 switch>
```

15.2.3.10.7 Viewing OSPFv2 multiple Instances Support

The **show ip ospf** commands will take an instance ID filter to get the information for a particular OSPFv2 instance. If no instance ID is specified in the show query, information for all the active OSPFv2 instances are shown.

The **show ip ospf** commands will also display instance ID along with router ID either in the output headers or as a separate column.

Sample output for the **show ip ospf** command with two OSPFv2 instances with ***ID 1*** and ***ID 2***.

```
switch# show ip ospf
OSPF instance 1 with ID 1.1.1.1 VRF default
 Supports opaque LSA
Maximum number of LSA allowed 12000
  Threshold for warning message 75%
  Ignore-time 5 minutes, reset-time 5 minutes
  Ignore-count allowed 5, current 0
 It is not an autonomous system boundary router and is not an area
 border router
 ...
OSPF instance 2 with ID 2.2.2.2 VRF default
 Supports opaque LSA
Maximum number of LSA allowed 12000
  Threshold for warning message 75%
```

```
  Ignore-time 5 minutes, reset-time 5 minutes
  Ignore-count allowed 5, current 0
 It is not an autonomous system boundary router and is not an area
 border router
...
```

Sample output for the **`show ip ospf`** command with Graceful Restart enabled for two OSPFv2 instances with ID ***10*** and ***11***.

```
switch# show ip ospf
 OSPF instance 10 with ID 2.2.2.2 VRF default
 Supports opaque LSA
Maximum number of LSA allowed 12000
  Threshold for warning message 75%
  Ignore-time 5 minutes, reset-time 5 minutes
...
 Graceful-restart is configured, grace-period 120 seconds
   State: In progress, expires in 113 seconds
 Graceful-restart-helper mode is enabled
...
 OSPF instance 11 with ID 3.3.3.3 VRF default
 Supports opaque LSA
Maximum number of LSA allowed 12000
  Threshold for warning message 75%
  Ignore-time 5 minutes, reset-time 5 minutes
...
 Graceful-restart is configured, grace-period 120 seconds
   State: In progress, expires in 113 seconds
 Graceful-restart-helper mode is enabled
...
```

Sample output for the **`show ip ospf neighbor detail`** command.

```
switch# show ip ospf neighbor
Neighbor ID   Instance  VRF      Pri  State      Dead Time   Address   Interface
2.2.2.2       1         default  1    FULL/DR    00:00:38    10.1.1.2   Ethernet1
4.4.4.4       2         default  1    FULL/DR    00:00:36    40.1.1.2   Ethernet4

switch# show ip ospf neighbor 2.2.2.2 detail
Neighbor 2.2.2.2, instance 1, VRF default, interface address 10.1.1.1

  In area 0.0.0.0 interface Ethernet1

  Neighbor priority is 1, State is FULL, 7 state changes

  Adjacency was established 00:38:48 ago

  Current state was established 00:38:48 ago

  DR IP Address 10.1.1.2 BDR IP Address 10.1.1.1

  Options is E

  Dead timer is due in 00:00:35

  Inactivity timer deferred 0 times

  LSAs retransmitted 1 time to this neighbor

  Graceful-restart-helper mode is Inactive
```

```
  Graceful-restart attempts: 0
```

Sample output for **show ip ospf neighbor detail** with BFD enabled.

```
switch# show ip ospf neighbor 2.2.2.2 detail
Neighbor 3.3.3.3, instance 10, VRF default, interface address 1.0.0.1
  In area 1.2.3.4 interface Ethernet1
  Neighbor priority is 1, State is FULL, 7 state changes
  Adjacency was established 22:03:05 ago
  Current state was established 22:03:05 ago
  DR IP Address 1.0.0.1 BDR IP Address 1.0.0.2
  Options is E
  Dead timer is due in 00:00:34
  Inactivity timer deferred 0 times
  LSAs retransmitted 1 time to this neighbor
  Bfd request is sent and the state is Down
  Graceful-restart-helper mode is Inactive
  Graceful-restart attempts: 0
Neighbor 6.6.6.6, instance 10, VRF default, interface address 1.0.1.1
  In area 1.2.3.4 interface Ethernet5
  Neighbor priority is 1, State is FULL, 7 state changes
  Adjacency was established 22:03:10 ago
  Current state was established 22:03:10 ago
  DR IP Address 1.0.1.1 BDR IP Address 1.0.1.2
  Options is E
  Dead timer is due in 00:00:30
  Inactivity timer deferred 0 times
  LSAs retransmitted 2 times to this neighbor
  Bfd request is sent and the state is Down
  Graceful-restart-helper mode is Inactive
  Graceful-restart attempts: 0
Neighbor 4.4.4.4, instance 12, VRF default, interface address 1.0.3.1
  In area 1.2.3.4 interface Ethernet2
  Neighbor priority is 1, State is FULL, 7 state changes
  Adjacency was established 22:03:10 ago
  Current state was established 22:03:10 ago
  DR IP Address 1.0.3.1 BDR IP Address 1.0.3.2
  Options is E
  Dead timer is due in 00:00:32
  Inactivity timer deferred 0 times
  LSAs retransmitted 1 time to this neighbor
  Graceful-restart-helper mode is Inactive
  Graceful-restart attempts: 0
```

The CAPI outputs for OSPFv2 show commands are already indexed by instance ID and remains unchanged.

The **show ip route** and **show ip route ospf** commands show routes from all OSPFv2 instances with no mention of instance ID. For example,

11.1.1.0/24 is learned from instance ***100*** and ***12.1.1.0/24*** from instance ***200***.

```
switch# show ip route

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
```

```
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

Gateway of last resort is not set

 O E2    11.1.1.0/24 [110/1] via 20.1.1.2, Ethernet3
 C       10.1.1.0/24 is directly connected, Ethernet1
 C       20.1.1.0/24 is directly connected, Ethernet3
 O       12.1.1.0/24 [110/20] via 10.1.1.2, Ethernet1

switch# show ip route ospf

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

 O E2    11.1.1.0/24 [110/1] via 20.1.1.2, Ethernet3
 O       12.1.1.0/24 [110/20] via 10.1.1.2, Ethernet1
```

The **show ip route summary** command displays the cumulative counts of OSPFv2 routes across all instances.

```
switch# show ip route summary

VRF: default
   Route Source                                  Number Of Routes
----------------------------------- ------------------------
   connected                                                2
   static (persistent)                                      0
   static (non-persistent)                                  0
   VXLAN Control Service                                    0
   static nexthop-group                                     0
 ospf                                                     9
     Intra-area: 2 Inter-area: 5 External-1: 0 External-2: 2
     NSSA External-1: 0 NSSA External-2: 0
   ospfv3                                                   0
   bgp                                                      0
     External: 0 Internal: 0
   isis                                                     0
     Level-1: 0 Level-2: 0
   rip                                                      0
   internal                                                 9
   attached                                                 1
   aggregate                                                0
   dynamic policy                                           0

   Total Routes                                            14
```

```
Number of routes per mask-length:
   /8: 2          /24: 3         /32: 9
```

15.2.3.10.8 Displaying OSPF Routes over GRE Tunnel Status

A show command is available to list the TCAM profile status on each linecard.

```
(config)# show hardware tcam profile
Configuration  Status FixedSystem  newprofile1  newprofile1
```

If the profile cannot be programmed, the Status column will print **ERROR**. See **Limitations** for additional information. The content of a PMF profile can be displayed with

```
(config)# show hardware tcam profile detail
```

Example

```
(config-hw-tcam)# show hardware tcam profile newprofile1 detail
Profile newprofile1 [ FixedSystem ]
 Feature         mpls
 --------------- ----------------------------------------------------
 Key size        160
 Actions         drop, redirect, set-ecn
 Packet type     ipv4 mpls ipv4 forwarding mpls decap
                 ipv4 mpls ipv6 forwarding mpls decap
                 mpls ipv4 forwarding mpls
                 mpls ipv6 forwarding mpls
                 mpls non-ip forwarding mpls

 Feature         acl vlan ipv6
 --------------- ----------------------------------------------------
 Key size        320
 Key fields      dst-ipv6, ipv6-next-header, l4-dst-port, l4-src-port,
                 src-ipv6-high, src-ipv6-low, tcp-control
 Actions         count, drop, mirror, redirect
 Packet type     ipv6 forwarding routed
...
```

Note that the profile contains all the features that are untouched after copying from the base profile.

Example

This example demonstrates how to create a new profile to match on **vlan** field in MAC ACL.

```
(config-hw-tcam)# profile macvlan copy default
(config-hw-tcam-profile-macvlan)# feature acl port mac
(config-hw-tcam-profile-macvlan-feature-acl-port-mac)# key field vlan
(config-hw-tcam-profile-macvlan-feature-acl-port-mac)# exit
(config-hw-tcam-profile-macvlan)# exit
Saving new profile 'macvlan'
(config-hw-tcam)# system profile macvlan
```

15.2.4 OSPFv2 Configuration Examples

This section describes the commands required to configure three OSPFv2 topologies.

- OSPFv2 Configuration Example 1

- OSPFv2 Configuration Example 2
- OSPFv2 Configuration Example 3

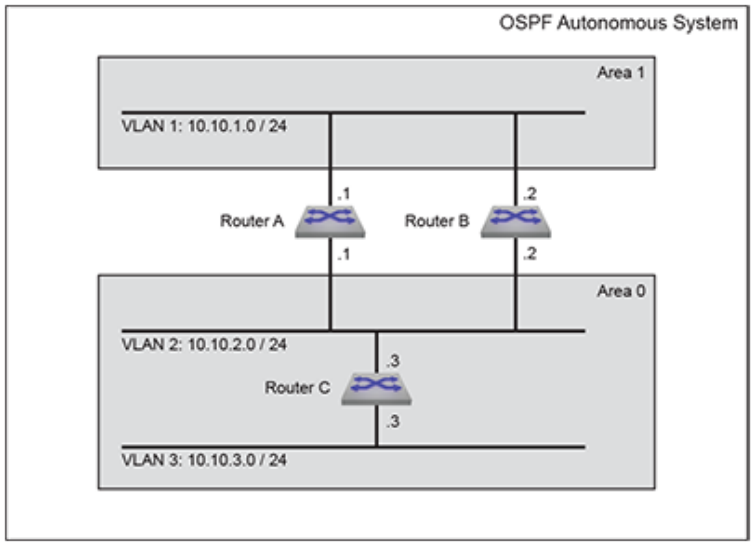
15.2.4.1 OSPFv2 Configuration Example 1

The OSPF Autonomous System in Example 1 contains two areas that are connected through two routers. The backbone area also contains an internal router that connects two subnets.

15.2.4.1.1 Example 1 Topology

OSPFv2 Example 1 displays the Example 1 topology. Two ABRs connect area ***0*** and area ***1*** ***Router A*** and ***Router B***. ***Router C*** is an internal router that connects two subnets in area ***0***.

Figure 15-3: OSPFv2 Example 1



Area 1 Configuration

Area 1 contains one subnet that is accessed by ***Router A*** and ***Router B***.

- ***Router A***: The subnet ***10.10.1.0/24*** is accessed through ***VLAN 1***.
- ***Router B***: The subnet ***10.10.1.0/24*** is accessed through ***VLAN 1***.
- Each router uses simple authentication, with password ***abcdefgh***.
- Designated Router (DR): ***Router A***.
- Backup Designated Router (BDR): ***Router B***.
- Each router defines an interface cost of ***10***.
- Router priority is not specified for either router on area ***1***.

Area 0 ABR Configuration

Area 0 contains one subnet that is accessed by ABRs ***Router A*** and ***Router B***.

- ***Router A***: The subnet ***10.10.2.0/24*** is accessed through ***VLAN 2***.
- ***Router B***: The subnet ***10.10.2.0/24*** is accessed through ***VLAN 2***.
- Designated Router (DR): ***Router B***.
- Backup Designated Router (BDR): ***Router A***.
- Each router uses simple authentication, with password ***ijklmnop***.
- Each router defines an interface cost of ***20***.
- Each router defines a retransmit-interval of ***10***.
- Each router defines a transmit-delay of ***2***.

- Router priority is specified such that ***Router B*** will be elected as the Designated Router.

Area 0 IR Configuration

Area ***0*** contains one internal router that connects two subnets.

- ***Router C***: The subnet ***10.10.2.0/24*** is accessed through ***VLAN 2***.
- ***Router C***: The subnet ***10.10.3.0/24*** is accessed through ***VLAN 3***.
- The subnet ***10.10.2.0/24*** link is configured as follows:
  - Interface cost of ***20***.
  - Retransmit-interval of ***10***.
  - Transmit-delay of ***2***.
- The subnet ***10.10.3.0/24*** link is configured as follows:
  - Interface cost of ***20***.
  - Dead interval of ***80*** seconds.

15.2.4.1.2 Example 1 Code

This code configures the OSPFv2 instances on the three switches.

1. Configure the interface addresses.
   a. Router A interfaces:

```
switch-A(config)# interface vlan 1
switch-A(config-if-vl1)# ip address 10.10.1.1/24
switch-A(config-if-vl1)# interface vlan 2
switch-A(config-if-vl2)# ip address 10.10.2.1/24
```

   b. Router B interfaces:

```
switch-B(config)# interface vlan 1
switch-B(config-if-vl1)# ip address 10.10.1.2/24
switch-B(config-if-vl1)# interface vlan 2
switch-B(config-if-vl2)# ip address 10.10.2.2/24
```

   c. Router C interfaces:

```
switch-C(config)# interface vlan 2
switch-C(config-if-vl2)# ip address 10.10.2.3/24
switch-C(config-if-vl2)# interface vlan 3
switch-C(config-if-vl3)# ip address 10.10.3.3/24
```

2. Configure the interface OSPFv2 parameters.
   a. Router A interfaces:

```
switch-A(config-if-vl2)# interface vlan 1
switch-A(config-if-vl1)# ip ospf authentication-key abcdefgh
switch-A(config-if-vl1)# ip ospf cost 10
switch-A(config-if-vl1)# ip ospf priority 6
switch-A(config-if-vl1)# interface vlan 2
switch-A(config-if-vl2)# ip ospf authentication-key ijklmnop
switch-A(config-if-vl2)# ip ospf cost 20
switch-A(config-if-vl2)# ip ospf retransmit-interval 10
switch-A(config-if-vl2)# ip ospf transmit-delay 2
switch-A(config-if-vl2)# ip ospf priority 4
```

b. Router B interfaces:

```
switch-B(config-if-vl2)# interface vlan 1
switch-B(config-if-vl1)# ip ospf authentication-key abcdefgh
switch-B(config-if-vl1)# ip ospf cost 10
switch-B(config-if-vl1)# ip ospf priority 4
switch-B(config-if-vl1)# interface vlan 2
switch-B(config-if-vl2)# ip ospf authentication-key ijklmnop
switch-B(config-if-vl2)# ip ospf cost 20
switch-B(config-if-vl2)# ip ospf retransmit-interval 10
switch-B(config-if-vl2)# ip ospf transmit-delay 2
switch-B(config-if-vl2)# ip ospf priority 6
```

c. Router C interfaces:

```
switch-C(config-if-vl3)# interface vlan 2
switch-C(config-if-vl2)# ip ospf cost 20
switch-C(config-if-vl2)# ip ospf retransmit-interval 10
switch-C(config-if-vl2)# ip ospf transmit-delay 2
switch-C(config-if-vl2)# interface vlan 3
switch-C(config-if-vl3)# ip ospf cost 20
switch-C(config-if-vl3)# ip ospf dead-interval 80
```

3. Attach the network segments to the areas.

a. Router A interfaces:

```
switch-A(config-if-vl2)# router ospf 1
switch-A(config-router-ospf)# router-id 169.10.0.1
switch-A(config-router-ospf)# network 10.10.1.0/24 area 1
switch-A(config-router-ospf)# network 10.10.2.0/24 area 0
```

b. Router B interfaces:

```
switch-B(config-if-vl2)# router ospf 1
switch-B(config-router-ospf)# router-id 169.10.0.2
switch-B(config-router-ospf)# network 10.10.1.0/24 area 1
switch-B(config-router-ospf)# network 10.10.2.0/24 area 0
```

c. Router C interfaces:

```
switch-C(config-if-vl3)# router ospf 1
switch-C(config-router-ospf)# router-id 169.10.0.3
switch-C(config-router-ospf)# network 10.10.2.0/24 area 0
switch-C(config-router-ospf)# network 10.10.3.0/24 area 0
```

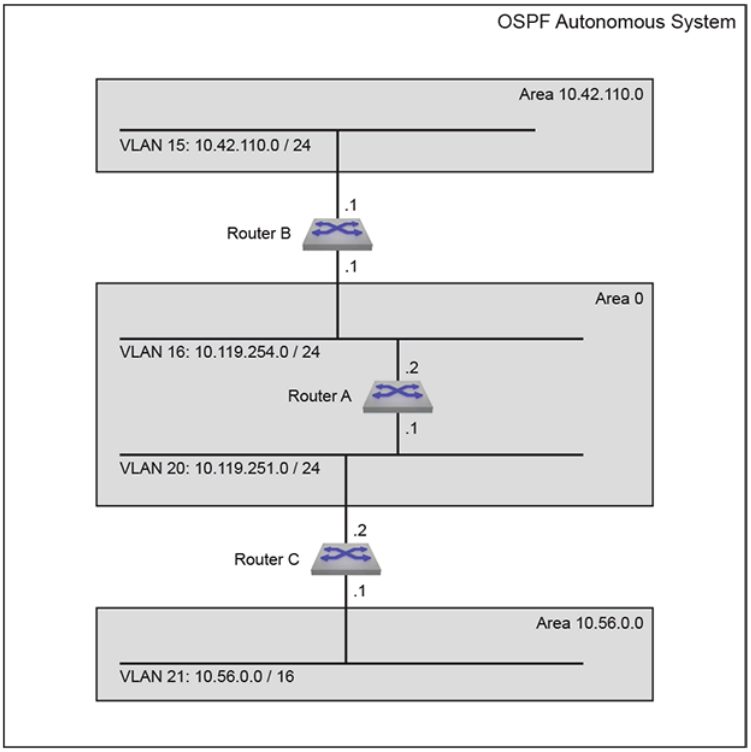
15.2.4.2 OSPFv2 Configuration Example 2

The AS in Example 2 contains three areas. Area ***0*** connects to the other areas through different routers. The backbone area contains an internal router that connects two subnets. Area ***0*** is normal; the other areas are stub areas.

15.2.4.2.1 Example 2 Topology

OSPFv2 Example 2 displays the Example 2 topology. One ABR (***Router B***) connects area ***0*** and area ***10.42.110.0***; another ABR (***Router C***) connects area ***0*** and area ***36.56.0.0***. ***Router A*** is an internal router that connects two subnets in area ***0***.

Figure 15-4: OSPFv2 Example 2



Area 10.42.110.0 Configuration

Area ***10.42.110.0*** contains one subnet that is accessed by ***Router B***.

- ***Router B***: The subnet ***10.42.110.0*** is accessed through ***VLAN 15***.
- ***Router B*** uses simple authentication, with password ***abcdefgh***.
- Each router defines a interface cost of ***10***.

Area 10.56.0.0 Configuration

Area ***10.56.0.0*** contains one subnet that is accessed by ***Router C***.

- ***Router C***: The subnet ***10.56.0.0*** is accessed through ***VLAN 21***.
- ***Router C*** uses simple authentication, with password ***ijklmnop***.
- Each router defines a interface cost of ***20***.

Area 0 ABR Configuration

Area ***0*** contains two subnets. ABR ***Router B*** connects one subnet to area ***10.42.110.0***. ABR ***Router C*** connects the other subnet to area ***10.56.0.0***.

- ***Router B***: The subnet ***10.119.254.0/24*** is accessed through ***VLAN 16***.
- ***Router C***: The subnet ***10.119.251.0/24*** is accessed through ***VLAN 20***.
- Designated Router (DR): ***Router B***.
- Backup Designated Router (BDR): ***Router C***.
- Each ABR uses simple authentication, with password ***ijklmnop***.
- Each router defines an interface cost of ***20***.
- Each router defines a retransmit-interval of ***10***.
- Each router defines a transmit-delay of ***2***.

Area 0 IR Configuration

Area ***0*** contains two subnets connected by an internal router.

- ***Router A***: The subnet ***10.119.254.0/24*** is accessed through ***VLAN 16***.
- ***Router A***: The subnet ***10.119.251.0/24*** is accessed through ***VLAN 20***.
- The subnet ***10.42.110.0*** is configured as follows:
  - Interface cost of ***10***.
- The subnet ***10.56.0.0/24*** is configured as follows:
  - Interface cost of ***20***.
  - Retransmit-interval of ***10***.
  - Transmit-delay of ***2***.

15.2.4.2.2 Example 2 Code

1. Configure the interface addresses.
   a. Router A interfaces:

```
switch-A(config)# interface vlan 16
switch-A(config-if-vl16)# ip address 10.119.254.2/24
switch-A(config-if-vl16)# interface vlan 20
switch-A(config-if-vl20)# ip address 10.119.251.1/24
```

   b. Router B interfaces:

```
switch-B(config)# interface vlan 15
switch-B(config-if-vl15)# ip address 10.42.110.1/24
switch-B(config-if-vl15)# interface vlan 16
switch-B(config-if-vl16)# ip address 10.119.254.1/24
```

   c. Router C interfaces:

```
switch-C(config)# interface vlan 20
switch-C(config-if-vl20)# ip address 10.119.251.2/24
switch-C(config-if-vl20)# interface vlan 21
switch-C(config-if-vl21)# ip address 10.56.0.1/24
```

2. Configure the interface OSPFv2 parameters.
   a. Router A interfaces:

```
switch-A(config-if-vl20)# interface vlan 16
switch-A(config-if-vl16)# ip ospf cost 10
switch-A(config-if-vl16)# interface vlan 20
switch-A(config-if-vl20)# ip ospf cost 20
switch-A(config-if-vl20)# ip ospf retransmit-interval 10
switch-A(config-if-vl20)# ip ospf transmit-delay 2
```

   b. Router B interfaces:

```
switch-B(config-if-vl16)# interface vlan 15
switch-B(config-if-vl15)# ip ospf authentication-key abcdefgh
switch-B(config-if-vl15)# ip ospf cost 10
switch-B(config-if-vl15)# interface vlan 16
switch-B(config-if-vl16)# ip ospf authentication-key ijklmnop
switch-B(config-if-vl16)# ip ospf cost 20
switch-B(config-if-vl16)# ip ospf retransmit-interval 10
switch-B(config-if-vl16)# ip ospf transmit-delay 2
switch-B(config-if-vl16)# ip ospf priority 6
```

c. Router C interfaces:

```
switch-C(config-if-vl21)# interface vlan 20
switch-C(config-if-vl20)# ip ospf authentication-key ijklmnop
switch-C(config-if-vl20)# ip ospf cost 20
switch-C(config-if-vl20)# ip ospf retransmit-interval 10
switch-C(config-if-vl20)# ip ospf transmit-delay 2
switch-C(config-if-vl20)# ip ospf priority 4
switch-C(config-if-vl20)# interface vlan 21
switch-C(config-if-vl21)# ip ospf authentication-key ijklmnop
switch-C(config-if-vl21)# ip ospf cost 20
switch-C(config-if-vl21)# ip ospf dead-interval 80
```

3. Attach the network segments to the areas.

a. Router A interfaces:

```
switch-A(config-if-vl20)# router ospf 1
switch-A(config-router-ospf)# router-id 10.24.1.1
switch-A(config-router-ospf)# network 10.119.254.0/24 area 0
switch-A(config-router-ospf)# network 10.119.251.0/24 area 0
switch-A(config-router-ospf)# area 0 range 10.119.251.0 0.0.7.255
```

b. Router B interfaces:

```
switch-B(config-if-vl16)# router ospf 1
switch-B(config-router-ospf)# router-id 10.24.1.2
switch-B(config-router-ospf)# area 10.42.110.0 stub
switch-B(config-router-ospf)# network 10.42.110.0/24 area 10.42.110.0
switch-B(config-router-ospf)# network 10.119.254.0/24 area 0
```

c. Router C interfaces:

```
switch-C(config-if-vl21)# router ospf 1
switch-C(config-router-ospf)# router-id 10.24.1.3
switch-C(config-router-ospf)# area 10.56.0.0 stub 0
switch-C(config-router-ospf)# network 10.119.251.0/24 area 0
switch-C(config-router-ospf)# network 10.56.0.0/24 area 36.56.0.0
```

15.2.4.3 OSPFv2 Configuration Example 3

The AS in Example 3 contains two areas that connect through one ABR.

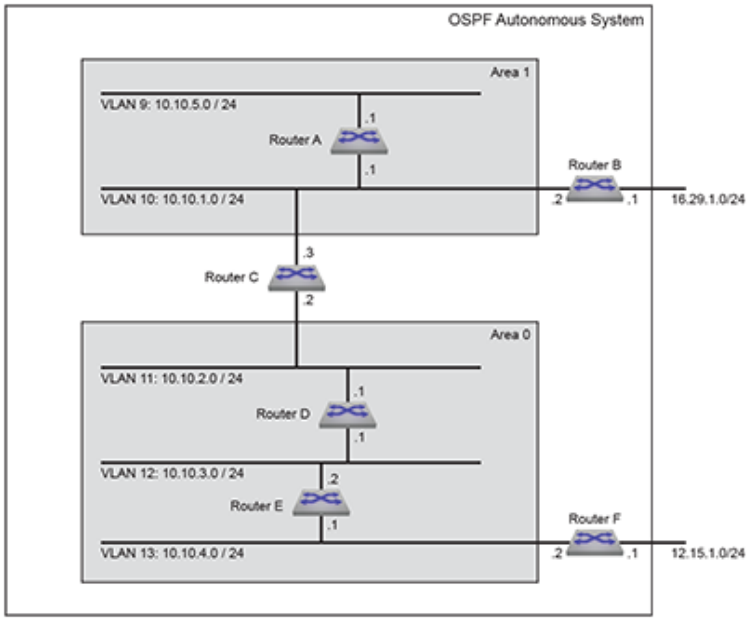
- ***Area 0***: Backbone area contains two internal routers that connect three subnets, one ASBR, and one ABR that connects to ***Area 1***.
- ***Area 1***: NSSA contains one internal router, one ASBR, and one ABR that connects to the backbone.

15.2.4.3.1 Example 3 Topology

OSPFv2 Example 3 displays the Example 3 topology. One ABR connects area ***0*** and area ***1***. ***Router C*** is an ABR that connects the areas. ***Router A*** is an internal router that connects two subnets in area ***1***. ***Router D***

and ***Router E*** are internal routers that connect subnets in area ***0***. ***Router B*** and ***Router F*** are ASBRs that connect static routes outside the AS to area ***1*** and area ***0***, respectively.

Figure 15-5: OSPFv2 Example 3



Area 0 ABR Configuration

ABR ***Router C*** connects one area ***0*** subnet to an area ***1*** subnet.

- ***Router C***: The subnet ***10.10.2.0/24*** is accessed through ***VLAN 11***.
- Authentication is not configured on the interfaces.
- All interface OSPFv2 parameters are set to their default values.

Area 0 IR Configuration

Area ***0*** contains two internal routers, each of which connects two of the three subnets in the area.

- ***Router D***: The subnet ***10.10.2.0/24*** is accessed through ***VLAN 11***.
- ***Router D***: The subnet ***10.10.3.0/24*** is accessed through ***VLAN 12***.
- ***Router E***: The subnet ***10.10.3.0/24*** is accessed through ***VLAN 12***.
- ***Router E***: The subnet ***10.10.4.0/24*** is accessed through ***VLAN 13***.
- All interface OSPFv2 parameters are set to their default values.

Area 0 ASBR Configuration

ASBR ***Router F*** connects one area ***0*** subnet to an external subnet.

- ***Router F***: The subnet ***10.10.4.0/24*** is accessed through ***Router F***.
- ***Router F***: The subnet ***12.15.1.0/24*** is accessed through ***VLAN 14***.
- All interface OSPFv2 parameters are set to their default values.

Area 1 ABR Configuration

ABR ***Router C*** connects one area ***0*** subnet to area ***1***.

- ***Router C***: The subnet ***10.10.1.0/24*** is accessed through ***VLAN 10***.
- Authentication is not configured on the interface.
- All interface OSPFv2 parameters are set to their default values.

Area 1 IR Configuration

Area ***1*** contains one internal router that connects two subnets in the area.

- ***Router A***: The subnet ***10.10.1.0/24*** is accessed through ***VLAN 10***.
- ***Router A***: The subnet ***10.10.5.0/24*** is accessed through ***Router A***.
- All interface OSPFv2 parameters are set to their default values.

Area 1 ASBR Configuration

ASBR ***Router B*** connects one area ***1*** subnet to an external subnet.

- ***Router B***: The subnet ***10.10.1.0/24*** is accessed through ***VLAN 10***.
- ***Router B***: The subnet ***16.29.1.0/24*** is accessed through ***VLAN 15***.
- All interface OSPFv2 parameters are set to their default values.

15.2.4.3.2 Example 3 Code

1. Configure the interfaces.
   a. Router A interfaces:

```
switch-A(config)# interface vlan 10
switch-A(config-if-vl10)# ip address 10.10.1.1/24
switch-A(config-if-vl10)# interface vlan 9
switch-A(config-if-vl11)# ip address 10.10.5.1/24
```

   b. Router B interfaces:

```
switch-B(config)# interface vlan 10
switch-B(config-if-vl10)# ip address 10.10.1.2/24
switch-B(config-if-vl10)# interface vlan 15
switch-B(config-if-vl18)# ip address 16.29.1.1/24
```

   c. Router C interfaces:

```
switch-C(config)# interface vlan 10
switch-C(config-if-vl10)# ip address 10.10.1.3/24
switch-C(config-if-vl10)# interface vlan 11
switch-C(config-if-vl11)# ip address 10.10.2.2/24
```

   d. Router D interfaces:

```
switch-D(config)# interface vlan 11
switch-D(config-if-vl11)# ip address 10.10.2.1/24
switch-D(config)# interface vlan 12
switch-D(config-if-vl12)# ip address 10.10.3.1/24
```

   e. Router E interfaces:

```
switch-E(config)# interface vlan 12
switch-E(config-if-vl12)# ip address 10.10.3.2/24
switch-E(config)# interface vlan 13
switch-E(config-if-vl13) #ip address 10.10.4.1/24
```

   f. Router F interfaces:

```
switch-F(config)# interface vlan 13
switch-F(config-if-vl13)# ip address 10.10.4.2/24
switch-F(config)# interface vlan 14
switch-F(config-if-vl14)# ip address 12.15.1.1/24
```

2. Attach the network segments to the areas.

   a. Router A interfaces:

```
switch-A(config-if-vl10)# router ospf 1
switch-A(config-router-ospf)# router-id 170.21.0.1
switch-A(config-router-ospf)# area 1 NSSA
switch-A(config-router-ospf)# network 10.10.1.0/24 area 1
```

   b. Router B interfaces:

```
switch-B(config-if-vl10)# router ospf 1
switch-B(config-router-ospf)# router-id 170.21.0.2
switch-B(config-router-ospf)# area 1 NSSA
switch-B(config-router-ospf)# network 10.10.1.0/24 area 1
```

   c. Router C interfaces:

```
switch-C(config-if-vl11)# router ospf 1
switch-C(config-router-ospf)# router-id 170.21.0.3
switch-C(config-router-ospf)# area 1 NSSA
switch-C(config-router-ospf)# network 10.10.1.0/24 area 1
switch-C(config-router-ospf)# network 10.10.2.0/24 area 0
```

   d. Router D interfaces:

```
switch-D(config-if-vl12)# router ospf 1
switch-D(config-router-ospf)# router-id 170.21.0.4
switch-D(config-router-ospf)# network 10.10.2.0/24 area 0
switch-D(config-router-ospf)# network 10.10.3.0/24 area 0
```

   e. Router E interfaces:

```
switch-E(config-if-vl13)# router ospf 1
switch-E(config-router-ospf)# router-id 170.21.0.5
switch-E(config-router-ospf)# network 10.10.3.0/24 area 0
switch-E(config-router-ospf)# network 10.10.4.0/24 area 0
```

   f. Router F interfaces:

```
switch-F(config-if-vl14)# router ospf 1
switch-F(config-router-ospf)# router-id 170.21.0.6
switch-F(config-router-ospf)# network 10.10.4.0/24 area 0
switch-F(config-router-ospf)# redistribute static
```

15.2.5 OSPFv2 Commands

Global Configuration Mode

- ip ospf router-id output-format hostnames
- line system
- router ospf

Interface Configuration Mode

- interface Tunnel
- ip ospf area
- ip ospf authentication
- ip ospf authentication-key

- ip ospf cost
- ip ospf dead-interval
- ip ospf disabled
- ip ospf hello-interval
- ip ospf message-digest-key
- ip ospf network point-to-point
- ip ospf priority
- ip ospf retransmit-interval
- ip ospf transmit-delay
- tunnel routes

Router-OSPFv2 Configuration Mode

- adjacency exchange-start threshold (OSPFv2)
- area default-cost (OSPFv2)
- area filter (OSPFv2)
- area nssa (OSPFv2)
- area nssa default-information-originate (OSPFv2)
- area nssa no-summary (OSPFv2)
- area not-so-stubby lsa type-7 convert type-5 (OSPFv2)
- area range (OSPFv2)
- area stub (OSPFv2)
- auto-cost reference-bandwidth (OSPFv2)
- compatible (OSPFv2)
- default-information originate (OSPFv2)
- distance ospf (OSPFv2)
- dn-bit-ignore (OSPFv2)
- log-adjacency-changes (OSPFv2)
- max-lsa (OSPFv2)
- max-metric router-lsa (OSPFv2)
- maximum-paths (OSPFv2)
- network area (OSPFv2)
- no area (OSPFv2)
- passive-interface default (OSPFv2)
- passive-interface (OSPFv2)
- point-to-point routes (OSPFv2)
- redistribute (OSPFv2)
- redistribute ospf instance
- router-id (OSPFv2)
- shutdown (OSPFv2)
- summary-address
- timers lsa rx min interval (OSPFv2)
- timers lsa tx delay initial (OSPFv2)
- timers spf delay initial (OSPFv2)

TCAM Profile Configuration Mode

- packet
- system profile

Display and Clear Commands

- clear ip ospf neighbor
- show hardware tcam profile
- show ip ospf
- show ip ospf border-routers
- show ip ospf database database-summary
- show ip ospf database <link state list>
- show ip ospf database <link-state details>
- show ip ospf interface
- show ip ospf interface brief
- show ip ospf lsa-log
- show ip ospf neighbor
- show ip ospf neighbor adjacency-changes
- show ip ospf neighbor state
- show ip ospf neighbor summary
- show ip ospf request queue
- show ip ospf retransmission queue
- show ip ospf spf-log
- show line system dom thresholds
- show line system status

15.2.5.1 auto-cost reference-bandwidth (OSPFv2)

The **`auto-cost reference-bandwidth`** command is a factor in the formula that calculates the default OSPFv2 cost for Ethernet interfaces.

OSPFv2-cost = (auto-cost value * ***1*** Mbps) / interface bandwidth.

The switch uses a minimum OSPFv2-cost of ***1***. The switch rounds down all non-integer results.

On a 10G Ethernet interface:

- if auto-cost = ***100***, then OSPFv2-cost = ***100*** Mbps / 10 Gbps = ***0.01***, and the default cost is set to ***1***.
- if auto-cost = ***59000***, then OSPFv2-cost = ***59000*** Mbps / 10 Gbps = ***5.9***, and the default cost is set to ***5***.

The **`no auto-cost reference-bandwidth`** and **`default auto-cost reference-bandwidth`** command removes the **`auto-cost reference-bandwidth`** command from ***running-config***. When this parameter is not set, the default cost for Ethernet interfaces is the default **`ip ospf cost`** value of 10.

Command Mode

Router-OSPF Configuration

Command Syntax

`auto-cost reference-bandwidth` ***rate***

`no auto-cost reference-bandwidth` ***rate***

`default auto-cost reference-bandwidth` ***rate***

Parameter

rate Values range from ***1 to 4294967***. Default is ***100***.

Example

To configure a default cost of ***20*** on 10G Ethernet interfaces:

1. Calculate the required auto-cost value:
   auto-cost = (OSPFv2-cost*interface bandwidth)/1 Mbps = (***20*******10000*** Mbps) / ***1*** Mbps = ***200000***

2. Configure this value as the auto-cost reference-bandwidth.

```
switch(config)# router ospf 6
switch(config-router-ospf)# auto-cost reference-bandwidth 200000
switch(config-router-ospf)#
```

15.2.5.2 adjacency exchange-start threshold (OSPFv2)

The **adjacency exchange-start threshold** command sets the exchange-start options for an OSPF instance.

The **no adjacency exchange-start threshold** and **default adjacency exchange-start threshold** command resets the default by removing the corresponding a **adjacency exchange-start threshold** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

default adjacency exchange-start threshold

adjacency exchange-start threshold ***peers***

no adjacency exchange-start threshold

Parameter

peers Value ranges from ***1- 4294967295***. Default value is ***10***.

Example

This command sets the adjacency exchange start threshold to ***20045623***.

```
switch(config)# router ospf 6
switch(config-router-ospf)# adjacency exchange-start threshold 20045623
switch(config-router-ospf)#
```

15.2.5.3 area default-cost (OSPFv2)

The **area default-cost** command specifies the cost for the default summary routes sent into a specified area. The default-cost is set to ***10***.

The **no area default-cost** and **default area default-cost** command resets the default-cost value of the specified area to ***10*** by removing the corresponding **area default-cost** command from **running-config**. The no area (OSPFv2) command removes all area commands for the specified area from ***running-config***, including the **area default-cost** command.

Command Mode

Router-OSPF Configuration

Command Syntax

area ***area_id*** **default-cost** ***def_cost***

no area ***area_id*** **default-cost** ***def_cost***

default area ***area_id*** **default-cost** ***def_cost***

Parameters

- ***area_id*** Area number: ***0 to 4294967295*** or ***0.0.0.0*** to **255.255.255.255** ***running-config*** stores value in dotted decimal notation.
- ***def_cost*** Value ranges from ***1*** to ***65535***. Default value is ***10***.

Example

This command configures a cost of ***15*** for default summary routes that an ABR sends into area ***23***.

```
switch(config)# router ospf 6
switch(config-router-ospf)# area 23 default-cost 15
switch(config-router-ospf)#
```

15.2.5.4 area filter (OSPFv2)

The **area filter** command prevents an area from receiving Type 3 Summary LSAs from a specified subnet.

The **no area filter** and **default area filter** commands remove the specified **area filter** command from ***running-config***. The **no area** command (see no area (OSPFv2) removes all area commands for the specified area from ***running-config***, including **area filter** commands.

Command Mode

Router-OSPF Configuration

Command Syntax

area ***area_id*** **filter** ***net_addr***

no area ***area_id*** **filter** ***net_addr***

default area ***area_id*** **filter** ***net_addr***

Parameters

- ***area_id*** Area number. ***0*** to ***4294967295*** or ***0.0.0.0*** to ***255.255.255.255***. ***running-config*** stores value in dotted decimal notation.
- ***net_addr*** Network IP address. Entry formats include address-prefix (CIDR) and address-mask. ***running-config*** stores value in CIDR notation.

Example

This command prevents the switch from entering Type 3 LSAs originating from the ***10.1.1.0/24*** subnet into its area ***2*** LSDB.

```
switch(config)# router ospf 6
switch(config-router-ospf)# area 2 filter 10.1.1.0/24
switch(config-router-ospf)#
```

15.2.5.5 area not-so-stubby lsa type-7 convert type-5 (OSPFv2)

The **area not-so-stubby lsa type-7 convert type-5** command configures the switch to always translate Type-7 Link-State Advertisement (LSAs) to Type-5 LSAs.

The **no area not-so-stubby lsa type-7 convert type-5** and **no area not-so-stubby lsa type-7 convert type-5** commands allow LSAs to be translated dynamically by removing the **no area not-so-stubby lsa type-7 convert type-5** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

area ***area_id*** **not-so-stubby lsa type-7 convert type-5**

no area ***area_id*** **not-so-stubby lsa type-7 convert type-5**

default area ***area_id*** **not-so-stubby lsa type-7 convert type-5**

Parameters

area_id Area number.

- Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
- Area ***0*** (or ***0.0.0.0***) is not configurable; it is always **normal**.
- ***running-config*** stores value in dotted decimal notation.

Example

This command configures the switch to always translate Type-7 Link-State Advertisement (LSAs) to Type-5 LSAs.

```
switch(config-router-ospf)# area 3 not-so-stubby lsa type-7 convert type-5
switch(config-router-ospf)#
```

15.2.5.6 area nssa (OSPFv2)

The **area nssa** command configures an OSPFv2 area as a Not-So-Stubby Area (NSSA). All routers in an AS must specify the same area type for identically numbered areas.

NSSA ASBRs advertise external LSAs that are part of the area, but do not advertise external LSAs from other areas.

Areas are **normal** by default; area type configuration is required only for stub NSSA areas. Area 0 is always a normal area and cannot be configured through this command.

The **no area nssa** command configures the specified area as a normal area by removing the specified **area nssa** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

area ***area_id*** **nssa** [**TYPE**]

no area ***area_id*** **nssa** [**TYPE**]

default area ***area_id*** **nssa** [**TYPE**]

Parameters

- ***area_id*** All parameters except ***area_id*** can be placed in any order.
  - Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
  - Area ***0*** (or ***0.0.0.0***) is not configurable; it is always **normal**.
  - ***running-config*** stores value in dotted decimal notation.
- **TYPE** Area type. Values include:
  - ***no parameter***
  - **nssa-only**

Example

This command configures area ***3*** as a NSSA area.

```
switch(config-router-ospf)# area 3 nssa nssa-only
switch(config-router-ospf)#
```

15.2.5.7 area nssa default-information-originate (OSPFv2)

The **`default area nssa default-information-originate`** command sets default route origination for the Not-So-Stubby Area (NSSA), allowing the redistribute policy to advertise a default route if one is present. The resulting OSPF behavior depends on the presence of an installed static default route and on whether static routes are redistributed in OSPF (using the redistribute (OSPFv2) command). The **`no area nssa default-information-originate`** command disables advertisement of the default route for the NSSA regardless of the redistribute policy. See Advertisement of Default Route for details.

Areas are **normal** by default; area type configuration is required only for stub and NSSA areas. Area ***0*** is always a normal area and cannot be configured through this command.

Default route origination is configured differently for different area types and supports three area types:

- Normal areas: advertisement of the default route is configured for all normal areas using the default-information originate (OSPFv2) command.
- Stub areas: the default route is automatically advertised in stub areas and cannot be configured.
- Not So Stubby Areas (NSSAs): advertisement of the default route is configured per area using the area nssa default-information-originate (OSPFv2) or area nssa no-summary (OSPFv2) command.

Table 70: Advertisement of Default Route

Static Default Route Installed	Redistribute Static	Command Form	Advertise in ABR	Advertise in ASBR
no	no	**default** or **no**	no	no
no	no	standard	yes	no
no	yes	**default**	yes	yes
no	yes	**no**	no	no
no	yes	standard	yes	no
yes	no	**default** or **no**	no	no
yes	no	standard	yes	yes
yes	yes	**default**	yes	yes
yes	yes	**no**	no	no
yes	yes	standard	yes	yes

Command Mode

Router-OSPF Configuration

Command Syntax

`area` ***`area_id`*** **`nssa default-information-originate`** [**VALUE**][**TYPE**][**EXCL**]

`no area` ***`area_id`*** **`nssa default-information-originate`**

`default area` ***`area_id`*** **`nssa default-information-originate`**

Parameters

- ***area_id*** All parameters except ***area_id*** can be placed in any order.
  - Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
  - Area ***0*** (or ***0.0.0.0***) is not configurable; it is always **normal**.
  - ***running-config*** stores value in dotted decimal notation.
- **VALUE** Values include:

- ***no parameter*** Default value of ***1***.
- **metric *1-65535***.

- **TYPE** Values include:
  - ***no parameter***.
  - **metric-type *1-2***.
- **EXCL** Values include:
  - ***no parameter***.
  - **nssa-only**.

Example

This command configures area ***3*** as an NSSA and generates a type 7 default LSA within the NSSA.

```
switch(config-router-ospf)# area 3 nssa default-information-originate nssa-only
switch(config-router-ospf)#
```

15.2.5.8 area nssa no-summary (OSPFv2)

The **`area nssa no-summary`** command configures the switch stop importing type-3 summary LSAs into the not-so-stubby area and sets the default summary route into the Not-So-Stubby Area (NSSA) in order to reach the inter-area prefixes.

The **`no area nssa no-summary`** and **`default area nssa no-summary`** commands allow type-3 summary LSAs into the NSSA area.

The **`no area nssa`** and **`default area nssa`** commands configure the specified area as a normal area.

Command Mode

Router-OSPF Configuration

Command Syntax

`area` ***area_id*** **`nssa no-summary`**

`no area` ***area_id*** **`nssa no-summary`**

`default area` ***area_id*** **`nssa no-summary`**

Parameters

area_id Area number.

- Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
- Area ***0*** (or ***0.0.0.0***) is not configurable; it is always **normal**.
- ***running-config*** stores value in dotted decimal notation.

Examples

- This command directs the device not to import type-3 summary LSAs into the NSSA area

```
switch(config)# router ospf 6
switch(config-router-ospf)# area 1.1.1.1 nssa no-summary
switch(config-router-ospf)#
```

- This command directs the device to import type-3 summary LSAs into the NSSA area.

```
switch(config)# router ospf 6
switch(config-router-ospf)# no area 1.1.1.1 nssa no-summary
switch(config-router-ospf)#
```

15.2.5.9 area range (OSPFv2)

The **area range** command configures OSPF Area Border Routers (ABRs) to consolidate or summarize routes, to set the cost setting routes, and to suppress summary route advertisements.

The no area (OSPFv2) command removes all area commands for the specified area from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

area ***area_id*** **range** ***net_addr*** [**ADVERTISE_SETTING**][**COST_SETTING**]

no area ***area_id*** **range** ***net_addr*** [**ADVERTISE_SETTING**][**COST_SETTING**]

default area ***area_id*** **range** ***net_addr*** [**ADVERTISE_SETTING**][**COST_SETTING**]

Parameters

- ***area_id*** Area number. ***0*** to ***4294967295*** or ***0.0.0.0*** to ***255.255.255.255 running-config*** stores value in dotted decimal notation.
- ***net_addr***.
- **ADVERTISE_SETTING** Values include:
  - ***no parameter***
  - **advertise**
  - **not-advertise**
- Values include:
  - ***no parameter***
  - **cost** ***range_cost*** Value ranges from ***1 to 65535***.

Examples

- The **network area** command assigns two subnets to an area. The **area range** command summarizes the addresses, which the ABR advertises in a single LSA.

```
switch(config)# router ospf 6
switch(config-router-ospf)# network 10.1.25.80 0.0.0.240 area 5
switch(config-router-ospf)# network 10.1.25.112 0.0.0.240 area 5
switch(config-router-ospf)# area 5 range 10.1.25.64 0.0.0.192
switch(config-router-ospf)#
```

- The **network area** command assigns a subnet to an area, followed by an **area range** command that suppresses the advertisement of that subnet.

```
switch(config-router-ospf)# network 10.12.31.0/24 area 5
switch(config-router-ospf)# area 5 range 10.12.31.0/24 not-advertise
switch(config-router-ospf)#
```

15.2.5.10 area stub (OSPFv2)

The **area stub** command sets the area type of an OSPF area to **stub**. All devices in an Area Stub (AS) must specify the same area type for identically numbered areas.

The **no area stub** command remove the specified stub area from the OSPFv2 instance by deleting all **area stub** commands from ***running-config*** for the specified area.

The **no area stub** command configures the specified area as a normal area.

Command Mode

Router-OSPF Configuration

Command Syntax

area ***area_id*** **stub** [summarize]

no area ***area_id*** **stub** [summarize]

default area ***area_id*** **stub** [summarize]

Parameters

- ***area_id*** Area number.
  - Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
  - Area ***0*** **(or** ***0.0.0.0***) is not configurable; it is always normal.
  - ***running-config*** stores value in dotted decimal notation.
- **summarize** Area type. Values include:
  - ***no parameter***
  - **no-summary**

Examples

- These commands configure area ***45*** as a stub area.

```
switch(config)# router ospf 3
switch(config-router-ospf)# area 45 stub
switch(config-router-ospf)#
```

- This command configures area ***10.92.148.17*** as a stub area.

```
switch(config-router-ospf)# area 10.92.148.17 stub
switch(config-router-ospf)#
```

15.2.5.11 clear ip ospf neighbor

The **clear ip ospf** command clears the neighbors statistics per interface.

Command Mode

Privileged EXEC

Command Syntax

clear ip ospf [**PROCESS_ID**] **neighbor** [**LOCATION**][**VRF_INSTANCE**]

Parameters

- **PROCESS_ID** OSPFv2 process ID. Values include:
  - ***no parameter***
  - ***1 to 65535***.
- **LOCATIONIP** Address or interface peer group name. Values include:
  - ***** clears all OSPF IPv4 neighbors.
  - ***ipv4_addr***
  - **ethernet** ***e_num***
  - **loopback** ***l_num***
  - **port-channel** ***p_num***
  - ***vlan v_num***
- **VRF_INSTANCE** Specifies the VRF instance.
  - **vrf** ***vrf_name*** configures the ***vrf_name*** instance.

Examples

- This command resets all OSPF neighbor statistics.

```
switch# clear ip ospf neighbor *
switch#
```

- This command resets the OSPF neighbor statistics for the specified ***ethernet 3*** interface.

```
switch# clear ip ospf neighbor ethernet 3
switch#
```

15.2.5.12 compatible (OSPFv2)

The **`compatible`** command allows the selective disabling of compatibility with ***RFC 2328.***

The **`no compatible`** and **`default compatible`** commands reverts OSPF to ***RFC 2328*** compatible and removes the compatible statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

`compatible rfc1583`

`no compatible rfc1583`

`default compatible rfc1583`

Examples

- This command sets the OSPF compatibility list with ***RFC 1583***.

```
switch(config)# router ospf 6
switch(config-router-ospf)# compatible rfc1583
switch(config-router-ospf)#
```

- This command disables ***RFC 1583*** compatibility.

```
switch(config)# router ospf 6
switch(config-router-ospf)# no compatible rfc1583
switch(config-router-ospf)#
```

15.2.5.13 default-information originate (OSPFv2)

The **`default-information originate`** command enables default route origination for normal areas. The user user may configure the metric value and metric type used in LSAs. The always option will cause the ASBR to create and advertise a default route whether or not one is configured.

The **`no default-information originate`** command prevents the advertisement of the default route. The **`default default-information originate`** command enables default route origination with default values (metric type 2, metric=1).

Command Mode

Router-OSPF Configuration

Command Syntax

`default-information originate` [**FORCE**][**VALUE**][**TYPE**][**MAP**]

`no default-information originate`

`default default-information originate`

Parameters

All parameters can be placed in any order.

- **FORCE** Advertisement forcing option. Values include:
  - ***no parameter***
  - **always**
- **VALUE** Values include:
  - ***no parameter***
  - **metric *1-65535***
- **TYPE** Values include:
  - ***no parameter***
  - **metric-type *1-2***
- **MAP** Sets attributes in the LSA based on a route map. Values include:
  - ***no parameter***
  - **route-map *map_name***.

Examples

- These commands always advertise the OSPFv2 default route regardless of whether the switch has a default route configured.

```
switch(config)# router ospf 1
switch((config-router-ospf)# default-information originate always
switch(config-router-ospf)# show active
router ospf 1
   default-information originate always
```

- These commands advertise a default route with a metric of ***100*** and an external metric type of ***1*** if a default route is configured.

```
switch(config)# router ospf 1
switch((config-router-ospf)# default-information originate metric 100
 metric-type 1
```

15.2.5.14 distance ospf (OSPFv2)

The **`distance ospf`** command specifies the administrative distance for intra-area, inter-area, or external OSPF routes. The command must be issued separately for each route type being configured. The default administrative distance for all routes is ***110***.

The **`no distance ospf`** and **`default distance ospf`** commands remove the corresponding **`distance ospf`** command from ***running-config***, returning the OSPFv2 administrative distance setting for the specified route type to the default value of ***110***.



Note: OSPF links will flap if the administrative distance value is adjusted while OSPF is running, whether it is adjusted by entering the **`distance ospf`** command directly through the CLI or by applying a configuration file that contains the command.

Command Mode

Router-OSPF Configuration

Command Syntax

`distance ospf` [external | inter-area | intra-area]

`no distance ospf` [external | inter-area | intra-area]

`default distance ospf` [external | inter-area | intra-area]

Parameters

- **external** Sets administrative distance for external routes.
- **inter-area** Sets administrative distance for inter-area routes.
- **intra-area** sSets administrative distance for intra-area routes.
- ***distance*** Values range from ***1 to 255***. Default value is ***110*** for all types.

Example

This command configures an administrative distance of ***85*** for all OSPFv2 intra-area routes on the switch. If issued while OSPF is running, this command will cause OSPF links to flap.

```
switch(config)# router ospf 6
switch(config-router-ospf)# distance ospf intra-area 85
switch(config-router-ospf)#
```

15.2.5.15 distribute-list in

A distribute list uses a route map or prefix list to filter specific routes from incoming OSPF LSAs. Filtering occurs after SPF calculation. The filtered routes are not installed on the switch, but are still included in LSAs sent by the switch. The **`distribute-list in`** command creates a distribute list in the configuration mode OSPF instance.

If a prefix list is used, destination prefixes that do not match the prefix list will not be installed. If a route map is used, routes may be filtered based on address, next hop, or metric. OSPF external routes may also be filtered by metric type or tag.

The **`no distribute-list in`** and **`default distribute-list in`** commands remove the **`distribute-list in`** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

`distribute-list` {prefix-list | route-map} *list_name* **`in`**

`no distribute-list` {prefix-list | route-map}

`default distribute-list` {prefix-list | route-map}

Parameters

- **prefix-list** Specifies a prefix-list as the filter.
- **route-map** Specifies a route-map as the filter.
- ***list_name*** The name of the prefix-list or route-map used to filter routes from incoming LSAs.

Related Commands

- area filter (OSPFv2)
- redistribute (OSPFv2)

Example

These commands configure a prefix list named ***dist_list1*** in OSPF instance ***5*** to filter certain routes from incoming OSPF LSAs.

```
switch(config)# router ospf 5
switch(config-router-ospf)# distribute-list prefix-list dist_list1 in
switch(config-router-ospf)#
```

15.2.5.16 dn-bit-ignore (OSPFv2)

The **dn-bit-ignore** command results in the DN bit in Type 3 Summary LSAs to be ignored during the Shortest Path First (SPF) calculations.

The **no dn-bit-ignore** and **default dn-bit-ignore** commands result in the DN bit in Type 3 Summary LSAs to not be ignored during SPF calculations.

Command Mode

Router-OSPF Configuration

Command Syntax

dn-bit-ignore

no dn-bit-ignore

default dn-bit-ignore

Examples

- This command ignores the DN bit.

```
switch(config)# router ospf 6
switch(config-router-ospf)# dn-bit-ignore
switch(config-router-ospf)#
```

- This command causes the DN bit not to be ignored.

```
switch(config)# router ospf 6
switch(config-router-ospf)# no dn-bit-ignore
switch(config-router-ospf)#
```

15.2.5.17 interface Tunnel

OSPF packets by default are sent with Time to Live (TTL) value ***1***. This may not work in tunnel scenarios where the peer tunnel end point could be more than one hop away. It is recommended to explicitly configure TTL on the tunnel interface. TTL configuration is allowed only if **path-mtu-discovery** is configured.

Command Mode

Configuration mode

Command Syntax

interface Tunnel ***Tunnel No***

Parameters

Tunnel No Tunnel number.

Example

```
(config)# interface Tunnel 5
(config-if-Tu0)# tunnel path-mtu-discovery
(config-if-Tu0)# tunnel ttl 5
```

15.2.5.18 ip ospf area

The **ip ospf area** command enables OSPFv2 on an interface and associates the area to the interface.

The **no ip ospf area** and **default ip ospf area** commands disable OSPFv2 on the configuration mode interface and remove the configured area from the system.

Note: The per interface configuration has precedence over the OSPF Configuration mode. In other words, the area configured by the ip ospf area command has precedence over the area configured by the network area command.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip ospf area` ***area_id***

`no ip ospf area` ***area_id***

`default ip ospf area` ***area_id***

Parameters

area_id The area ID. The valid values are ***0*** to ***4294967295*** or a decimal range between ***0.0.0.0*** and ***255.255.255.255***.

Example

These commands enable OSPFv2 on the ***et2*** interface and associates area identifier ***1.1.1.1*** to the interface.

```
switch(config)# Interface ethernet 2
switch(config-if-Et2)# ip address 1.0.0.1/24
switch(config-if-Et2)# ip ospf area 1.1.1.1
router ospf 1
```

15.2.5.19 ip ospf authentication

The **`ip ospf authentication`** command enables OSPFv2 authentication for the configuration mode interface..

The **`no ip ospf authentication`** and **`default ip ospf authentication`** commands disable OSPFv2 authentication on the configuration mode interface by removing the corresponding **`ip ospf authentication`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip ospf authentication` [**METHOD**]

`no ip ospf authentication`

`default ip ospf authentication`

Parameters

METHOD OSPFv2 authentication method. Options include:

- ***no parameter***
- **message-digest**

Examples

- This command enables simple authentication on ***vlan 12***.

```
switch(config)# interface vlan 12
switch(config-if-vl12)# ip ospf authentication
switch(config-if-vl12)#
```

- This command enables message-digest authentication on ***vlan 12***.

```
switch(config-if-vl12)# ip ospf authentication message-digest
switch(config-if-vl12)#
```

15.2.5.20 ip ospf authentication-key

The **`ip ospf authentication-key`** command configures the OSPFv2 authentication password for the configuration mode interface.

The **`no ip ospf authentication-key`** and **`default ip ospf authentication-key`** commands removes the OSPFv2 authentication password from the configuration mode interface by removing the corresponding **`ip ospf authentication-key`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip ospf authentication-key` [**ENCRYPT_TYPE**] ***key_text***

`no ip ospf authentication-key`

`default ip ospf authentication-key`

Parameters

- **ENCRYPT_TYPE** Encryption level of the ***key_text*** parameter. Values include:
  - ***no parameter*** the ***key_text*** is in clear text.
  - ***0 key_text*** is in clear text. Equivalent to ***no parameter***.
  - ***7 key_text*** is MD5 encrypted.
- ***key_text*** the authentication-key password.

Example

This command specifies a password in clear text.

```
switch(config)# interface vlan 12
switch(config-if-Vl12)# ip ospf authentication-key 0 code123
switch(config-if-Vl12)# show active
interface Vlan12
   ip ospf authentication-key 7 baYllFzVbcx4yHq1IhmMdw==
switch(config-if-Vl12)#
```

The ***running-config*** stores the password as an encrypted string.

15.2.5.21 ip ospf cost

The **`ip ospf cost`** command configures the OSPFv2 cost for the configuration mode interface. The default cost depends on the interface type:

- Ethernet: determined by the **`auto-cost reference-bandwidth (OSPFv2)`** command.

- Port channel: ***10***.
- VLAN: ***10***.

The **no ip ospf cost** and **default ip ospf cost** commands restore the default OSPFv2 cost for the configuration mode interface by removing the corresponding **ip ospf cost** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip ospf cost ***interface_cost***

no ip ospf cost

default ip ospf cost

Parameters

interface_cost Value ranges from ***1 to 65535***; default is ***10***.

Example

This command configures a cost of ***15*** for ***vlan 2***.

```
switch(config)# interface vlan 2
switch(config-if-Vl2)# ip ospf cost 15
switch(config-if-Vl2)#
```

15.2.5.22 ip ospf dead-interval

The **ip ospf dead-interval** command configures the dead interval for the configuration mode interface.

The **no ip ospf dead-interval** and **default ip ospf dead-interval** commands restore the default dead interval of ***40*** seconds on the configuration mode interface by removing the corresponding **ip ospf dead-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip ospf dead-interval ***time***

no ip ospf dead-interval

default ip ospf dead-interval

Parameters

time Value ranges from ***1 to 8192***; default is ***40***.

Example

This command configures a dead interval of ***120*** seconds for ***vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip ospf dead-interval 120
switch(config-if-Vl4)#
```

15.2.5.23 ip ospf disabled

The **ip ospf disabled** command disables OSPFv2 on the configuration mode interface without disrupting the OSPFv2 configuration. When OSPFv2 is enabled on the switch, the it is also enabled by default on all interfaces.

The OSPFv2 instance is disabled on the entire switch with the **shutdown (OSPFv2)** command.

The **no ip ospf disabled** and **default ip ospf disabled** commands enable OSPFv2 on the configuration mode interface by removing the corresponding **ip ospf disabled** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip ospf disabled

no ip ospf disabled

default ip ospf disabled

Examples

- This command shuts down OSPFv2 activity on ***vlan 5***.

```
switch(config)# interface vlan 5
switch(config-if-Vl5)# ip ospf disabled
switch(config-if-Vl5)#
```

- This command resumes OSPFv2 activity on ***vlan 5***.

```
switch(config-if-Vl5)# no ip ospf disabled
switch(config-if-Vl5)#
```

15.2.5.24 ip ospf hello-interval

The **ip ospf hello-interval** command configures the OSPFv2 hello interval for the configuration mode interface.

The same hello interval should be specified for Each OSPFv2 neighbor, and should not be longer than any neighbors dead interval.

The **no ip ospf hello-interval** and **default ip ospf hello-interval** commands restore the default hello interval of ***10*** seconds on the configuration mode interface by removing the **ip ospf hello-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip ospf hello-interval` ***time***

`no ip ospf hello-interval`

`default ip ospf hello-interval`

Parameter

time Hello interval (seconds). Values range from ***1 to 8192***; default is ***10***.

Example

This command configures a hello interval of ***30*** seconds for ***vlan 2***.

```
switch(config)# interface vlan 2
switch(config-if-Vl2)# ip ospf hello-interval 30
switch(config-if-Vl2)#
```

15.2.5.25 ip ospf message-digest-key

The **`ip ospf message-digest-key`** command configures a message digest authentication key for the configuration mode interface.

The **`no ip ospf message-digest-key`** and **`default ip ospf message-digest-key`** commands remove the message digest authentication key for the specified key ID on the configuration mode interface by deleting the corresponding **`ip ospf message-digest-key`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip ospf message-digest-key` ***key_id*** md5 **ENCRYPT_TYPE** ***key_text***

`no ip ospf message-digest-key` ***key_id***

`default ip ospf message-digest-key` ***key_id***

Parameters

- ***key_id*** Key ID number. Value ranges from ***1 to 255***.
- **ENCRYPT_TYPE** Encryption level of the ***key_text*** parameters. Values include:
  - ***no parameter***
  - ***0 key_text***
  - ***7 key_text***
- ***key_text*** message key (password).

Example

This command configures ***code123*** as the MD5 key with a corresponding key ID of ***23***.

```
switch(config)# interface vlan 12
switch(config-if-vl12)# ip ospf message-digest-key 23 md5 0 code123
switch(config-if-vl12)#
```

The ***running-config*** stores the password as an encrypted string.

15.2.5.26 ip ospf network point-to-point

The **ip ospf network point-to-point** command sets the configuration mode interface as a point-to-point link. By default, interfaces are configured as broadcast links.

The **no ip ospf network** and **default ip ospf network** commands set the configuration mode interface as a broadcast link by removing the corresponding **ip ospf network**command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip ospf network point-to-point

no ip ospf network

default ip ospf network

Examples

- These commands configure ethernet interface ***10*** as a point-to-point link.

```
switch(config)# interface ethernet 10
switch(config-if-Et10)# ip ospf network point-to-point
switch(config-if-Et10)#
```

- This command restores ethernet interface ***10*** as a broadcast link.

```
switch(config-if-Et10)# no ip ospf network
switch(config-if-Et10)#
```

15.2.5.27 ip ospf retransmit-interval

The **ip ospf retransmit-interval** command configures the link state advertisement retransmission interval for the interface.

The **no ip ospf retransmit-interval** and **default ip ospf retransmit-interval** commands restore the default retransmission interval of ***5*** seconds on the configuration mode interface by removing the corresponding **ip ospf retransmit-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip ospf retransmit-interval ***period***

no ip ospf retransmit-interval

default ip ospf retransmit-interval

Parameters

period Retransmission interval (seconds). Value ranges from ***1 to 8192***; default is ***5***.

Example

This command configures a retransmission interval of ***15*** seconds for ***vlan 3***.

```
switch(config)# interface vlan 3
switch(config-if-Vl3)# ip ospf retransmit-interval 15
switch(config-if-Vl3)#
```

15.2.5.28 ip ospf router-id output-format hostnames

The **ip ospf router-id output-format hostnames** command causes the switch to display DNS names in place of numeric OSPFv2 router IDs in all OSPFv2 show commands, including:

- **show ip ospf**
- **show ip ospf border-routers**
- **show ip ospf database <link state list>**
- **show ip ospf database database-summary**
- **show ip ospf database <link-state details>**
- **show ip ospf interface**
- **show ip ospf neighbor**
- **show ip ospf request queue**
- **show ip ospf retransmission queue**

The **no ip ospf router-id output-format hostnames** and **default ip ospf router-id output-format hostnames** commands remove the **ip ospf router-id output-format hostnames** command from ***running-config***, restoring the default behavior of displaying OSPFv2 router IDs by their numeric value.

Command Mode

Global Configuration

Command Syntax

ip ospf router-id output-format hostnames

no ip ospf router-id output-format hostnames

default ip ospf router-id output-format hostnames

Example

This command programs the switch to display OSPFv2 router IDs by the corresponding DNS name in subsequent show commands.

```
switch(config)# ip ospf router-id output-format hostnames
switch(config)#
```

15.2.5.29 ip ospf transmit-delay

The **ip ospf transmit-delay** command configures the transmission delay for OSPFv2 packets over the configuration mode interface.

The **no ip ospf transmit-delay** and **default ip ospf transmit-delay** commands restore the default transmission delay (***1*** second) on the configuration mode interface by removing the corresponding **ip ospf transmit-delay** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip ospf transmit-delay` ***trans***

`no ip ospf transmit-delay`

`default ip ospf transmit-delay`

Parameters

trans LSA transmission delay (seconds). Value ranges from ***1 to 8192***; default is ***1***.

Example

This command configures a transmission delay of ***5*** seconds for ***vlan 6***.

```
switch(config)# interface vlan 6
switch(config-if-Vl6)# ip ospf transmit-delay 5
switch(config-if-Vl6)#
```

15.2.5.30 ip ospf priority

The **`ip ospf priority`** command configures OSPFv2 router priority for the configuration mode interface..

The **`no ip ospf priority`** and **`default ip ospf priority`** commands restore the default priority (***1***) on the configuration mode interface by removing the corresponding **`ip ospf priority`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration Interface-Port-Channel Configuration Interface-VLAN Configuration

Command Syntax

`ip ospf priority` ***priority_level***

`no ip ospf priority`

`default ip ospf priority`

Parameter

priority_level priority level. Value ranges from ***0 to 255***. Default value is ***1***.

Examples

- This command configures a router priority of ***15*** for ***vlan 8***.

```
switch(config)# interface vlan 8
switch(config-if-Vl8)# ip ospf priority 15
switch(config-if-Vl8)#
```

- This command restores the router priority of ***1*** for ***vlan 7***.

```
switch(config)# interface vlan 7
switch(config-if-Vl7)# no ip ospf priority
switch(config-if-Vl7)#
```

15.2.5.31 line system

The **`line system`** command places the switch in the OSPF - Line System configuration mode.

The **`no line system`** command removes the Line System configurations from the ***running-config***.

Command Mode

Global Configuration Mode

Command Syntax

`line system`

`no line system`

Parameters

The following paramaters are allowed to be configured under LS mode.

- **port *number*** Transceiver slot number. Value ranges from ***1 - 66***.
  - The following parameters are allowed under **LS port mode**:
    - booster Booster settings
    - pre-amp Pre-amp settings

Example

This command places the switch in the OSPF Line System configuration mode.

```
switch# config
switch(config)# line system
switch(config-ls)#
```

15.2.5.32 log-adjacency-changes (OSPFv2)

The **`log-adjacency-changes`** command enables syslog messages to be sent when it detects OSPFv2 link state changes or when it detects that a neighbor has gone up or down. Log message sending is enabled by default.

The **`default log-adjacency-changes`** command restores the default state by removing the **`log-adjacency-changes`** statement from ***running-config***.

The default option (sending a message only when a neighbor goes up or down) is active when running-config does not contain any form of the command. Entering the command in any form replaces the previous command state in ***running-config***.

The **`no log-adjacency-changes`** disables link state change syslog reporting.

The **`default log-adjacency-changes`** command restores the default state by removing the **`log-adjacency-changes detail`** or **`no log-adjacency-changes`** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

`log-adjacency-changes` detail

`no log-adjacency-changes`

`default log-adjacency-changes`

Examples

- This command configures the switch to send a syslog message when a neighbor goes up or down.

```
switch(config)# router ospf 6
switch(config-router-ospf)# log-adjacency-changes
switch(config-router-ospf)#
```

  - After entering the command, **`show active`** does not display a **`log-adjacency-changes`** statement.

```
switch(config-router-ospf)# show active router ospf 1
```

```
switch(config-router-ospf)#
```

- This command configures the switch to send a Syslog message when it detects any link state change.

```
switch(config-router-ospf)# log-adjacency-changes detail
switch(config-router-ospf)#
```

  - After entering the command, **show active** displays a **log-adjacency-changes detail** command.

```
switch(config-router-ospf)# show active router ospf 1
switch(config-router-ospf)# log-adjacency-changes detail
switch(config-router-ospf)#
```

15.2.5.33 maximum-paths (OSPFv2)

The **maximum-paths** command controls the number of parallel routes that OSPFv2 supports. The default maximum is ***16*** paths.

The **no maximum-paths** and **default maximum-paths** commands restore the maximum number of parallel routes that OSPFv2 supports on the switch to the default value of 16 by placing the **maximum-paths *16*** statement in ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

maximum-paths ***paths***

no maximum-paths

default maximum-paths

Parameters

paths Maximum number of parallel routes.

Value ranges from ***1*** to the number of interfaces available per ECMP group, which is platform dependent.

- **Arad:** Value ranges from ***1 to 128***. Default value is ***128***.
  - **FM6000:** Value ranges from ***1 to 32***. Default value is ***32***.
  - **PetraA:** Value ranges from ***1 to 16***. Default value is ***16***.
  - **Trident:** Value ranges from ***1 to 32***. Default value is ***32***.
  - **Trident II:** Value ranges from ***1 to 128***. Default value is ***128***.

Example

This command configures the maximum number of OSPFv2 parallel paths to ***12***.

```
switch(config)# router ospf 6
switch(config-router-ospf)# maximum-paths 12
switch(config-router-ospf)#
```

15.2.5.34 max-lsa (OSPFv2)

The **max-lsa** command specifies the maximum number of LSAs allowed in the LSDB. Setting the limit to zero removes the LSDB restriction and disables LSA overload actions. Actions triggered by LSDB overload conditions include:

- **Warning**: the switch logs OSPF MAXLSAWARNING if the LSDB contains a specified percentage of the LSA maximum.
- **Temporary shutdown**: when the LSDB exceeds the LSA maximum, OSPFv2 is disabled and does not accept or acknowledge new LSAs. The switch re-starts OSPFv2 after a specified period (the default is five minutes).
- **Permanent shutdown**: the switch permanently disables OSPFv2 after performing a specified number of temporary shutdowns (the default is ***5***). This state usually indicates the need to resolve a network condition that consistently generates excessive LSA packets.

The **`no max-lsa`** and **`default max-lsa`** commands restore all LSA overload parameters to their default settings.

Note: if OSPFv2 has entered permanent shutdown, it can also be restarted by increasing the LSA limit to a value larger than the number of LSAs in the database. Setting the max-LSA value to zero will also restart OSPFv2, and will disable overload protection.

Command Mode

Router-OSPF Configuration

Command Syntax

`max-lsa` ***lsa_num*** [**WARNING**] [**IGNORE_TIME**][**IGNORE_COUNT**][**RESET**]

`no max-lsa`

`default max-lsa`

Parameters

- ***lsa_num*** Maximum number of LSAs. Value ranges from ***0 to 100,000***.
  - ***0*** Disables overload protection.
  - ***1 to 100000*** Specifies maximum value; default value is ***12,000***.
- **WARNING** Warning threshold, as a percentage of the maximum number of LSAs (% of ***lsa_num***).
  - ***no parameter*** Default of ***75%***.
  - ***percent*** Ranges from ***25 to 99***.
- **IGNORE_TIME** Temporary shutdown period (minutes). Options include:
  - ***no parameter*** Default value of ***5*** minutes.
  - **ignore-time *period*** Value ranges from ***1 to 60***.
- **IGNORE_COUNT** Number of temporary shutdowns required to trigger a permanent shutdown.
  - ***no parameter*** Default value of ***5***.
  - **ignore-count *episodes*** Ranges from ***1 to 20***.
- **RESET** Period of not exceeding LSA limit required to reset temporary shutdown counter to zero.
  - ***no parameter*** Default value of ***5*** minutes.
  - **reset-time *r_period*** Ranges from ***1 to 60***.

Example

This command defines an LSA limit of ***8000*** and other parameters.

```
switch(config-router-ospf)# max-lsa 8000 40 ignore-time 6 ignore-count 3 reset-
time 20
switch(config-router-ospf)#
```

15.2.5.35 max-metric router-lsa (OSPFv2)

The **`max-metric router-lsa`** command configures OSPF to include the maximum value in LSA metric fields to keep other network devices from using the switch as a preferred intermediate SPF hop.

The **no max-metric router-lsa** and **default max-metric router-lsa** commands disable the advertisement of a maximum metric.

Command Mode

Router-OSPF Configuration

Command Syntax

max-metric router-lsa [**EXTERNAL**][**STUB**][**STARTUP**][**SUMMARY**]

no max-metric router-lsa [**EXTERNAL**][**STUB**][**STARTUP**][**SUMMARY**]

default max-metric router-lsa [**EXTERNAL**][**STUB**][**STARTUP**][**SUMMARY**]

Parameters

All parameters can be placed in any order.

- **EXTERNAL** Advertised metric value. Values include:
  - ***no parameter*** Default value of ***1***.
  - **external-lsa** Range: ***1 to 16777215***. Default value is ***0xFF0000***.
- **STUB** Advertised metric type. Values include:
  - ***no parameter*** Default value of ***2***.
  - **include-stub**
- **STARTUP** Limit scope of LSAs. Values include:
  - ***no parameter***
  - **on-startup**
  - **on-startup wait-for-bgp**
  - **on-startup** Range: ***5 to 86400***.
- **wait-for-bgp** or an **on-start** time value is not included in ***no*** and **default** commands.
- **SUMMARY** Advertised metric value. Values include:
  - ***no parameter***
  - **summary-lsa**
  - **summary-lsa** Range: ***1 to 16777215***.

Example

This command configures OSPF to to include the maximum value in LSA metric fields until BGP has converged:

```
switch(config-router-ospf)# max-metric router-lsa on-startup wait-for-bgp
switch(config-router-ospf)#
```

15.2.5.36 network area (OSPFv2)

The **network area** command assigns the specified IPv4 subnet to an OSPFv2 area.

The **no network area** and **default network area** commands delete the specified network area assignment by removing the corresponding **network area** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

network ***ipv4_subnet*** **area** ***area_id***

no network ***ipv4_subnet*** **area** ***area_id***

default network ***ipv4_subnet*** **area** ***area_id***

Parameters

- ***ipv4_subnet*** IPv4 subnet. Entry formats include address-prefix (CIDR) or address-wildcard mask. The ***running-config*** stores value in CIDR notation.
- ***area_id*** Area number. ***0*** to ***4294967295*** or ***0.0.0.0*** to ***255.255.255.255***. The ***running-config*** stores value in dotted decimal notation.

Example

These equivalent commands each assign the subnet ***10.1.10.0/24*** to area ***0***.

```
switch(config-router-ospf)# network 10.1.10.0 0.0.0.255 area 0
switch(config-router-ospf)# network 10.1.10.0/24 area 0
switch(config-router-ospf)#
```

15.2.5.37 no area (OSPFv2)

The **no area** command removes the corresponding **area** command from the ***running-config*****:**

- **no/default area not-so-stubby lsa type-7 convert type-5** commands remove the **translate type7 always** parameter without changing the area type.
- **no/default area nssa**, **no/default area stub**, and **no/default area stub no-summary** commands restore the areas type to **normal**.
- The **no/default area default-information-originate** command removes all area commands for the specified area from ***running-config***.
- The **no/default area** command removes all area commands for the specified area from the ***running-config***.
- The **no/default area** command removes all area commands for the specified area from the ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

no area ***area_id*** [**TYPE**]

default area ***area_id*** [**TYPE**]

Parameters

- ***area_id*** area number.
  - Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
  - Area ***0*** (or ***0.0.0.0***) is not configurable; it is always **normal**.
  - The ***running-config*** stores value in dotted decimal notation.
- **TYPE** Area type. Values include:
  - **nssa**
  - **nssa translate type7 always**
  - **stub**
  - **stub no-summary**

Examples

- These commands remove area ***1*** from the running configuration.

```
switch(config)# router ospf 6
switch(config-router-ospf)# no area 1
```

```
switch(config-router-ospf)#
```

- These commands remove area ***10.92.148.17*** as an NSSA.

```
switch(config-router-ospf)# no area 10.92.148.17 nssa
switch(config-router-ospf)#
```

15.2.5.38 packet

This describes packet types that the feature is applied on.

Command Mode

system-feature-source-profile **(config-hw-tcam-profile-*profile*-feature-*feature*)**

Command Syntax

packet *packet header tokens* forwarding<[bridged | routed | mpls][multicast][decap]

Parameters

- ***packet header tokens*** The packet header is described as a series of CLI packet header tokens after the **packet** token. It starts from the outer most header after Ethernet. For example, a regular IPv4 packet is **packet ipv4** and a VXLAN packet is **packet ipv4 vxlan eth ipv4**.
- **forwarding** The **forwarding** token indicates the forwarding type of the packet.
  - **bridged**
  - **routed**
  - **mpls**
    - **multicast** Indicates if the packet is a multicast packet.
    - **decap** Indicates if the packet is decapsulated after a tunnel.

Guidelines

On DCS-7020, DCS-7280R/R2 or DCS-7500R/R2, enabling OSPF routes over GRE tunnels requires the system TCAM profile to have “Tunnel IPv4” feature enabled so that control packets such as OSPF hellos received over GRE tunnel interfaces are appropriately classified. This can be achieved by creating a user defined TCAM profile.

The user defined TCAM profile may be created either manually from scratch or by copying from an existing TCAM profile. The system TCAM profile must have the feature **tunnel ipv4** for OSPFv2 over GRE tunnel interfaces to work.This is applicable regardless of whether the TCAM profile is copied from an existing profile or created from scratch.

To create a user defined TCAM profile.

```
(config)# hardware tcam
(config-hw-tcam)# profile profilename copy default
(config-hw-tcam-profile-profile)# feature tunnel ipv4 copy system-feature-
source-profile
```

The following steps are optional if the feature is added by copying from system-feature-source-profile.

- Set the packet types for the feature as follows to match GRE tunnelled ipv4 routed unicast and multicast packets.

```
(config-hw-tcam-profile-profile-feature-feature)# packet ipv4 non-vxlan
 forwarding routed decap
(config-hw-tcam-profile-profile-feature-feature)# packet ipv4 non-vxlan
 forwarding routed multicast decap
```

- Specify the qualifiers to match on.

```
(config-hw-tcam-profile-profile-feature-feature)# key field inner-dst-ip
 inner-ip-protocol inner-l4-dst-port inner-l4-src-port inner-ttl
```

- Set the key size limit to ***160***.

```
(config-hw-tcam-profile-profile-feature-feature)#key size limit 160
(config-hw-tcam-profile-profile-feature-feature)#exit
```

It maybe necessary to disassociate some features which are not applicable to GRE encapsulated packets from the GRE TCAM program to make room for the **tunnel ipv4** feature.

Related Commands

- system profile
- interface Tunnel
- show hardware tcam profile

15.2.5.39 passive-interface default (OSPFv2)

The **passive-interface default** command configures all interfaces as OSPFv2 passive by default. The switch advertises the passive interface as part of the router LSA.

The **passive-interface (OSPFv2)** configures the OSPFv2 active-passive status for a specific interface:

- When **passive-interface default** is not set, all interfaces are OSPFv2 active by default and passive interfaces are denoted by **passive-interface** statements in ***running-config***.
- When **passive-interface default** is set, all interfaces are OSPFv2 passive by default and active interfaces are denoted by **no passive-interface** statements in ***running-config***.

The **no passive-interface** and **default passive-interface** commands configures all interfaces as OSPFv2 active by default by removing the **passive-interface default** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

passive-interface default

no passive-interface default

default passive-interface default

Examples

- This command configures the default interface setting as OSPFv2 passive. This command also removes all **passive-interface** statements from the ***running-config***.

```
switch(config)# router ospf 6
switch(config-router-ospf)# passive-interface default
switch(config-router-ospf)#
```

- This command configures the default interface setting as OSPFv2 active. This command also removes all **no passive-interface** statements from the ***running-config***.

```
switch(config-router-ospf)# no passive-interface default
switch(config-router-ospf)#
```

15.2.5.40 passive-interface (OSPFv2)

The **passive-interface** command disables OSPFv2 on an interface range. The switch advertises the passive interface as part of the LSA.

The default OSPFv2 interface activity is configured by the **passive-interface default (OSPFv2)** command:

- When **passive-interface default** is not set, all interfaces are OSPFv2 active by default and passive interfaces are denoted by **passive-interface** statements in the ***running-config***.
- When **passive-interface default** is set, all interfaces are OSPFv2 passive by default and active interfaces are denoted by **no passive-interface** statements in the ***running-config***.

The **no passive-interface** command enables OSPFv2 on the specified interface range. The **default passive-interface** command sets the interface to the default interface activity setting by removing the corresponding **passive-interface** or **no passive-interface** statement from the ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

passive-interface INTERFACE_NAME

no passive-interface INTERFACE_NAME

default passive-interface INTERFACE_NAME

Parameters

- **INTERFACE_NAME** Interface to be configured. Options include:
  - **ethernet *e_range***
  - **port-channel *p_range***
  - **vlan *v_range***
  - **vxlan *vx_range***

Examples

- These commands configure Ethernet interfaces ***2*** through ***5*** as passive interfaces.

```
switch(config)# router ospf 6
switch(config-router-ospf)# passive-interface ethernet 2-5
switch(config-router-ospf)#
```

- This command configures VLAN interfaces ***50-54***, ***61***, ***68***, and ***102-120*** as passive interfaces.

```
switch(config-router-ospf)# passive-interface vlan 50-54,61,68,102-120
switch(config-router-ospf)#
```

- This command configures ***vlan 2*** as an active interface.

```
switch(config-router-ospf)# no passive-interface vlan 2
switch(config-router-ospf)#
```

15.2.5.41 point-to-point routes (OSPFv2)

The **point-to-point routes** command enables the switch to maintain a local Routing Information Base (RIB) to store information it learns from its neighbors.

The **no point-to-point routes** and **default point-to-point routes** commands program the switch to include point-to-point links in its RIB by removing the **point-to-point routes** command from the ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

point-to-point routes

no point-to-point routes

default point-to-point routes

Examples

- This command configures the switch to optimize the local RIB by not including point-to-point routes.

```
switch(config)# router ospf 6
switch(config-router-ospf)# no point-to-point routes
switch(config-router-ospf)#
```

- This command configures the switch to include point-to-point routes.

```
switch(config-router-ospf)# point-to-point routes
switch(config-router-ospf)#
```

15.2.5.42 redistribute (OSPFv2)

The **redistribute** command enables the advertising of all specified routes on the switch into the OSPFv2 domain as external routes.

The **no redistribute** and **default redistribute** commands remove the corresponding **redistribute** command from the ***running-config***, disabling route redistribution for the specified route type.

Command Mode

Router-OSPF Configuration

Command Syntax

redistribute **ROUTE_TYPE** [**ROUTE_MAP**]

no redistribute **ROUTE_TYPE** [**ROUTE_MAP**]

default redistribute **ROUTE_TYPE** [**ROUTE_MAP**]

Parameters

- **ROUTE_TYPE** Source from which routes are redistributed. Options include:
  - **connected** routes that are established when IPv4 is enabled on an interface.
  - **BGP** routes from a BGP domain.
  - **RIP** routes from a RIP domain.
  - **static** IP static routes.
- **ROUTE_MAP** Route map that determines the routes that are redistributed. Options include:
  - ***no parameter***
  - **route-map** ***map_name***

Examples

- The **redistribute static** command starts the advertising of static routes as OSPFv2 external routes.

```
switch(config)# router ospf 6
switch(config-router-ospf)# redistribute static
switch(config-router-ospf)#
```

- The **no redistribute bgp** command stops the advertising of BGP routes as OSPFv2 external routes.

```
switch(config-router-ospf)# no redistribute bgp
switch(config-router-ospf)#
```

15.2.5.43 redistribute ospf

Redistributing connected routes causes the OSPFv2 instance to advertise all connected routes on the switch as external OSPFv2 routes. Connected routes are routes that are established when IPv4 is enabled on an interface.

Command Mode

config-router-bgp

Command Syntax

redistribute ospf [include [leaked] | match [external | internal | nssa-external] | route-map ***word***]

no redistribute ospf [match [external | internal | nssa-external]

default redistribute ospf [match [external | internal | nssa-external]

Parameters

- **include** Include following routes while redistributing.
  - **leaked** Include leaked routes of this protocol while redistributing.
- **match** Routes learned by the OSPF protocol.
  - **external** OSPF routes learned from external sources.
  - **internal** OSPF routes learned from internal sources.
  - **nssa-external** OSPF routes learned from external NSSA sources.
- **route-map** Name a router map.
  - ***word*** Route map name.

Example

Use the following commands to redistribute OSPFv2 routes into the BGP domain.

```
switch(config)# router bgp 1
switch(config-router-bgp)# redistribute OSPF
switch(config-router-bgp)#
```

15.2.5.44 redistribute ospf instance

The **redistribute ospf instance** command redistributes either the non-leaked routes, or both leaked and non-leaked routes. The **exit** command returns the switch to the ***global*** configuration mode.

Command Mode

Router-OSPF Configuration

Command Syntax

redistribute ospf instance [**OPTIONS**]

Parameters

- **include** Include leaked routes.
  - ***leaked*** OSPF leaked routes.
- **match** Routes learned by the OSPF protocol.

- **external** OSPF routes learned from external sources.
- **internal** OSPF routes learned from internal sources.
- **nssa-external** OSPF routes learned from external NSSA sources.

Examples

- This command redistributes the OSPFv2 external routes from all other OSPFv2 instances in the same VRF into the given instance.

```
switch(config-router-ospf)# redistribute ospf instance match external
```

- This command redistributes the OSPFv2 internal leaked and non-leaked routes from all other instances in all VRFs into the given instance.

```
switch(config-router-ospf)# redistribute ospf instance include leaked match
 internal
```

15.2.5.45 router ospf

The **router ospf** command places the switch in router-ospf configuration mode. The switch will create a process ID for the new instance if one does not already exist. The **exit** command returns the switch to the ***global*** configuration mode.

The **show ip ospf** command displays the process ID of the OSPFv2 instances configured on the switch.

The **no router ospf** and **default router ospf** commands delete the specified OSPFv2 instance.

The ***router-ospf*** configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting ***router-ospf*** configuration mode does not affect ***running-config***. The **exit** command returns the switch to the ***global*** configuration mode.

Refer to the Router-OSPFv2 Configuration Mode for a list of commands available in ***router-ospf*** configuration mode.

Command Mode

Global Configuration

Command Syntax

router ospf ***process_id*** [**VRF_INSTANCE**]

no router ospf ***process_id*** [**VRF_INSTANCE**]

default router ospf ***process_id*** [**VRF_INSTANCE**]

Parameters

- ***process_id*** OSPFv2 process ID. Values range from ***1 to 65535***.
- **VRF_INSTANCE**
  - ***no parameter*** Configures the default VRF instance.
  - **vrf** ***vrf_name*** Configures the vrf_name instance.

Examples

- This command creates an OSPFv2 instance with process ID ***145*** in the main VRF.

```
switch(config)# router ospf 145
switch(config-router-ospf)#
```

- This command deletes the specified OSPFv2 instance.

```
switch(config)# no router ospf 145
```

```
switch(config)#
```

15.2.5.46 router-id (OSPFv2)

The **router-id** command assigns a router ID for an OSPFv2 instance. This number uniquely identifies the router within an Autonomous System. Status commands use the router ID to identify the switch.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IP address, if a loopback interface is configured on the switch.
3. The highest IP address present on the router.



Note: When configuring VXLAN on an MLAG, always manually configure the OSPFv2 router ID to prevent the switch from using the common VTEP IP address as the router ID.

The **no router-id** and **default router-id** commands remove the router ID command from the ***running-config***; the switch uses the loopback or highest address as the router ID.

Command Mode

Router-OSPF Configuration

Command Syntax

router-id [***identifier***]

no router-id [***identifier***]

default router-id [***identifier***]

Parameters

identifier Value ranges from ***0.0.0.0*** to ***255.255.255.255***.

Example

This command assigns ***10.5.4.2*** as the router ID for the OSPFv2 instance.

```
switch(config)# router ospf 6
switch(config-router-ospf)# router-id 10.5.4.2
switch(config-router-ospf)#
```

15.2.5.47 show hardware tcam profile

Use the **show hardware tcam profile** command to verify that the user-defined-tcam profile is applied correctly without errors on the DCS-7020, DCS-7280R/R2, or DCS-7500R/R2 platforms.

This command is applicable the following platforms:

- DCS-7280E
- DCS-7280R
- DCS-7280R2
- DCS-7020R
- DCS-7500E
- DCS-7500R
- DCS-7500R2

Command Mode

EXEC

Command Syntax

show hardware tcam profile [***profile***] detail

Parameters

- **tcam** Specifies the TCAM information.
- **profile *profile*** Specifies the TCAM profile information.
- **detail** Displays detailed tcam profile information.

Example

This example displays the detailed tcam profile information for profile named ***newprofile1***.

```
(config-hw-tcam)# show hardware tcam profile newprofile1 detail
Profile newprofile1 [ FixedSystem ]
 Feature         mpls
 --------------- ----------------------------------------------------
 Key size        160
 Actions         drop, redirect, set-ecn
 Packet type     ipv4 mpls ipv4 forwarding mpls decap
                 ipv4 mpls ipv6 forwarding mpls decap
                 mpls ipv4 forwarding mpls
                 mpls ipv6 forwarding mpls
                 mpls non-ip forwarding mpls

 Feature         acl vlan ipv6
 --------------- ----------------------------------------------------
 Key size        320
 Key fields      dst-ipv6, ipv6-next-header, l4-dst-port, l4-src-port,
                 src-ipv6-high, src-ipv6-low, tcp-control
 Actions         count, drop, mirror, redirect
 Packet type     ipv6 forwarding routed
...
```

15.2.5.48 show ip ospf border-routers

The **show ip ospf border-routers** command displays the internal OSPFv2 routing table entries to Area Border Routers (ABRs) and Autonomous System Boundary Routers (ASBRs) for each of the OSPFv2 areas.

Command Mode

EXEC

Command Syntax

show ip ospf border-routers [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE Specifies the VRF instance.

- ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
- **vrf *vrf_name*** displays information from the specified VRF.

Example

This command displays the ABRs and ASBRs.

```
switch# show ip ospf border-routers
OSPF Process 10.17.0.42, VRF default

Router ID       Area            Type
10.17.0.1      0.0.0.0         ASBR switch>
```

15.2.5.49 show ip ospf database database-summary

The **show ip ospf database database-summary** command displays the number of link state advertisements in the OSPFv2 database.

Command Mode

EXEC

Command Syntax

show ip ospf [**AREA**] **database database-summary** [**VRF_INSTANCE**]

Parameters

- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name*** displays information from the specified VRF.
- **AREA** areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - ***no parameter***
  - ***process_id***
  - ***process_id area_id***
  - ***process_id*** input range: ***1 to 65535***.
  - ***area_id*** input range: ***0 to 4294967295*** or ***0.0.0.0 to 255.255.255.255***.

Example

This command displays the LSDB content summary for area ***0***.

```
switch# show ip ospf 1 0 database database-summary

  LSA Type      Count
  Router        18
  Network       21
  Summary Net   59
  Summary ASBR  4
  Type-7 Ext    0
  Opaque Area   0
  Type-5 Ext    4238
  Opaque AS     0
  Total         4340 switch>
```

15.2.5.50 show ip ospf database <link state list>

The **show ip ospf database <link state list>** command displays the OSPFv2 link state advertisements that originate on a specified switch.

Command Mode

EXEC

Command Syntax

show ip ospf [**AREA**] **database**[**ROUTER**] [**VRF_INSTANCE**]

Parameters

- **AREA** Areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - ***no parameter***
  - ***process_id***

- ***process_id area_id***
- ***process_id*** value ranges from ***1 to 65535***.
- ***area_id*** is entered in decimal or dotted decimal notation.

- **ROUTER** Router or switch for which the command provides data. Options include:
  - ***no parameter***
  - **adv-router** [***a.b.c.d***]
  - **self-originate**
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name*** displays information from the specified VRF.

Example

This command displays OSPFv2 LSAs that originate at the router with a router ID of ***10.26.0.31***.

```
switch# show ip ospf database adv-router 10.26.0.31

OSPF Router with ID(10.26.0.23) (Process ID 1) (VRF default)

10.26.0.3110.26.0.319180x80002b4a0x13153

Type-5 AS External Link States

Link IDADV RouterAgeSeq#Checksum
10.24.238.23810.26.0.316780x800003d20x8acf0
10.24.238.24410.26.0.316780x800003d20x4e060
10.24.238.22410.26.0.316780x800003d20x17510
<-------OUTPUT OMITTED FROM EXAMPLE-------->

Type 11 Opaque LSDB

TypeLink IDADV RouterAgeSeq# Checksum switch>
```

15.2.5.51 show ip ospf database <link-state details>

The **`show ip ospf database`** command displays details of the specified link state advertisements.

Command Mode

EXEC

Command Syntax

`show ip ospf` [**AREA**] **`database`** **LINKSTATE_TYPE** ***linkstate_id*** [**ROUTER**] [**VRF_INSTANCE**]

Parameters

- **AREA** Areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - ***no parameter***
  - ***process_id***
  - ***process_id area_id***
  - ***process_id*** input range: ***1 to 65535***.
  - ***area_id*** input range: ***0 to 4294967295*** or ***0.0.0.0 to 255.255.255.255***.
- **LINKSTATE_TYPE** Link state types. Parameter options include:
  - **detail** Displays all link states.
  - **router**
  - **network**

- **summary**
- **asbr-summary**
- **external**
- **nssa-external**
- **opaque-link**
- **opaque-area**
- **opaque-as**

- ***linkstate_id*** Network segment described by the LSA (dotted decimal notation).

  Value depends on the LSA type.
- **ROUTER** Router or switch for which the command provides data. Options include:
  - ***no parameter***
  - **adv-router** [***a.b.c.d***]
  - **self-originate**
- **VRF_INSTANCE** Parameter has no effect; this command displays information about the specified process and area regardless of VRF.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name*** displays information from the specified VRF.

Examples

- This command displays the router link states contained in the area ***2*** LSDB.

```
switch# show ip ospf 1 2 database router

OSPF Router with ID(10.168.103.1) (Process ID 1) (VRF default)

Router Link States (Area 0.0.0.2)

LS age: 00:02:16
Options: (E DC)
LS Type: Router Links
Link State ID: 10.168.103.1
Advertising Router: 10.168.103.1
LS Seq Number: 80000032
Checksum: 0x1B60
Length: 36
Number of Links: 1

Link connected to: a Transit Network
(Link ID) Designated Router address: 10.168.2.1
(Link Data) Router Interface address: 10.168.2.1
Number of TOS metrics: 0
TOS 0 Metrics: 10


LS age: 00:02:12
Options: (E DC)
LS Type: Router Links
Link State ID: 10.168.104.2
Advertising Router: 10.168.104.2
LS Seq Number: 80000067
Checksum: 0xA29C
Length: 36
Number of Links: 1

Link connected to: a Transit Network
(Link ID) Designated Router address: 10.168.2.1
(Link Data) Router Interface address: 10.168.2.2
```

```
Number of TOS metrics: 0
TOS 0 Metrics: 10 switch>
```

- This command displays Link State DataBase (LSDB) contents for area ***2***.

```
switch# show ip ospf 1 2 database

OSPF Router with ID(10.168.103.1)(Process ID 1) (VRF default)

Router Link States (Area 0.0.0.2)

Link IDADV RouterAgeSeq#Checksum Link count
10.168.103.110.168.103.100:29:080x80000031 0x001D5F 1
10.168.104.210.168.104.200:29:090x80000066 0x00A49B 1

Net Link States (Area 0.0.0.2)

Link IDADV RouterAgeSeq#Checksum
10.168.2.110.168.103.100:29:080x80000001 0x00B89D

Summary Net Link States (Area 0.0.0.2)

Link IDADV RouterAgeSeq#Checksum
10.168.0.010.168.103.100:13:200x80000028 0x0008C8
10.168.0.010.168.104.200:09:160x80000054 0x00A2FF
10.168.3.010.168.104.200:24:160x80000004 0x00865F
10.168.3.010.168.103.100:24:200x80000004 0x002FC2
10.168.103.010.168.103.100:14:200x80000028 0x0096D2
10.168.103.010.168.104.200:13:160x80000004 0x00364B
10.168.104.010.168.104.200:08:160x80000055 0x002415
10.168.104.010.168.103.100:13:200x80000028 0x00EF6E switch>
```

15.2.5.52 show ip ospf interface brief

The **`show ip ospf interface brief`** command displays a summary of OSPFv2 information.

Command Mode

EXEC

Command Syntax

`show ip ospf` [**PROCESS_ID**] **`interface brief`** [**VRF_INSTANCE**]

Parameters

- **PROCESS_ID** OSPFv2 process ID. Values include:
  - ***no parameter***
  - ***1 to 65535***
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name*** displays information from the specified VRF.

Related Command

`show ip ospf interface`

Example

This command displays a summary of interface information for the switch.

```
switch# show ip ospf interface brief
InterfacePIDAreaIP AddressCostStateNbrs
```

```
Loopback010.0.0.010.168.103.1/2410DR0
Vlan110.0.0.010.168.0.1/2410BDR1
Vlan210.0.0.210.168.2.1/2410BDR1
Vlan310.0.0.310.168.3.1/2410DR0 switch>
```

15.2.5.53 show ip ospf interface

The **`show ip ospf interface`** command displays interface information that is related to OSPFv2.

Command Mode

EXEC

Command Syntax

`show ip ospf` [**PROCESS_ID**] **`interface`** [**INTERFACE_NAME**][**VRF_INSTANCE**]

Parameters

- **PROCESS_ID** OSPFv2 process ID. Values include:
  - ***no parameter***
  - ***1 to 65535***
- **INTERFACE_NAME** Interface type and number. Values include:
  - ***no parameter***
  - **ethernet *e_num***
  - **loopback *l_num***
  - **port-channel *p_num***
  - **vlan *v_num***
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name*** displays information from the specified VRF.

Related Command

`show ip ospf interface brief`

Example

This command displays complete OSPFv2 information for ***vlan 1***.

```
switch# show ip ospf interface vlan 1
Vlan1 is up, line protocol is up (connected)
Internet Address 10.168.0.1/24, VRF default, Area 0.0.0.0
Process ID 1, Router ID 10.168.103.1, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router is 10.168.104.2
Backup Designated router is 10.168.103.1
Timer intervals configured, Hello 10, Dead 40, Retransmit 5
Neighbor Count is 1
MTU is 1500 switch>
```

15.2.5.54 show ip ospf lsa-log

The **`show ip ospf lsa-log`** command displays log entries when LSA update messages are sent or received for OSPF.

Command Mode

EXEC

Command Syntax

show ip ospf [**PROCESS_ID**] **ospf-log**

Parameters

PROCESS_ID OSPFv2 process ID. Values include:

- ***no parameter***
- ***1 to 65535***

Example

This command displays log entries when LSA update messages are sent or received for OSPF.

```
switch# show ip ospf lsa-log
OSPF Process 3.3.3.3, LSA Throttling Log:
[04:21:09] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 2000 msecs
[04:21:08] type 1: 3.3.3.3/32 [3.3.3.3], event 2, backoff restarted, new hold
 value 900 msecs
[04:21:00] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 3000 msecs
[04:21:00] type 1: 3.3.3.3/32 [3.3.3.3], event 4, maxwait value changed, new
 hold value 3000 msecs
/* Here the maxwait value was changed to 3000 from earlier 32000, this is not
 part of the log */
[04:20:42] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 32000 msecs
[04:20:10] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 32000 msecs
[04:19:54] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 16000 msecs
[04:19:46] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 8000 msecs
[04:19:42] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 4000 msecs
[04:19:40] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 2000 msecs
[04:19:39] type 1: 3.3.3.3/32 [3.3.3.3], event 2, backoff restarted, new hold
 value 900 msecs
[04:19:22] type 1: 4.4.4.4/32 [4.4.4.4], event 3, discarded, was early by 995
 msecs
[04:19:22] type 1: 3.3.3.3/32 [3.3.3.3], event 0, backoff started, new hold
 value 1000 msecs switch#
```

15.2.5.55 show ip ospf neighbor adjacency-changes

The **show ip ospf neighbor adjacency-changes** command displays the OSPFv2 neighbor adjacency change log for specified interfaces.

Command Mode

EXEC

Command Syntax

show ip ospf neighbor [**INTERFACE_NAME**][**NEIGHBOR**] **adjacency-changes** [**VRF_INSTANCE**]

Parameters

- **INTERFACE_NAME** Interface type and number. Values include:
  - ***no parameter***
  - **ethernet *e_num***
  - **loopback *l_num***
  - **port-channel *p_num***

- **vlan *v_num***
- **NEIGHBOR** OSPFv2 neighbor. Options include:
  - ***no parameter***
  - ***ipv4_addr***
  - ***host_name***
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name*** displays information from the specified VRF.

Example

This command displays the adjacency changes to ***vlan 2***.

```
switch# show ip ospf neighbor vlan 2 adjacency-changes
[08-04 08:55:32] 10.168.104.2, interface Vlan2 adjacency established
[08-04 09:58:51] 10.168.104.2, interface Vlan2 adjacency dropped: interface
 went down
[08-04 09:58:58] 10.168.104.2, interface Vlan2 adjacency established
[08-04 09:59:34] 10.168.104.2, interface Vlan2 adjacency dropped: interface
 went down
[08-04 09:59:42] 10.168.104.2, interface Vlan2 adjacency established
[08-04 10:01:40] 10.168.104.2, interface Vlan2 adjacency dropped: nbr did not
list our router ID
[08-04 10:01:46] 10.168.104.2, interface Vlan2 adjacency established switch>
```

15.2.5.56 show ip ospf neighbor state

The **`show ip ospf neighbor state`** command displays the state information on OSPF neighbors on a per-interface basis.

Command Mode

EXEC

Command Syntax

`show ip ospf neighbor state` **STATE_NAME** [**VRF_INSTANCE**]

Parameters

- **STATE_NAME** Output filtered by the devices OSPF state. Options include valid OSPF states:
  - **2-ways**
  - **attempt**
  - **down**
  - **exch-start**
  - **exchange**
  - **full**
  - **graceful-restart**
  - **init**
  - **loading**
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name*** displays information from the specified VRF.

Example

This command displays OSPF information for neighboring routers that are fully adjacent.

```
switch# show ip ospf neighbor state full
Neighbor ID     VRF       Pri   State            Dead Time   Address         Interface
Test1           default     1   FULL/BDR         00:00:35    10.17.254.105   Vlan3912
Test2           default     1   FULL/BDR         00:00:36    10.17.254.29    Vlan3910
Test3           default     1   FULL/DR          00:00:35    10.25.0.1       Vlan101
Test4           default     1   FULL/DROTHER     00:00:36    10.17.254.67    Vlan3908
Test5           default     1   FULL/DROTHER     00:00:36    10.17.254.68    Vlan3908
Test6           default     1   FULL/BDR         00:00:32    10.17.254.66    Vlan3908
Test7           default     1   FULL/DROTHER     00:00:34    10.17.36.4      Vlan3036
Test8           default     1   FULL/BDR         00:00:35    10.17.36.3      Vlan3036
Test9           default     1   FULL/DROTHER     00:00:31    10.17.254.13    Vlan3902
Test10          default     1   FULL/BDR         00:00:37    10.17.254.11    Vlan3902
Test11          default     1   FULL/DROTHER     00:00:33    10.17.254.163   Vlan3925
Test12          default     1   FULL/DR          00:00:37    10.17.254.161   Vlan3925
Test13          default     1   FULL/DROTHER     00:00:31    10.17.254.154   Vlan3923
Test14          default     1   FULL/BDR         00:00:39    10.17.254.156   Vlan3923
Test15          default     1   FULL/DROTHER     00:00:33    10.17.254.35    Vlan3911
Test16          default     1   FULL/DR          00:00:34    10.17.254.33    Vlan3911
Test17          default     1   FULL/DR          00:00:36    10.17.254.138   Ethernet12
Test18          default     1   FULL/DR          00:00:37    10.17.254.2     Vlan3901 switch>
```

15.2.5.57 show ip ospf neighbor summary

The **show ip ospf neighbor** summary command displays a single line of summary information for each OSPFv2 neighbor.

Command Mode

EXEC

Command Syntax

show ip ospf [**PROCESS_ID**] **neighbor summary** [**VRF_INSTANCE**]

Parameters

- **PROCESS_ID** OSPFv2 process ID. Values include:
  - ***no parameter***
  - ***1 to 65535***
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name*** displays information from the specified VRF.

Example

This command displays the summary information for the OSPFv2 neighbors.

```
switch# show ip ospf neighbor summary
OSPF Router with (Process ID 1) (VRF default)
0 neighbors are in state DOWN
0 neighbors are in state GRACEFUL RESTART
2 neighbors are in state INIT
0 neighbors are in state LOADING
0 neighbors are in state ATTEMPT
18 neighbors are in state FULL
0 neighbors are in state EXCHANGE
0 neighbors are in state 2 WAYS
0 neighbors are in state EXCH START switch>
```

15.2.5.58 show ip ospf neighbor

The **show ip ospf neighbor** command displays OSPFv2 neighbor information for specified interfaces.

Command Mode

EXEC

Command Syntax

show ip ospf [**PROCESS_ID**] **neighbor** [**INTERFACE_NAME**] [**NEIGHBOR**] [**DATA**] [**VRF_INSTANCE**]

Parameters

- **PROCESS_ID** OSPFv2 process ID. Values include:
  - ***no parameter***
  - ***1 to 65535***
- **INTERFACE_NAME** Interface type and number. Values include:
  - ***no parameter***
  - **ethernet *e_num***
  - **loopback *l_num***
  - **port-channel *p_num***
  - **vlan *v_num***
- **NEIGHBOR** OSPFv2 neighbor. Options include:
  - ***no parameter***
  - ***ipv4_addr***
- **DATA** Type of information the command displays. Values include:
  - ***no parameter***
  - **detail**
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
  - **vrf *vrf_name***displays information from the specified VRF.

Examples

- This command displays the switchs neighbors.

```
switch# show ip ospf neighbor
Neighbor IDVRFPriStateDead TimeAddressInterface
10.168.104.2default1FULL/DR00:00:3510.168.0.2Vlan1
10.168.104.2default8FULL/BDR00:00:3110.168.2.2Vlan2 switch>
```

- This command displays details about the neighbors to ***vlan 2***.

```
switch# show ip ospf neighbor vlan 2 detail
Neighbor 10.168.104.2, VRF default, interface address 10.168.2.2
In the area 0.0.0.2 via interface Vlan2
Neighbor priority is 8, State is FULL, 13 state changes
Adjacency was established 000:01:25:48 ago
DR is 10.168.2.1 BDR is 10.168.2.2
Options is E
Dead timer due in 00:00:34 switch>
```

15.2.5.59 show ip ospf request queue

The **show ip ospf request queue** command displays a list of all OSPFv2 Link State Advertisements (LSAs) requested by a router.

Command Mode

EXEC

Command Syntax

show ip ospf request queue [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE Specifies the VRF instance.

- ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
- **vrf *vrf_name*** displays information from the specified VRF.

Example

This command displays an LSA request list.

```
switch# show ip ospf request queue
Neighbor 10.168.104.2 vrf default interface: 10.168.0.2 address vlan1
Type LS ID ADV RTR Seq No Age Checksum
Neighbor 10.168.104.2 vrf default interface: 10.168.2.2 address vlan2
Type LS ID ADV RTR Seq No Age Checksum switch>
```

15.2.5.60 show ip ospf retransmission queue

The **show ip ospf retransmission queue** command displays a list of all OSPFv2 Link State Advertisements (LSAs) waiting to be re-sent.

Command Mode

EXEC

Command Syntax

show ip ospf retransmission queue [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE Specifies the VRF instance.

- ***no parameter*** displays information from all VRFs, or from context-active VRF if set.
- **vrf *vrf_name*** displays information from the specified VRF.

Example

This command displays an empty retransmission list.

```
switch# show ip ospf retransmission queue
Neighbor 10.168.104.2 vrf default interface vlan1 address 10.168.0.2
LSA retransmission not currently scheduled. Queue length is 0

TypeLink IDADV RouterAgeSeq# Checksum
Neighbor 10.168.104.2 vrf default interface vlan2 address 10.168.2.2
LSA retransmission not currently scheduled. Queue length is 0

TypeLink IDADV RouterAgeSeq# Checksum switch>
```

15.2.5.61 show ip ospf spf-log

The **show ip ospf spf-log** command displays when and how long the switch took to run a full SPF calculation for OSPF.

Command Mode

EXEC

Command Syntax

show ip ospf [**PROCESS_ID**] **ospf-log**

Parameters

PROCESS_ID OSPFv2 process ID. Values include:

- ***no parameter***
- ***1 to 65535***

Example

This command displays the SPF information for OSPF.

```
switch# show ip ospf spf-log
OSPF Process 172.26.0.22
When      Duration(msec)
13:01:34  1.482
13:01:29  1.547
13:01:24  1.893
13:00:50  1.459
13:00:45  1.473
13:00:40  2.603
11:01:49  1.561
11:01:40  1.463
11:01:35  1.467
11:01:30  1.434
11:00:54  1.456
11:00:49  1.472
11:00:44  1.582
15:01:49  1.575
15:01:44  1.470
15:01:39  1.679
15:01:34  1.601
15:00:57  1.454
15:00:52  1.446
15:00:47  1.603 switch>
```

15.2.5.62 show ip ospf

The **show ip ospf** command displays OSPFv2 routing information

Command Mode

EXEC

Command Syntax

show ip ospf [PROCESS_ID][VRF_INSTANCE]

Parameters

- **PROCESS_ID** OSPFv2 process ID. Values include:
  - ***no parameter***
  - ***1 to 65535***
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter*** configures the default VRF instance.
  - **vrf *vrf_name*** configures the ***vrf_name*** instance.

Example

This command displays configuration parameters, operational statistics, status of the OSPFv2 instance, and a brief description of the areas on the switch.

```
switch# show ip ospf
Routing Process "ospf 1" with ID 10.168.103.1 VRF default
Supports opaque LSA
Maximum number of LSA allowed 12000
Threshold for warning message 75%
Ignore-time 5 minutes, reset-time 5 minutes
Ignore-count allowed 5, current 0
It is an area border router
Hold time between two consecutive SPFs 5000 msecs
SPF algorithm last executed 00:00:09 ago
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msecs
Number of external LSA 0. Checksum Sum 0x000000
Number of opaque AS LSA 0. Checksum Sum 0x000000
Number of LSA 27.
Number of areas in this router is 3. 3 normal 0 stub 0 nssa
Area BACKBONE(0.0.0.0)
Number of interfaces in this area is 2
It is a normal area
Area has no authentication
SPF algorithm executed 153 times
Number of LSA 8. Checksum Sum 0x03e13a
Number of opaque link LSA 0. Checksum Sum 0x000000
Area 0.0.0.2
Number of interfaces in this area is 1
It is a normal area
Area has no authentication
SPF algorithm executed 153 times
Number of LSA 11. Checksum Sum 0x054e57
Number of opaque link LSA 0. Checksum Sum 0x000000
Area 0.0.0.3
Number of interfaces in this area is 1
It is a normal area
Area has no authentication
SPF algorithm executed 5 times
Number of LSA 6. Checksum Sum 0x02a401
Number of opaque link LSA 0. Checksum Sum 0x000000
```

15.2.5.63 show line system dom thresholds

The **show line system dom thresholds** command reports DOM information reported by the OSFP-LS module. This includes standard fields such as temperature and voltage. In addition to standard DOM fields, the OSFP-LS also monitors the laser temperature for each of its amplifiers. The reported RX power reflects the total RX power seen on that path. TX bias current monitoring is not supported on these modules and should be ignored.

Command Mode

EXEC

Command Syntax

show line system [[port **RANGE**] **dom thresholds**]

Example

This command displays the DOM information reported by the OSFP-LS module.

```
switch# show line system port 10 dom thresholds
Ch: Channel, mA: milliamperes, dBm: decibels (milliwatts),
C: Celsius, V: Volts, NA or N/A: not applicable.
```

```
Port 10
Last update: 0:00:04 ago
                                        High Alarm  High Warn   Low Warn  Low Alarm
                                  Value  Threshold  Threshold  Threshold  Threshold  Unit Indicator
                                -------------------------------------------------------------------
   Temperature                    32.83      70.00      65.00       0.00      -5.00     C
   Voltage                         3.29       3.47       3.37       3.23       3.14     V
Booster
   TX bias current                  N/A        N/A        N/A        N/A        N/A    mA
   Optical TX power (line)        -3.58       7.96       7.50      -9.03     -15.06   dBm
   Optical RX power (local)       -28.24     -16.23     -17.26     -30.00     -33.01   dBm
   Laser Temperature              43.55      80.00      75.00      -5.00     -10.00     C
Pre-amp
   TX bias current                  N/A        N/A        N/A        N/A        N/A    mA
   Optical TX power (local)       -3.38       7.96       7.50      -9.03     -15.06   dBm
   Optical RX power (line)        -15.42      5.77       4.77      -28.24     -30.97   dBm
   Laser Temperature              43.54      80.00      75.00      -5.00     -10.00     C
```

15.2.5.64 show line system status

The **`show line system status`** command displays module status. The OSFP-LS is compliant to the Common Management Interface Specification (CMIS), and implements various CMIS-defined status flags. Data path 1 reflects the outgoing booster path and data path 2 reflects the incoming pre-amp path.

Command Mode

EXEC

Command Syntax

`show line system` [port **RANGE**] **`status`**

Example

This command displays the displays module status.

```
switch(config-ls-port10,19)# show line system status
                              Current State        Changes       Last Change
                              -------------        -------       -----------
Port 10
  Transceiver                 AMP-ZR                     3       0:23:03 ago
  Transceiver SN              XDG203505010
  Presence                    present
  Adapters                    none
  Bad EEPROM checksums                                   0       never
  Resets                                                 0       0:23:08 ago
  Interrupts                                             0       never
  Data path firmware fault    ok                         0       never
  Module firmware fault       ok                         0       never
  Temperature high alarm      ok                         0       never
  Temperature high warn       ok                         0       never
  Temperature low alarm       ok                         0       never
  Temperature low warn        ok                         0       never
  Voltage high alarm          ok                         0       never
  Voltage high warn           ok                         0       never
  Voltage low alarm           ok                         0       never
  Voltage low warn            ok                         0       never
  Module state                ready                      2       0:22:59 ago
  Data path 1 state           initialized               12       0:16:35 ago
  Data path 2 state           initialized               12       0:16:35 ago
  Data path 3 state           unknown                    0       never
  Data path 4 state           unknown                    0       never
  Data path 5 state           unknown                    0       never
  Data path 6 state           unknown                    0       never
  Data path 7 state           unknown                    0       never
  Data path 8 state           unknown                    0       never
Booster
  Operational speed           400Gbps
  RX LOS                      ok                         0       never
```

```
  TX fault                      ok                            0       never
  RX CDR LOL                    ok                            0       never
  TX power high alarm           ok                            0       never
  TX power high warn            ok                            0       never
  TX power low alarm            alarm                         3       0:16:37 ago
  TX power low warn             warn                          3       0:16:37 ago
  TX bias high alarm            ok                            0       never
  TX bias high warn             ok                            0       never
  TX bias low alarm             ok                            0       never
  TX bias low warn              ok                            0       never
  RX power high alarm           ok                            0       never
  RX power high warn            ok                            0       never
  RX power low alarm            ok                            2       0:16:35 ago
  RX power low warn             ok                            2       0:16:35 ago
  TX LOS
    Host lane 1                 ok                            0       never
    Host lane 2                 ok                            0       never
    Host lane 3                 ok                            0       never
    Host lane 4                 ok                            0       never
    Host lane 5                 ok                            0       never
    Host lane 6                 ok                            0       never
    Host lane 7                 ok                            0       never
    Host lane 8                 ok                            0       never
  TX CDR LOL
    Host lane 1                 ok                            0       never
    Host lane 2                 ok                            0       never
    Host lane 3                 ok                            0       never
    Host lane 4                 ok                            0       never
    Host lane 5                 ok                            0       never
    Host lane 6                 ok                            0       never
    Host lane 7                 ok                            0       never
    Host lane 8                 ok                            0       never
  TX adaptive input EQ fault
    Host lane 1                 ok                            0       never
    Host lane 2                 ok                            0       never
    Host lane 3                 ok                            0       never
    Host lane 4                 ok                            0       never
    Host lane 5                 ok                            0       never
    Host lane 6                 ok                            0       never
    Host lane 7                 ok                            0       never
    Host lane 8                 ok                            0       never
Pre-amp
  Operational speed             50Gbps
  RX LOS                        ok                            0       never
  TX fault                      ok                            0       never
  RX CDR LOL                    ok                            0       never
  TX power high alarm           ok                            0       never
  TX power high warn            ok                            0       never
  TX power low alarm            alarm                         3       0:16:37 ago
  TX power low warn             warn                          3       0:16:37 ago
  TX bias high alarm            ok                            0       never
  TX bias high warn             ok                            0       never
  TX bias low alarm             ok                            0       never
  TX bias low warn              ok                            0       never
  RX power high alarm           ok                            0       never
  RX power high warn            ok                            0       never
  RX power low alarm            ok                            2       0:16:35 ago
  RX power low warn             ok                            2       0:16:35 ago
```

Some lines of output do not apply to the OSFP-LS modules (For example, **`Operational speed`**, **`RX CDR LOL`**). These lines of output should be ignored. The fields of interest are Module state, **`Data path 1 state, and Data path 2 state`**. Under normal operating conditions, **`Module state`** will read ready

and the **Data path state** fields will read either **initialized or activated**. Initialized means that the module is ready to operate but is receiving no signal to amplify. Activated means that the amplifier is active.

15.2.5.65 shutdown (OSPFv2)

The **shutdown** command disables OSPFv2 on the switch. OSPFv2 is disabled on individual interfaces with the **shutdown (OSPFv2)** command.

The **no shutdown** and **default shutdown** commands enable the OSPFv2 instance by removing the **shutdown** statement from the OSPF block in ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- This command disables OSPFv2 activity on the switch.

```
switch(config)# router ospf 6
switch(config-router-ospf)# shutdown
switch(config-router-ospf)#
```

- This command resumes OSPFv2 activity on the switch.

```
switch(config-router-ospf)# no shutdown
switch(config-router-ospf)#
```

15.2.5.66 summary-address

The **summary-address** command allows aggregation of external routes advertised by an OSPF ASBR. It is used to aggregate AS External and NSSA External LSAs.

The **default summary-address** and **no summary-address** commands delete the current **summary-address** configurations.

Command Mode

Router Configuration Mode

Command Syntax

summary-address {***ip_address subnet_mask*** | ***ip_prefix***} [***attribute_map WORD*** | ***not_advertise*** | tag]

default summary-address {***ip_address summary_mask*** | ***ip_prefix***}

no summary-address {***ip_address summary_mask*** | ***ip_prefix***}

Parameters

- ***ip_address subnet_mask*** IPv4 subnet in dotted decimal notation.
- ***ip_prefix*** IPv4 subnet in CIDR notation.
- ***attribute_map WORD*** allows using a route-map to set the attributes to be advertised in the LSA. Options include:
  - set metric
  - set metric-type

  - set tag
- ***not_advertise*** suppresses the advertisement of contributing external prefixes by the router.
- **tag** allows setting the tag in the advertised external LSA. The tag value ranges from ***0 to 4294967295***. The default value is ***0***.

Guidelines

This feature reduces the size of External LSDB in OSPF, does not impact inter area and intra area LSAs. This command installs a Null0 route in FIB when at least one contributor is present.

Restriction

Only OSPF redistributed routes are aggregated.

Example

This command advertises an external LSA for ***50.0.0.0/16*** prefix if at least one BGP contributing route is present which falls in the subnet ***50.0.0.0/16***.



Note: The **show** commands display aggregation of BGP prefixes ***50.0.0.0/24*** and ***50.0.1.0/24*** into one OSPF AS External LSA for ***50.0.0.0/16*** prefix. A route-map is to set metric and tag for the advertised LSA.

```
switch(config)# router ospf 5
switch(config-router-ospf)# redistribute bgp
switch(config-router-ospf)# summary-address 50.0.0.0/16 attribute-map
 BGP_AGGR
switch(config-router-ospf)# exit
switch(config)# show ip route bgp

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

 B E    50.0.0.0/24 [200/0] via 3.0.0.12, Ethernet3
 B E    50.0.1.0/24 [200/0] via 3.0.0.12, Ethernet3
switch(config)# show running-config
...
route-map BGP_AGGR permit 10
   set metric 42
   set tag 19
...
router ospf 1
   router-id 1.0.0.10
   redistribute bgp
   max-lsa 12000
   summary-address 50.0.0.0/16 attribute-map BGP_AGGR

switch(config)# show ip ospf database external

            OSPF Router with ID(1.0.0.10) (Process ID 1) (VRF default)


                 Type-5 AS External Link States

  LS Age: 9
  Options: (E DC)
```

```
  LS Type: AS External Links
  Link State ID: 50.0.0.0
  Advertising Router: 1.0.0.10
  LS Seq Number: 0x80000001
  Checksum: 0x2c0c
  Length: 36
  Network Mask: 255.255.0.0
        Metric Type: 2
        Metric: 42
        Forwarding Address: 0.0.0.0
        External Route Tag: 19

switch(config)# show ip route aggregate

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

 A O    50.0.0.0/16 is directly connected, Null0
```

15.2.5.67 system profile

Use the **system profile** command to apply the user defined TCAM profile to the system.

Command Mode

hardware tcam mode

Command Syntax

system profile ***profilename***

Parameter

profilename Name of the selected system profile.

Example

```
(config-hw-tcam)# system profile profilename
```

15.2.5.68 timers lsa rx min interval (OSPFv2)

The **timers lsa rx min interval** command sets the minimum interval for acceptance of identical Link State Advertisements (LSAs) from OSPFv2 neighbors.

The **no timers lsa rx min interval** and **default timers lsa rx min interval** commands restore the minimum interval to the default of ***1*** second by removing the **timers lsa rx min interval** command from the ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

timers lsa rx min interval ***lsa_time***

no timers lsa rx min interval

default timers lsa rx min interval

Parameter

lsa_time Minimum time (in milliseconds) after which the switch will accept an identical LSA from OSPFv2 neighbors. Default is ***1000*** (***1*** second).

Example

This command sets the minimum LSA arrival interval to ***10*** milliseconds.

```
switch(config)# router ospf 6
switch(config-router-ospf)# timers lsa rx min interval 10
switch(config-router-ospf)#
```

15.2.5.69 timers lsa tx delay initial (OSPFv2)

The **timers lsa tx delay initial** command sets the rate-limiting values for OSPF link-state advertisement generation.

The **no timers lsa tx delay initial** and **default imers throttle lsa all** commands restore the defaults by removing the **timers lsa tx delay initial** command from the ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

timers lsa tx delay initial [***initial_delay*** | ***min_hold*** | ***max_wait***]

no timers lsa tx delay initial

default timers lsa tx delay initial

Parameters

- ***initial_delay*** Value ranges from ***0 to 600000*** (ms). Default is ***1000***.
- ***min_hold*** Value ranges from ***0 to 600000*** (ms). Default is ***5000***.
- ***max_wait*** Value ranges from ***0 to 600000*** (ms). Default is ***5000***.

Example

This command sets the rate-limiting values for OSPF link-state advertisemen to ***10*** milliseconds.

```
switch(config)# router ospf 6
switch(config-router-ospf)# timers lsa tx delay initial 10
switch(config-router-ospf)#
```

15.2.5.70 timers spf delay initial (OSPFv2)

The purpose of SPF throttling is to delay Shortest Path First (SPF) calculations when network topology is changing rapidly. The **timers spf delay initial** command controls the intervals at which the switch will perform SPF calculations. The command sets three values:

- Initial delay: how long the switch waits to perform an SPF calculation after a topology change in a network that has been stable throughout the hold interval. Because a topology change often causes several link state updates to be sent, the initial delay is configured to allow the network to settle before the switch performs an SPF calculation. If an additional topology change occurs during the initial interval, the SPF calculation still takes place after the expiration of the initial delay period and no other change is made to the throttle timers.

- Hold interval: this is an additional wait timer which scales to slow SPF calculations during periods of network instability. If a network change occurs during the hold period, an SPF calculation is scheduled to occur at the expiration of the hold interval. Subsequent hold intervals are doubled if further topology changes occur during a hold interval until either the hold interval reaches its configured maximum or no topology change occurs during the interval. If the next topology change occurs after the expiration of the hold interval, the hold interval is reset to its configured value and the SPF calculation is scheduled to take place after the initial delay.
- Maximum interval: the maximum time the switch will wait after a topology change before performing an SPF calculation.

The **no timers spf delay initial** and **default timers spf delay initial** commands restore the default OSPFv2 SPF calculation intervals by removing the **timers spf delay initial** command from the ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

timers spf delay initial [***initial_delay*** | ***hold_interval*** | ***max_interval***]

no timers spf

default timers spf

Parameters

- ***initial_delay*** Initial delay between a topology change and SPF calculation. Value ranges from ***0 to 65535000*** (ms). Default is ***0*** (ms).
- ***hold_interval*** Additional wait time after SPF calculation to allow the network to settle. If a topology change occurs during the hold interval, another SPF calculation is scheduled to occur after the hold interval expires. The next hold interval is doubled if topology changes occur during the hold interval. If doubling exceeds the maximum value, the maximum value is used instead. Value ranges from ***0 to 65535000*** (ms). Default is ***5000*** (ms).
- ***max_interval*** Maximum hold interval before the switch will perform an SPF calculation. Value ranges from ***0 to 65535000*** (ms). Default is ***5000*** (ms).

Example

These commands set the SPF timers on the switch.

```
switch(config)# router ospf 6
switch(config-router-ospf)# timers spf 5 100 20000
switch(config-router-ospf)#
```

15.2.5.71 tunnel routes

Use the **tunnel routes** command or the **default** form of the command to enable OSPFv2 routes over GRE tunnels. The tunnel routes are enabled, by default. Use the **no** form of the command to disable the tunnel routes.

Command Mode

Router OSPF configuration (config-router-ospf)

Command Syntax

tunnel routes

no tunnel routes

default tunnel routes

Examples

- To enable OSPFv2 routes over GRE tunnels.

```
switch(config)# router ospf 6
switch(config-router-ospf)# tunnel routes
switch(config-router-ospf)#
```

- To disable OSPFv2 routes over GRE tunnels.

```
switch(config)# router ospf 6
switch(config-router-ospf)# no tunnel routes
switch(config-router-ospf)#
```

- To enable the default OSPFv2 routes over GRE tunnels.

```
switch(config)# router ospf 6
switch(config-router-ospf)# default tunnel routes
switch(config-router-ospf)#
```

15.3 Open Shortest Path First – Version 3

Open Shortest Path First (OSPF) is a link-state routing protocol that operates within a single autonomous system. OSPF version 3 is defined by ***RFC 5340***.

This chapter contains the following sections.

- OSPFv3 Introduction
- OSPFv3 Conceptual Overview
- Configuring OSPFv3
- OSPFv3 Configuration Examples
- OSPFv3 Commands

15.3.1 OSPFv3 Introduction

OSPFv3 is based on OSPFv2 and includes enhancements that utilize IPv6 features. However, OSPFv3 is configured and operates independently of any implementation of OSPFv2 on the switch. OSPFv2 features that OSPFv3 implements include:

- Packet types
- Neighbor discovery and adjacency formation mechanisms
- LSA aging and flooding
- SPF calculations
- DR election procedure
- Multiple area support
- Router-ID (32 bits)

The following list describes the OSPFv3 differences and enhancements from OSPFv2:

- IPv6 128-bit addresses
- Use of link-local addresses
- OSPFv3 runs over links instead of subnets
- Support flood pacing

Arista switches support the following OSPFv3 functions:

- A single OSPFv3 instance for each VRF
- Intra- and inter-area routing

- Type 1 and 2 external routing
- Broadcast and P2P interfaces
- Stub areas
- Redistribution of static and connected routes into OSPFv3

15.3.2 OSPFv3 Conceptual Overview

This section contains the following topics:

- Storing Link States
- Topology
- Link Updates
- OSPFv3 Security
- Flood Pacing
- OSPFv3 BFD Sessions for Adjacencies in any State
- Support for OSPFv3 dn-bit-ignore

15.3.2.1 Storing Link States

OSPFv3 is a dynamic, link-state routing protocol, where links represent routable paths. Dynamic routing protocols calculate the most efficient path between locations based on bandwidth and device status.

A Link State Advertisement (LSA) is an OSPFv3 packet that communicates a router's topology to other routers. The Link State DataBase (LSDB) stores an area’s topology database and is composed of LSAs received from other routers. Routers update the LSDB by storing LSAs from other routers.

15.3.2.2 Topology

An Autonomous System (AS) is the IP domain within which a dynamic protocol controls the routing of traffic. In OSPFv3, an AS is composed of areas, which define the LSDB computation boundaries. All routers in an area store identical LSDBs. Routers in different areas exchange updates without storing the entire database, reducing information maintenance on large, dynamic networks.

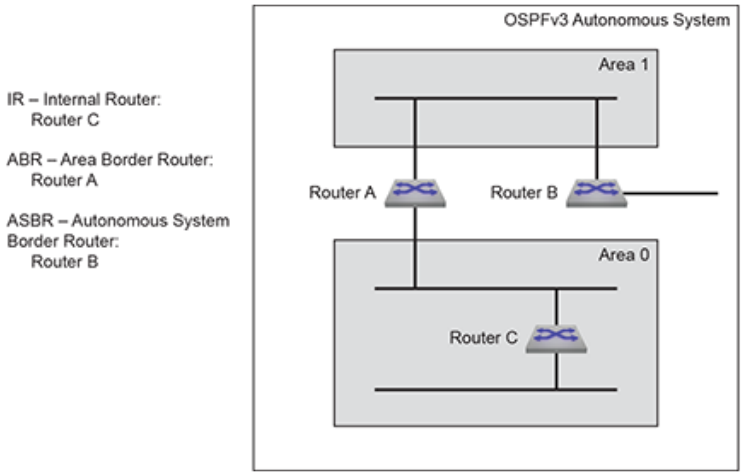
An AS shares internal routing information from its areas and external routing information from other processes to inform routers outside the AS about routes the network can access. Routers that advertise routes on other ASs commit to carry data to the IP space on the route.

OSPFv3 defines these routers:

- **Internal Router (IR):** a router whose interfaces are contained in a single area. All IRs in an area maintain identical LSDBs.
- **Area Border Router (ABR):** a router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area.
- **Autonomous System Boundary Router (ASBR):** a gateway router connecting the OSPFv3 domain to external routes, including static routes and routes from other autonomous systems.

OSPFv3 Router Types displays the OSPFv3 router types.

Figure 15-6: OSPFv3 Router Types



OSPFv3 areas are assigned a number between ***0*** and ***4,294,967,295***. Area numbers are often expressed in dotted decimal notation, similar to IP addresses.

Each AS has a backbone area, designated as area ***0***, that connects to all other areas. The backbone receives routing information from all areas, then distributes it to the other areas as required.

OSPFv3 area types include:

- Normal area accepts intra-area, inter-area, and external routes. The backbone is a normal area.
- Stub area does not receive router advertisements external to the AS. Stub area routing is based on a default route.

15.3.2.3 Link Updates

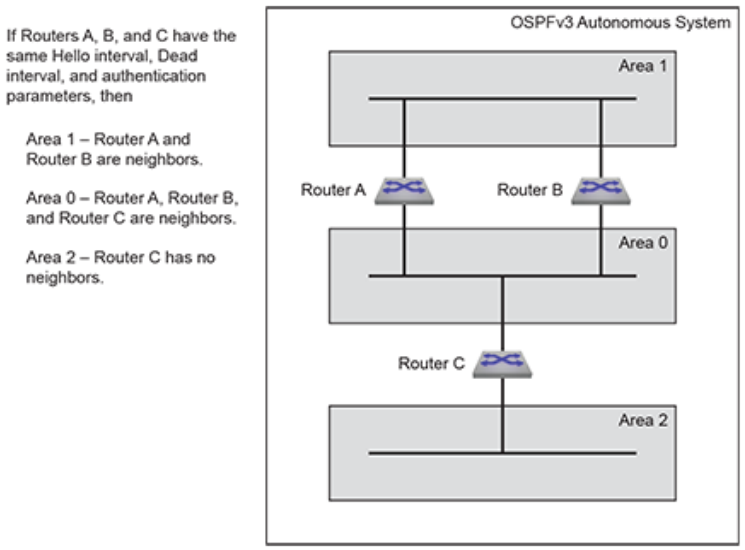
Routers periodically send hello packets to advertise status and establish neighbors. A routers hello packet includes IP addresses of other routers from which it received a hello packet within the time specified by the router dead interval. Routers become neighbors when they detect each other in their hello packets if they:

- share a common network segment.
- are in the same area.
- have the same hello interval, dead interval, and authentication parameters.

Neighbors form adjacencies to exchange LSDB information. A neighbor group uses hello packets to elect a Designated Router (DR) and Backup Designated Router (BDR). The DR and BDR become adjacent to all other neighbors, including each other. Only adjacent neighbors share database information.

OSPFv3 Neighbors illustrates OSPFv3 neighbors.

Figure 15-7: OSPFv3 Neighbors



The DR is the central contact for database exchanges. Switches send database information to their DR, which relays the information to the other neighbors. All routers in an area maintain identical LSDBs. Switches also send database information to their BDR, which stores this data without distributing it. If the DR fails, the BDR distributes LSDB information to its neighbors.

OSPFv3 routers distribute LSAs by sending them on all of their active interfaces. The router does not send hello packets from passive interfaces preventing adjacencies. The router does not process any OSPFv2 packets received on a passive interface.

When a routers LSDB is changed by an LSA, it sends the changes to the BDR and DR for distribution to the other neighbors. Routing information is updated only when the topology changes.

Routing devices use Dijkstras algorithm to calculate the shortest path to all known destinations, based on cumulative route cost. The cost of an interface indicates the transmission overhead and is usually inversely proportional to its bandwidth.

15.3.2.4 OSPFv3 Security

The OSPFv3 protocol relies on the IPsec Authentication Header (AH) and Encapsulating Security Payload (ESP) header to provide data integrity, authentication and confidentiality. Transport mode provides IPsec to OSPFv3 packets.

The IPsec SA has Security Policy Index (SPI), HMAC algorithm, and a secret key as parameters. These parameters are used to compute Integrity Check Value (ICV), that is used to authenticate peers. When authentication is enabled, all corresponding peers must use same SA parameters to clear OSPFv3 ICV verification. SA can be configured at both area and interface levels.

Note: On the same area or interface, EOS allows security configuration with either AH or ESP but not both. We can have one area or interface configured with AH and another with ESP.

15.3.2.4.1 OSPFv3 Authentication

While sending OSPFv3 packets, the HMAC-MD5 or SHA algorithm hash is inserted in the IPsec header and the packet is sent over the wire for peer authentication.

While receiving OSPFv3 packets, the computed hash is verified with the one present in the IPsec header. If it fails, OSPFv3 packets are discarded.

15.3.2.4.2 OSPFv3 Encryption

ESP provides confidentiality to OSPFv3 packets. When confidentiality is enabled, ESP encrypts the sent data and decrypts the received data. OSPFv3 packets that are not encapsulated with security payload are discarded.

OSPFv3 encryption uses algorithms of Triple Data Encryption Standard (3DES) and Advanced Encryption Standard (AES). 3DES uses a ***192*** bit key, whereas the AES key length varies by ***128***, ***192*** and ***256*** bits.

15.3.2.5 Flood Pacing

OSPFv3 flood pacing allows configuring the minimum interval between the transmission of consecutive Link State (LS) update packets in a network. Flood pacing provides the following benefits:

- Prevents the rapid drain of flood queue by sending consecutive LSU packets with a delay.
- Helps mitigate high CPU or socket buffer utilization issues that occur when a switch instantly floods a large number of LSAs.
- When LSDB is updated frequently, an incremented flood pacing interval scales down LSA flooding.



Note: A high flood pacing interval may lead to convergence delays in large OSPF LSDBs.

15.3.2.6 OSPFv3 BFD Sessions for Adjacencies in any State

- BFD sessions are only established for OSPFv3 adjacencies that are in the FULL state. In a LAN environment this results in BFD sessions not being established for OSPFv3 adjacencies with DR Other neighbors.
- This feature provides configuration that enables the establishment of BFD sessions for OSPFv3 adjacencies that are in any state. This results in the BFD sessions being established for OSPFv3 adjacencies with DR Other neighbors.

15.3.2.7 Support for OSPFv3 dn-bit-ignore

The OSPFv3 **`dn-bit-ignore`** command allows enabling or disabling the inclusion of LSAs having “Down” (DN) bit set in SPF calculations. The DN Bit is a loop prevention mechanism that implements when using OSPF as a CE - PE IGP protocol.

The DN-bit usage in OSPFv3 is explained in ***RFC6565***. With ***Release EOS-4.25.0F***, OSPFv3 honors the DN-bit in **type-3**, **type-5**, or **type-7** LSAs in non-default VRFs. LSAs are not used in SPF calculation, and are not installed in the routing table. Using the **`dn-bit-ignore`** command changes this behavior. It is recommended to understand the entire topology before configuring the **`dn-bit-ignore`** command as it may lead to forwarding loops.

15.3.3 Configuring OSPFv3

These sections describe basic OSPFv3 configuration steps:

- Configuring an OSPFv3 Instance
- Configuring OSPFv3 Areas
- Configuring Interfaces for OSPFv3
- Enabling OSPFv3
- Configuring OSPFv3 Security
- Configuring OSPFv3 Flood Pacing
- Configurating OSPv3 dn-bit-ignore
- Displaying OSPFv3 Status

15.3.3.1 Configuring an OSPFv3 Instance

15.3.3.1.1 Entering OSPFv3 Configuration Mode

OSPFv3 configuration commands apply to the specified OSPFv3 instance. To perform OSPFv3 configuration commands, the switch must be in router-OSPFv3 configuration mode. The **ipv6 router ospf** command places the switch in router-OSPFv3 configuration mode, creating an OSPFv3 instance if OSPFv3 was not previously instantiated on the switch. If no VRF is specified, the OSPFv3 instance is in the default VRF. To instantiate or configure OSPFv3 on a non-default VRF, specify that VRF when using the **ipv6 router ospf** command.

The process ID identifies the OSPFv3 instance and is local to the router. Neighbor OSPFv3 routers can have different process IDs. OSPFv3 instances configured in different VRFs on the switch must have different process IDs.

The switch supports one OSPFv3 instance for each VRF. When an OSPFv3 instance already exists, the **ipv6 router ospf** command must specify its process ID (and VRF, if it is not configured in the default VRF). Attempts to define additional instances in the same VRF will generate errors. The **show ipv6 ospf** command displays information about OSPFv3 instances, including their process IDs.

Example

This command places the switch in router-OSPFv3 configuration mode for the default VRF. If OSPFv3 was not previously instantiated in the default VRF, the command creates an OSPFv3 instance in the default VRF with a process ID of ***9***.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

15.3.3.1.2 Defining the Router ID

The router ID is a 32-bit number assigned to a router running OSPFv3. This number uniquely labels the router within an Autonomous System. Status commands identify the switch through the router ID. When configuring OSPFv3 instances in multiple VRFs, each should have a different router ID.

Note: A router ID is required to run OSPF, and Arista recommends setting the router ID manually for the following reasons:

1. If there are no IPv4 addresses configured on the switch, a manually configured router ID is required.
2. The router ID also does not change if the interface status or IP address changes, which can cause confusion if the ID has been selected automatically.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IPv6 address, if a loopback interface is active on the switch.
3. The highest IPv6 address on the router.

Note: When configuring VXLAN on an MLAG, always manually configure the OSPFv3 router ID to prevent the switch from using the common VTEP IP address as the router ID.

The **router-id (OSPFv3)** command configures the router ID for an OSPFv3 instance.

Example

This command assigns ***15.1.1.1*** as the OSPFv3 router ID.

```
switch(config-router-ospf3)# router-id 15.21.4.9
switch(config-router-ospf3)# show active
```

```
ipv6 router ospf 9
   router-id 15.21.4.9
switch(config-router-ospf3)#
```

15.3.3.1.3 Global OSPFv3 Parameters

These router-OSPFv3 configuration mode commands define OSPFv3 behavior for the OSPFv3 instance under which they are used.

Logging Adjacency Changes

The **log-adjacency-changes (OSPFv3)** command configures the switch to log OSPFv3 link-state changes and transitions of OSPFv3 neighbors into the up or down state.

Examples

- This command configures the switch to log transitions of OSPFv3 neighbors into the up or down state.

```
switch(config-router-ospf3)# log-adjacency-changes
switch(config-router-ospf3)#
```

- This command configures the switch to log all OSPFv3 link-state changes.

```
switch(config-router-ospf3)# log-adjacency-changes detail
switch(config-router-ospf3)#
```

Intra-Area Distance

The **distance ospf intra-area (OSPFv3)** command configures the administrative distance for routes contained in a single OSPFv3 area. Administrative distances compare dynamic routes configured by different protocols. The default administrative distance for intra-area routes is ***10***.

Example

This command configures an administrative distance of ***90*** for OSPFv3 intra-area routes.

```
switch(config-router-ospf3)# distance ospf intra-area 90
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   distance ospf intra-area 90
switch(config-router-ospf3)#
```

Passive Interfaces

The **passive-interface (OSPFv3)** command prevents the transmission of hello packets on the specified interface. Passive interfaces drop all adjacencies and do not form new adjacencies. Although passive interfaces do not send or receive LSAs, other interfaces may generate LSAs for the network segment. The router does not send OSPFv3 packets from a passive interface or process OSPFv3 packets received on a passive interface. The router advertises the passive interface in the router LSA.

The **no passive-interface** command re-enables OSPFv3 processing on the specified interface.

Examples

- This command configures ***vlan 200*** as a passive interface.

```
switch(config-router-ospf3)# passive-interface vlan 200
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   passive-interface Vlan200
```

```
switch(config-router-ospf3)#
```

- This command configures ***vlan 200*** as an active interface.

```
switch(config-router-ospf3)# no passive-interface vlan 200
switch(config-router-ospf3)# show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

Redistributing Connected Routes

Redistributing connected routes causes the OSPFv3 instance to advertise all connected routes on the switch as external OSPFv3 routes. Connected routes are routes that are established when IPv6 is enabled on an interface.

Example

The **redistribute (OSPFv3)** **connected** command converts connected routes to OSPFv3 external routes.

```
switch(config-router-ospf3)# redistribute connected
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   redistribute connected
switch(config-router-ospf3)#
```

Redistributing Static Routes

Redistributing static routes causes the OSPFv3 instance to advertise all static routes on the switch as external OSPFv3 routes. The switch does not support redistributing individual static routes.

Example

The **redistribute (OSPFv3)** **static** command converts static routes to OSPFv3 external routes.

```
switch(config-router-ospf3)# redistribute static
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   redistribute static
switch(config-router-ospf3)#
```

15.3.3.2 Configuring OSPFv3 Areas

OSPFv3 areas are configured through area commands. The switch must be in router-OSPFv3 configuration mode, as described in Entering OSPFv3 Configuration Mode, to run area commands.

Areas are assigned a 32-bit number that is expressed in decimal or dotted-decimal notation. When an OSPFv3 instance configuration contains multiple areas, the switch only configures areas associated with its interfaces.

15.3.3.2.1 Configuring the Area Type

The **no area (OSPFv3)** command specifies the area type. The switch supports three area types:

- **Normal area**: Area that accepts intra-area, inter-area, and external routes. The backbone area (area ***0***) is a normal area.
- **Stub area**: Area where external routes are not advertised. External routes are reached through a default summary route (***0.0.0.0***) inserted into stub areas. Networks with no external routes do not require stub areas.

The default area type is normal.

Example

These commands configures area ***200*** as a NSSA area and ***300*** as a stub area.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# area 200 nssa
switch(config-router-ospf3)# area 300 stub
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 0.0.0.200
   area 0.0.1.44 stub
switch(config-router-ospf3)#
```

15.3.3.2.2 Configuring Area Parameters

These router-OSPFv3 configuration mode commands define OSPFv3 behavior in a specified area.

Default Summary Route Cost

The **area default-cost (OSPFv3)** command specifies the cost of the default summary route that ABRs send into a stub area or NSSA. Summary routes, also called inter-area routes, originate in areas different than their destination. When the **area default-cost** command is not configured for an area, the default-cost of that area is set to ***10***.

Example

This command configures a cost of ***25*** for the default summary route in area ***0.0.1.194 (450)***.

```
switch(config-router-ospf3)# area 450 default-cost 25
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 0.0.1.194 default-cost 25
```

Area Stub

The **area stub (OSPFv3)** command configures the area type of an OSPFv3 area. All routers in an AS must specify the same area type for identically numbered areas.

Stub areas are areas in which external routes are not advertised. To reach these external routes, the stub area uses a default summary route (***0.0.0.0***). Networks without external routes do not require stub areas.

Areas are **normal** by default; area type configuration is required only for stub NSSA areas. Area ***0*** is always a normal area and cannot be configured through this command.

Examples

- This command configures area ***45*** as a stub area.

```
switch(config)# ipv6 router ospf 3
switch(config-router-ospf3)# area 45 stub
switch(config-router-ospf3)#
```

- This command configures area ***10.92.148.17*** as a stub area.

```
switch(config-router-ospf3)# area 10.92.148.17 stub
switch(config-router-ospf3)#
```

Area Range

The **area range (OSPFv3)** command is used by OSPFv3 Area Border Routers (ABRs) to consolidate or summarize routes, to configure a cost setting for those routes, and to suppress summary route advertisements.

By default, an ABR creates a summary LSA for each route in an area and advertises that LSA to adjacent areas. The **area range (OSPFv3)** command aggregates routing information on area boundaries, allowing the ABR to use one summary LSA to advertise multiple routes.

Examples

- These commands consolidate and summarize routes at an area boundary ***1***.

```
switch(config)# ipv6 router ospf 1
switch(config-router-ospf3)# area 1 range 2001:0DB8:0:1::/64
switch(config-router-ospf3)#
```

- These commands change the address range status to DoNotAdvertise. Neither one of the individual intra-area routes falling under range or the ranged prefix is advertised as summary LSA.

```
switch(config)# ipv6 router ospf 1
switch(config-router-ospf3)# area 1 range 2001:0DB8:0:1::/64 not-advertise
switch(config-router-ospf3)#
```

15.3.3.3 Configuring Interfaces for OSPFv3

OSPFv3 interface configuration commands enable OSPFv3 on an interface, assign the interface to an area, and specify transmission parameters for routed ports and SVIs that handle OSPFv3 packets.

15.3.3.3.1 Assigning an Interface to an Area

The **ipv6 ospf area** command enables OSPFv3 on the configuration mode interface and associates the specified area to the interface. Each routed interface can be associated with one OSPFv3 area; subsequent **ipv6 ospf area** commands that designate a different area on an interface replace any existing command for the interface.

Example

These commands enable OSPFv3 instance ***9*** on ***interface vlan 200*** and associate area ***0*** to the interface.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 ospf 9 area 0
switch(config-if-Vl200)# show active
interface Vlan200
   ipv6 ospf 9 area 0.0.0.0
switch(config-if-Vl200)#
```

15.3.3.3.2 Configuring Intervals

Interval configuration commands determine OSPFv3 packet transmission characteristics for a specified VLAN interface. Interval configuration commands are entered in vlan-interface configuration mode.

Hello Interval

The hello interval specifies the period between consecutive hello packet transmissions from an interface. Each OSPFv3 neighbor should specify the same hello interval, which should not be longer than any neighbors dead interval.

The **ospfv3 hello-interval** command configures the hello interval for the configuration mode interface. The default is ***10*** seconds.

Example

These commands configure a hello interval of ***45*** seconds for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 hello-interval 45
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 hello-interval 45
switch(config-if-Vl200)#
```

Dead Interval

The dead interval specifies the period that an interface waits for an OSPFv3 packet from a neighbor before it disables the adjacency under the assumption that the neighbor is down. The dead interval should be configured identically on all OSPFv3 neighbors and be longer than the hello interval of any neighbor.

The **`ospfv3 dead-interval`** command configures the dead interval for the configuration mode interface. The default is ***40*** seconds.

Example

This command configures a dead interval of ***75*** seconds for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 dead-interval 75
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 dead-interval 75
switch(config-if-Vl200)#
```

Retransmission Interval

Routers that send OSPFv3 advertisements to an adjacent router expect to receive an acknowledgment from that neighbor. Routers that do not receive an acknowledgment will retransmit the advertisement. The retransmission interval specifies the period between retransmissions.

The **`ospfv3 ipv6 retransmit-interval`** command configures the LSA retransmission interval for the configuration mode interface. The default retransmission interval is ***5*** seconds.

Example

This command configures a retransmission interval of ***25*** seconds for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 ipv6 retransmit-interval 25
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 ipv6 retransmit-interval 25
switch(config-if-Vl200)#
```

Transmission Delay

The transmission delay is an estimate of the time that an interface requires to transmit a link-state update packet. OSPFv3 adds this delay to the age of outbound packets to more accurately reflect the age of the LSA when received by a neighbor.

The **`ospfv3 transmit-delay`** command configures the transmission delay for the configuration mode interface. The default transmission delay is one second.

Example

his command configures a transmission delay of ***10*** seconds for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 transmit-delay 10
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 transmit-delay 10
switch(config-if-Vl200)#
```

15.3.3.3.3 Configuring Interface Parameters

Interface Cost

The OSPFv3 interface cost reflects the overhead of sending packets across the interface. The cost is typically assigned to be inversely proportional to the bandwidth of the interface. The **`ospfv3 cost`** command configures the OSPFv3 cost for the configuration mode interface. The default cost is ***10***.

Example

This command configures a cost of ***50*** for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 cost 50
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 cost 50
switch(config-if-Vl200)#
```

Router Priority

Router priority determines preference during Designated Router (DR) and Backup Designated Router (BDR) elections. Routers with higher priority numbers have preference over other routers. Routers with a priority of zero cannot be elected as a DR or BDR.

The **`ospfv3 priority`** command configures router priority for the configuration mode interface. The default priority is ***1***.

Example

This command configures a router priority of ***128*** for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 priority 128
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 priority 128
switch(config-if-Vl200)#
```

15.3.3.4 Enabling OSPFv3

15.3.3.4.1 IP Routing

OSPFv3 requires that IPv6 unicast routing is enabled on the switch. When IP routing is not enabled, entering OSPFv3 configuration mode generates a message.

Examples

- This message is displayed if, when entering router-OSPFv3 configuration mode, IPv6 unicast routing is not enabled.

```
switch(config)# ipv6 router ospf 9
! IPv6 routing not enabled
switch(config-router-ospf3)#
```

- This command enables IP routing on the switch.

```
switch(config)# ipv6 unicast-routing
```

15.3.3.4.2 Disabling OSPFv3

The **shutdown (OSPFv3)** disables OSPFv3 operations on the switch without disrupting the OSPFv3 configuration. To disable OSPFv3 on an interface, remove the **ipv6 ospf area** statement for the corresponding interface.

The **no shutdown** command resumes OSPFv3 activity.

Examples

- This command disables OSPFv3 activity on the switch.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# shutdown
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   shutdown
switch(config-router-ospf3)#
```

- This command resumes OSPFv3 activity on the switch.

```
switch(config-router-ospf3)# no shutdown
switch(config-router-ospf3)# show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

15.3.3.5 Configuring OSPFv3 Security

You can configure OSPFv3 security for either an area or an interface, or both, using either an Authentication Header (AH) or an Encapsulating Security Payload (ESP).

When OSPFv3 security is configured on an area, the configured settings apply to all interfaces in that area. Interface-specific configuration overrides configuration on the area to which the interface belongs.

15.3.3.5.1 Configuring OSPFv3 Authentication

Configuring OSPFv3 Authentication for Areas

The **area authentication ipsec spi** command configures OSPFv3 authentication on an area.

Example

This command configures OSPFv3 authentication on an area with MD5 hash algorithm.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# area 0.0.0.0 authentication ipsec spi 34 md5 0
 8FD6158BFE81ADD961241D8E4169D411
switch(config-router-ospf3)# show active
ipv6 router ospf 9
```

```
   area 0.0.0.0 authentication ipsec spi 34 md5 7 $1$cNpcrQl1czqdvKAzKLtYVr6
I7+R3niuWouDKKYCFNs4/XOWG/Iap5Q==
switch(config-router-ospf3)#
```

Configuring OSPFv3 Authentication for Interfaces

The **ospfv3 authentication ipsec spi** command configures OSPFv3 authentication on an interface.

Example

This command configures OSPFv3 authentication on an interface with MD5 hash algorithm.

```
switch(config-if-Et9)# ospfv3 authentication ipsec spi 3456 md5 0 8FD6158BFE81A
DD961241D8E4169D411
switch(config-if-Et9)# show active
interface Ethernet9
   no switchport
   ospfv3 authentication ipsec spi 3456 md5 7 $1$xtmcMSPzEn+Njp8Lb4qryVV
OjKcjsrYuv6dx1O+nSwKQdaiRt2RPTQ==
switch(config-if-Et9)#
```

15.3.3.5.2 Configuring OSPFv3 Encryption

Configuring OSPFv3 Encryption for Areas

The **area encryption ipsec spi** command configures OSPFv3 security on an area.

Example

This command configures OSPFv3 security on an area with 3DES-CBC encryption and MD5 hash algorithm.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# area 0.0.0.0 encryption ipsec spi 5678 esp 3des-
cbc md5 passphrase 0 8FD6158BFE81ADD961241D8E4169D411
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 0.0.0.0 encryption ipsec spi 5678 esp 3des-cbc md5 passphrase 7
$1$cNpcrQl1czqdvKAzKLtYVr6I7+R3niuWouDKKYCFNs4/XOWG/Iap5Q==
switch (config-router-ospf3)#
```

Configuring OSPFv3 Encryption for Interfaces

The **ospfv3 encryption ipsec spi** command configures OSPFv3 security on an interface.

Example

This command configures OSPFv3 security on an interface with 3DES-CBC encryption and SHA1 algorithm.

```
switch(config)# interface ethernet 9
switch(config-if-Et9)# ospfv3 encryption ipsec spi 345 esp 3des-cbc sha1
 passphrase 0 2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
switch(config-if-Et9)# show active
interface Ethernet9
   no switchport
   ospfv3 encryption ipsec spi 345 esp 3des-cbc sha1 passphrase 7
$1$VmUkWk6IL2S343bR3BbH0RhgvxHhwBpfvB4VXKNOOQF7HJBp5VvXTfBaVYbgCkWU
switch(config-if-Et9)#
```

15.3.3.6 Configuring OSPFv3 Flood Pacing

Flood pacing can be configured for global OSPFv3 instances and address families. The **timers pacing flood** command configures OSPFv3 flood pacing.

Examples

- This command configures OSPFv3 flood pacing timer to ***50*** ms in the global OSPFv3 instance.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# timers pacing flood 50
switch(config-router-ospf3)# show ipv6 ospf
Routing Process "ospfv3 9" with ID 13.13.13.13 and Instance 0 VRF default
  FIPS mode disabled
  It is not an autonomous system boundary router and is not an area border
 router
  Minimum LSA arrival interval 1000 msecs
  Initial LSA throttle delay 1000 msecs
  Minimum hold time for LSA throttle 5000 msecs
  Maximum wait time for LSA throttle 5000 msecs
  Interface flood pacing timer 50 msecs
  It has 0 fully adjacent neighbors
  Number of areas in this router is 1. 1 normal, 0 stub, 0 nssa
  Number of LSAs 1
  Initial SPF schedule delay 0 msecs
  Minimum hold time between two consecutive SPFs 5000 msecs
  Current hold time between two consecutive SPFs 5000 msecs
  Maximum wait time between two consecutive SPFs 5000 msecs
  SPF algorithm last executed 21d19h ago
  No scheduled SPF
  Adjacency exchange-start threshold is 20
  Maximum number of next-hops supported in ECMP is 32
  Number of backbone neighbors is 0
  Graceful-restart is not configured
  Graceful-restart-helper mode is enabled
  Area 0.0.0.0
    Number of interface in this area is 0
    It is a normal area
    SPF algorithm executed 2 times
```

- This command configures OSPFv3 flood pacing timer to ***50*** ms for ipv4 address family.

```
switch(config)# router ospfv3
switch(config-router-ospfv3)# address-family ipv4
switch(config-router-ospfv3-af)# timers pacing flood 50
switch(config-router-ospfv3-af)# show ospfv3
OSPFv3 address-family ipv4
Routing Process "ospfv3" with ID 11.1.11.1 and Instance 64 VRF default
  FIPS mode disabled
  It is not an autonomous system boundary router and is not an area border
 router
  Minimum LSA arrival interval 1000 msecs
  Initial LSA throttle delay 1000 msecs
  Minimum hold time for LSA throttle 5000 msecs
  Maximum wait time for LSA throttle 5000 msecs
  Interface flood pacing timer 50 msecs
  It has 0 fully adjacent neighbors
  Number of areas in this router is 1. 1 normal, 0 stub, 0 nssa
  Number of LSAs 1
  Initial SPF schedule delay 0 msecs
  Minimum hold time between two consecutive SPFs 5000 msecs
  Current hold time between two consecutive SPFs 5000 msecs
  Maximum wait time between two consecutive SPFs 5000 msecs
  SPF algorithm last executed 00:01:05 ago
  No scheduled SPF
  Adjacency exchange-start threshold is 20
  Maximum number of next-hops supported in ECMP is 32
  Number of backbone neighbors is 0
```

```
   Graceful-restart is not configured
   Graceful-restart-helper mode is enabled
   Area 0.0.0.0
     Number of interface in this area is 0
     It is a normal area
     SPF algorithm executed 2 times
```

15.3.3.7 Configurating OSPv3 dn-bit-ignore

OSPFv3

Use the command **dn-bit-ignore** to include type-3/5/7 LSAs having their DN-bit set in the SPF calculation.

Use the commands **dn-bit-ignore** or **default dn-bit-ignore** to revert the behavior back to default. This command is available in ***ipv6 router ospf vrf*** configuration mode and the ***router ospfv3 vrf*** configuration mode. Note that this command is not available in the default VRF, and that both configuration styles are captured below.

router ospfv3 Configuration Style

The **dn-bit-ignore** command is available under the ***router ospfv3 vrf*** configuration mode. This disables the dn-bit check for Type-3/5/7 LSAs in non-default VRFs.

```
switch(config)# router ospfv3 vrf red
switch(config-router-ospfv3-vrf-red)# dn-bit-ignore
```

ipv6 router ospf Configuration Style

The **dn-bit-ignore** command is also available under the ***ipv6 router ospf vrf*** configuration mode. This disables the dn-bit check for Type-3/5/7 LSAs in non-default VRFs.

```
switch(config)# ipv6 router ospf 1 vrf red
switch(config-router-ospfv3-vrf-red)# dn-bit-ignore
```

15.3.3.8 Displaying OSPFv3 Status

This section describes OSPFv3 **show** commands that display OSPFv3 status. General switch methods that provide OSPFv3 information include pinging routes, viewing route status (**show ip route** command), and viewing the configuration (**show running-config** command).

15.3.3.8.1 OSPFv3 Summary

The **show ipv6 ospf** command displays general OSPFv3 configuration information, operational statistics and status for the OSPFv3 instance, followed by a brief description of the areas configured on the switch.

Example

This command displays OSPFv3 routing process information.

```
switch(config-router-ospf3)#show ipv6 ospf
Routing Process "ospfv3 1" with ID 1.1.1.1 and Instance 0 VRF default
  It is not an autonomous system boundary router and is not an area border
 router
  Minimum LSA arrival interval 1000 msecs
  Initial LSA throttle delay 1000 msecs
  Minimum hold time for LSA throttle 5000 msecs
  Maximum wait time for LSA throttle 5000 msecs
  Interface flood pacing timer 50 msecs
  It has 0 fully adjacent neighbors
```

```
  ...
  Graceful-restart is not configured
  Graceful-restart-helper mode is enabled
```

15.3.3.8.2 Viewing OSPFv3 on the Interfaces

The **show ipv6 ospf interface** command displays OSPFv3 information for switch interfaces configured for OSPFv3. Different command options allow the display of either all interfaces or a specified interface. The command can also be configured to display complete information or a brief summary.

Example

This command displays OSPFv3 information for interfaces where OSPFv3 is enabled.

```
switch#show ipv6 ospf interface
Ethernet17 is up
  Interface Address fe80::48c:73ff:fe00:1319%Ethernet12, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State Backup DR, Priority 1
  Designated Router is 10.37.0.37
  Backup Designated Router is 10.37.0.23
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
Vlan31 is up
  Interface Address fe80::48c:73ff:fe00:1319%Vlan31, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State Backup DR, Priority 1
  Designated Router is 10.37.0.22
  Backup Designated Router is 10.37.0.23
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
Vlan32 is up
  Interface Address fe80::48c:73ff:fe00:1319%Vlan32, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State DR Other, Priority 1
  Designated Router is 10.37.0.11
  Backup Designated Router is 10.37.0.22
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 2 switch#
```

15.3.3.8.3 Viewing the OSPFv3 Database

The **show ipv6 ospf database <link state list>** command displays the LSAs in the LSDB for the specified area. If no area is listed, the command displays the contents of the database for each area on the switch. The database command provides options to display subsets of the LSDB database, a summary of database contents, and the link states that comprise the database.

Example

This command displays the OSPFv3 database of Link State Advertisements (LSAs).

```
switch#show ipv6 ospf database
Routing Process "ospf 9":

  AS Scope LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 AEX        0.0.0.5    10.37.0.37   15 0x80000005   0x00be82
 AEX        0.0.0.9    10.37.0.22 1747 0x8000002b   0x00df56
 AEX        0.0.0.3    10.37.0.46  599 0x8000002d   0x00651d

Area 0.0.0.0 LSDB
```

```
Type        Link ID     ADV Router  Age       Seq#   Checksum
 RTR        0.0.0.0    10.37.0.32  234 0x80000031   0x00585a
 NTW       0.0.0.26    10.37.0.32  271 0x80000005   0x005609
 NAP       0.0.0.26    10.37.0.32  274 0x80000005   0x00964c

Interface vlan3911 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.38    10.37.0.22  267 0x80000005   0x00a45a
 LNK       0.0.0.23    10.37.0.23  270 0x8000002c   0x005b7e

  Interface vlan3902 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.17    10.37.0.11 1535 0x8000002b   0x007120
 LNK       0.0.0.37    10.37.0.22    7 0x8000002b   0x00ce23
 LNK       0.0.0.22    10.37.0.23  250 0x8000002d   0x00c350 switch#
```

15.3.3.8.4 Viewing OSPFv3 Neighbors

The **show ipv6 ospf neighbor** command displays information about the routers that are neighbors to the switch. Command options allow the display of summary or detailed information about the neighbors to all areas and interfaces on the switch. The command also allows for the display of neighbors to individual interfaces or areas. The **adjacency-changes** option displays the interfaces adjacency changes.

Example

This command displays the switch's neighbors.

```
switch# show ipv6 ospf neighbor
Routing Process "ospf 9":
Neighbor 10.37.0.37 priority is 1, state is Full
  In area 0.0.0.0 interface et12
  DR is 10.37.0.37 BDR is 10.37.0.23
  Options is 0
  Dead timer is due in 37 seconds
Neighbor 10.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3911
  DR is 10.37.0.22 BDR is 10.37.0.23
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 10.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3902
  DR is 10.37.0.11 BDR is 10.37.0.22
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 10.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3908
  DR is 10.37.0.22 BDR is 10.37.0.21
  Options is 0
  Dead timer is due in 39 seconds switch#
```

15.3.3.8.5 Viewing OSPFv3 Routes

The **show ipv6 routes** command provides an OSPFv3 option.

Example

This command displays the switch's OSPFv3 routes.

```
switch# show ipv6 route ospf
```

```
IPv6 Routing Table - 43 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:3279:81a4:1112::/64 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1114::/64 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1124::/64 [10/20]
     via fe80::21c:41ff:fe01:5fe1, Vlan3901
     via fe80::21c:41ff:fe01:5fe1, Vlan3902
     via fe80::21c:41ff:fe01:5fe1, Vlan3908
O   fd7a:3279:81a4:1a00::25/128 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1a00::28/128 [150/11]
     via fd7a:3279:81a4:fe40::5, Vlan3908
```

15.3.3.8.6 Viewing OSPFv3 dn-bit-ignore

Use the **`show running-config`** command to verify whether the **`dn-bit-ignore`** command is configured.

15.3.4 OSPFv3 Configuration Examples

This section describes the commands required to configure three OSPFv3 topologies.

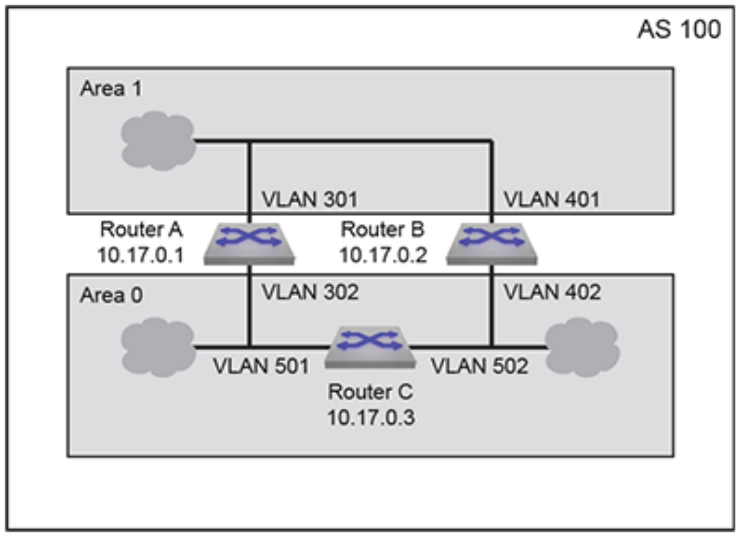
15.3.4.1 OSPFv3 Configuration Example 1

The AS in Example 1 contains two areas that are connected through two routers. The backbone area also contains an internal router that connects two links.

15.3.4.1.1 Example 1 Topology

OSPFv3 Example 1 displays the Example 1 topology. Two ABRs connect area ***0*** and area ***1*** ***Router A*** and ***Router B***. ***Router C*** is an internal router that connects two links in area ***0***. Area ***0*** is normal; area ***1*** is stub.

Figure 15-8: OSPFv3 Example 1



Area 1 Configuration

Area ***1*** contains links to ABRs ***Router A*** and ***Router B***.

- ***Router A*** is accessed through ***VLAN 301***.

- ***Router B*** is accessed through ***VLAN 401***.
- Designated Router (DR): ***Router A***.
- Backup Designated Router (BDR): ***Router B***.
- Each router defines an interface cost of ***10***.
- Router priority is not specified for either router on area ***1***.

Area 0 ABR Configuration

Area 0 contains links to ABRs ***Router A*** and ***Router B***.

- ***Router A*** is accessed through ***VLAN 302***.
- ***Router B*** is accessed through ***VLAN 402***.
- Designated Router (DR): ***Router B***.
- Backup Designated Router (BDR): ***Router A***.
- Each router defines an interface cost of ***20***.
- Each router defines a retransmit-interval of ***10***.
- Each router defines a transmit-delay of **2**.
- Router priority is specified such that ***Router B*** will be elected as the Designated Router.

Area 0 IR Configuration

Area ***0*** contains two links to an internal router.

- ***Router C*** is accessed through ***VLAN 501*** and ***VLAN 502***.
- ***VLAN 501*** is configured as follows:
  - Interface cost of ***20***.
  - Retransmit-interval of ***10***.
  - Transmit-delay of ***2***.
- ***VLAN 502*** is configured as follows:
  - Interface cost of ***20***.
  - Dead interval of ***80*** seconds.

15.3.4.1.2 Example 1 Code

This code configures the OSPFv3 instances on the three switches.

1. Configure the areas and router IDs.
   a. Router A OSPFv3 instance configuration:

```
switch-A(config)# ipv6 router ospf 100
switch-A(config-router-ospfv3)# area 1 stub
switch-A(config-router-ospfv3)# router-id 10.17.0.1
```

   b. Router B OSPFv3 instance configuration:

```
switch-B(config)# ipv6 router ospf 100
switch-B(config-router-ospfv3)# area 1 stub
switch-B(config-router-ospfv3)# router-id 10.17.0.2
```

   c. Router C OSPFv3 instance configuration: interfaces:

```
switch-C(config)# ipv6 router ospf 100
switch-C(config-router-ospfv3)# router-id 10.17.0.3
```

2. Configure the interface OSPFv3 area and transmission parameters.

a. ***Router A*** interfaces:

```
switch-A(config)# interface vlan 301
switch-A(config-if-Vl301)# ipv6 ospf 100 area 1
switch-A(config-if-Vl301)# ospfv3 cost 10
switch-A(config-if-Vl301)# ospfv3 priority 6
switch-A(config-if-Vl301)# exit
switch-A(config)# interface vlan 302
switch-A(config-if-Vl302)# ipv6 ospf 100 area 0
switch-A(config-if-Vl302)# ospfv3 cost 20
switch-A(config-if-Vl302)# ospfv3 ipv6 retransmit-interval 10
switch-A(config-if-Vl302)# ospfv3 transmit-delay 2
switch-A(config-if-Vl302)# ospfv3 priority 4
```

b. ***Router B*** interfaces:

```
switch-B(config)# interface vlan 401
switch-B(config-if-Vl401)# ipv6 ospf 100 area 1
switch-B(config-if-Vl401)# ospfv3 cost 10
switch-B(config-if-Vl401)# ospfv3 priority 4
switch-B(config-if-Vl401)# exit
switch-B(config)# interface vlan 402
switch-B(config-if-Vl402)# ipv6 ospf 100 area 0
switch-B(config-if-Vl402)# ospfv3 cost 20
switch-B(config-if-Vl402)# ospfv3 ipv6 retransmit-interval 10
switch-B(config-if-Vl402)# ospfv3 transmit-delay 2
switch-B(config-if-Vl402)# ospfv3 priority 6
```

c. ***Router C*** interfaces:

```
switch-C(config)# interface vlan 501
switch-C(config-if-Vl501)# ipv6 ospf 100 area 0
switch-C(config-if-Vl501)# ospfv3 cost 20
switch-C(config-if-Vl501)# ospfv3 ipv6 retransmit-interval 10
switch-C(config-if-Vl501)# ospfv3 transmit-delay 2
switch-C(config-if-Vl501)# exit
switch-C(config)# interface vlan 502
switch-C(config-if-Vl502)# ipv6 ospf 100 area 0
switch-C(config-if-Vl502)# ospfv3 cost 20
switch-C(config-if-Vl502)# ospfv3 dead-interval 80
```

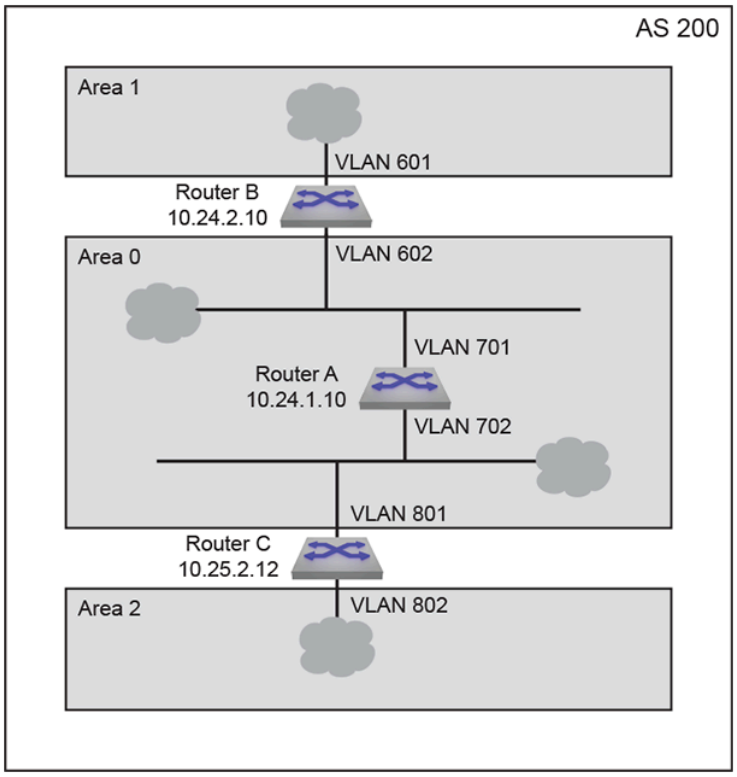
15.3.4.2 OSPFv3 Configuration Example 2

The AS in Example 2 contains three areas. Area ***0*** connects to the other areas through different routers and contains an internal router connecting two links. Area ***0*** is normal; the other areas are stub areas.

15.3.4.2.1 Example 2 Topology

OSPFv3 Example 2 displays the Example 2 topology. One ABR (***Router B***) connects area ***0*** and area ***1***; another ABR (***Router C***) connects area ***0*** and area ***2***. ***Router A*** is an internal router that connects two links in area ***0***.

Figure 15-9: OSPFv3 Example 2



Area 1 Configuration

Area ***1*** contains one link that is accessed by ***Router B***.

- ***Router B*** is accessed through ***VLAN 601***.
- The router defines a interface cost of ***10***.

Area 2 Configuration

Area ***2*** contains one link that is accessed by ***Router C***.

- ***Router C*** is accessed through ***VLAN 802***.
- The router defines a interface cost of ***20***.

Area 0 ABR Configuration

One ABR ***Router B*** link connects area ***1*** to area ***0***. One ABR ***Router C*** link connects area ***0*** to area ***2***.

- ***Router B*** is accessed through ***VLAN 602***
- ***Router C*** is accessed through ***VLAN 801***.
- Designated Router (DR): ***Router B***.
- Backup Designated Router (BDR): ***Router C***.
- Each router defines an interface cost of ***20***.
- Each router defines a retransmit-interval of ***10***.
- Each router defines a transmit-delay of ***2***.

Area 0 IR Configuration

Area ***0*** contains links connected by an internal router.

- Router A is accessed through ***vlan 701*** and ***vlan 702***.
- The ***vlan 701*** link is configured as follows:
  - Interface cost of ***10***.
- The ***vlan 702*** link is configured as follows:
  - Interface cost of ***20***.
  - Retransmit-interval of ***10***.
  - Transmit-delay of ***2***.

15.3.4.2.2 Example 2 Code

1. Configure the areas and router IDs.
   a. Router A OSPFv3 instance configuration:

```
switch-A(config)# ipv6 router ospf 200
switch-A(config-router-ospfv3)# router-id 10.24.1.10
```

   b. Router B OSPFv3 instance configuration:

```
switch-B(config)# ipv6 router ospf 200
switch-B(config-router-ospfv3)# area 1 stub
switch-B(config-router-ospfv3)# router-id 10.24.2.10
```

   c. Router C OSPFv3 instance configuration:

```
switch-C(config)# ipv6 router ospf 200
switch-C(config-router-ospfv3)# area 1 stub
switch-C(config-router-ospfv3)# router-id 10.25.2.12
```

2. Configure the interface OSPFv3 area and transmission parameters.
   a. ***Router A*** interfaces:

```
switch-A(config)# interface vlan 701
switch-A(config-if-Vl701)# ipv6 ospf 200 area 0
switch-A(config-if-Vl701)# ospfv3 cost 10
switch-A(config-if-Vl701)# exit
switch-A(config)# interface vlan 702
switch-A(config-if-Vl702)# ipv6 ospf 200 area 0
switch-A(config-if-Vl702)# ospfv3 cost 20
switch-A(config-if-Vl702)# ospfv3 ipv6 retransmit-interval 10
switch-A(config-if-Vl702)# ospfv3 transmit-delay 2
```

   b. ***Router B*** interfaces:

```
switch-B(config)# interface vlan 601
switch-B(config-if-Vl601)# ospfv3 200 area 1
switch-B(config-if-Vl601)# ospfv3 cost 10
switch-B(config-if-Vl601)# exit
switch-B(config)# interface vlan 602
switch-B(config-if-Vl602)# ospfv3 200 area 0
switch-B(config-if-Vl602)# ospfv3 cost 20
switch-B(config-if-Vl602)# ospfv3 ipv6 retransmit-interval 10
switch-B(config-if-Vl602)# ospfv3 transmit-delay 2
switch-B(config-if-Vl602)# ospfv3 priority 6
```

c. ***Router C*** interfaces:

```
switch-C(config)# interface vlan 801
switch-C(config-if-Vl801)# ospfv3 200 area 0
switch-C(config-if-Vl801)# ospfv3 cost 20
switch-C(config-if-Vl801)# ospfv3 ipv6 retransmit-interval 10
switch-C(config-if-Vl801)# ospfv3 transmit-delay 2
switch-C(config-if-Vl801)# exit
switch-C(config)# interface vlan 802
switch-C(config-if-Vl802)# ospfv3 200 area 2
switch-C(config-if-Vl802)# ospfv3 cost 20
switch-C(config-if-Vl802)# ospfv3 dead-interval 80
```

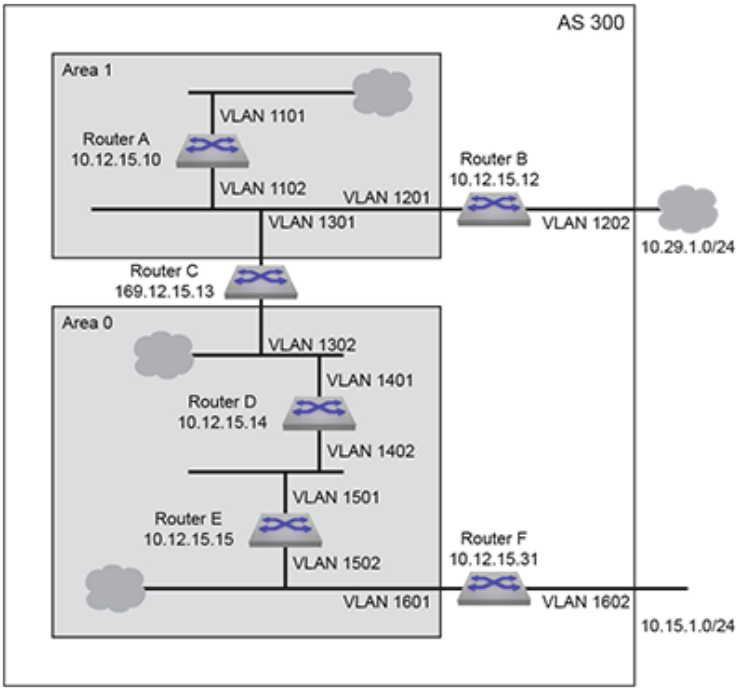
15.3.4.3 OSPFv3 Configuration Example 3

The AS in Example 3 contains two areas that connect through one ABR. Each area also contains an ASBR that connects static routes to the AS.

15.3.4.3.1 Example 3 Topology

OSPFv3 Example 3 displays the Example 3 topology. One ABR connects area ***0*** and area ***1***. ***Router C*** is an ABR that connects the areas. ***Router A*** is an internal router that connects two links in area ***1***. ***Router D*** and ***Router E*** are internal routers that connect links in area ***0***. ***Router B*** and ***Router F*** are ASBRs that connect static routes outside the AS to area ***1*** and area ***0***, respectively.

Figure 15-10: OSPFv3 Example 3



Area 0 ABR Configuration

ABR ***Router C*** connects one area ***0*** link to an area ***1*** link.

- ***Router C*** is accessed through ***VLAN 1302***.
- All interface OSPFv3 parameters are set to their default values.

Area 0 IR Configuration

Area ***0*** contains two internal routers, each of which connects two of the three links in the area.

- ***Router D*** is accessed through ***VLAN 1401*** and ***VLAN 1402***.
- ***Router E*** is accessed through ***VLAN 1501*** and ***VLAN 1502***.
- All interface OSPFv3 parameters are set to their default values.

Area 0 ASBR Configuration

ASBR ***Router F*** connects one area ***0*** link to an external link.

- ***Router F*** is accessed through ***VLAN 1601***.
- ***Router F*** connects to the external AS through ***VLAN 1602***.
- All interface OSPFv3 parameters are set to their default values.

Area 1 ABR Configuration

ABR ***Router C*** connects one area ***0*** link to an area ***1*** link.

- ***Router C*** is accessed by area ***1*** through ***VLAN 1301***.
- ***Router C*** is accessed by area ***0*** through ***VLAN 1302***.
- All interface OSPFv3 parameters are set to their default values.

Area 1 IR Configuration

Area ***1*** contains one internal router that connects two links in the area.

- ***Router A*** is accessed through ***VLAN 1101*** and ***VLAN 1102***.
- All interface OSPFv3 parameters are set to their default values.

Area 1 ASBR Configuration

ASBR ***Router B*** connects one area ***1*** link to an external link.

- ***Router B*** is access through ***VLAN 1201***.
- ***Router B*** connects to the external AS through ***VLAN 1202***.
- All interface OSPFv3 parameters are set to their default values.

15.3.4.3.2 Example 3 Code

1. Configure the areas and router IDs.
   a. ***Router A*** OSPFv3 instance configuration:

```
switch-A(config)# ipv6 router ospf 300
switch-A(config-router-ospfv3)# router-id 10.12.15.10
switch-A(config-router-ospfv3)# area 1 stub
```

   b. ***Router B*** OSPFv3 instance configuration:

```
switch-B(config)# ipv6 router ospf 300
switch-B(config-router-ospfv3)# router-id 10.12.15.12
switch-B(config-router-ospfv3)# area 1 stub
```

   c. ***Router C*** OSPFv3 instance configuration:

```
switch-C(config)# ipv6 router ospf 300
switch-C(config-router-ospfv3)# router-id 10.12.15.13
switch-C(config-router-ospfv3)# area 1 stub
```

d. ***Router D*** OSPFv3 instance configuration:

```
switch-D(config)# ipv6 router ospf 300
switch-D(config-router-ospfv3)# router-id 10.12.15.14
```

e. ***Router E*** OSPFv3 instance configuration:

```
switch-E(config)# ipv6 router ospf 300
switch-E(config-router-ospfv3)# router-id 10.12.15.15
```

f. ***Router F*** OSPFv3 instance configuration:

```
switch-F(config)# ipv6 router ospf 300
switch-F(config-router-ospfv3)# router-id 10.12.15.31
```

2. Configure the interfaces.

a. ***Router A*** interfaces:

```
switch-A(config)# interface vlan 1101
switch-A(config-if-Vl1101)# ospfv3 300 area 1
switch-A(config-if-Vl1101)# exit
switch-A(config)# interface vlan 1102
switch-A(config-if-Vl1102)# ospfv3 300 area 1
```

b. ***Router B*** interfaces:

```
switch-B(config)# interface vlan 1201
switch-B(config-if-Vl1201)# ospfv3 300 area 1
switch-B(config-if-Vl1201)# exit
```

c. ***Router C*** interfaces:

```
switch-C(config)# interface vlan 1301
switch-C(config-if-Vl1301)# ospfv3 300 area 1
switch-C(config-if-Vl1301)# exit
switch-C(config)# interface vlan 1302
switch-C(config-if-Vl1302)# ospfv3 300 area 0
```

d. ***Router D*** interfaces:

```
switch-D(config)# interface vlan 1401
switch-D(config-if-Vl1401)# ospfv3 300 area 0
switch-D(config-if-Vl1401)# exit
switch-D(config)# interface vlan 1402
switch-D(config-if-Vl1402)# ospfv3 300 area 0
```

e. ***Router E*** interfaces:

```
switch-E(config)# interface vlan 1501
switch-E(config-if-Vl1501)# ospfv3 300 area 0
switch-E(config-if-Vl1501)# exit
switch-E(config)# interface vlan 1502
switch-E(config-if-Vl1502)# ospfv3 300 area 0
```

f. ***Router F*** interfaces:

```
switch-F(config)# interface vlan 1601
switch-F(config-if-Vl1601)# ospfv3 300 area 0
switch-F(config-if-Vl1601)# exit
```

15.3.5 OSPFv3 Commands

Global Configuration Mode

- clear ospfv3 ipv6 force-spf
- ipv6 router ospf

Interface Configuration Mode

- ipv6 ospf area
- ospfv3 authentication ipsec spi
- ospfv3 cost
- ospfv3 dead-interval
- ospfv3 encryption ipsec spi
- ospfv3 hello-interval
- ospfv3 ipv6 retransmit-interval
- ospfv3 network
- ospfv3 priority
- ospfv3 transmit-delay

Router-OSPFv3 Configuration Mode

- adjacency exchange-start threshold (OSPFv3)
- area authentication ipsec spi
- area default-cost (OSPFv3)
- area encryption ipsec spi
- area nssa (OSPFv3)
- area nssa default-information-originate (OSPFv3)
- area not-so-stubby lsa type-7 convert type-5 (OSPFv3)
- area range (OSPFv3)
- area stub (OSPFv3)
- default-information originate (OSPFv3)
- default-metric (OSPFv3)
- distance ospf intra-area (OSPFv3)
- log-adjacency-changes (OSPFv3)
- max-metric router-lsa (OSPFv3)
- maximum-paths (OSPFv3)
- no area (OSPFv3)
- passive-interface (OSPFv3)
- redistribute (OSPFv3)
- router-id (OSPFv3)
- shutdown (OSPFv3)
- timers
- timers lsa rx min interval (OSPFv3)
- timers lsa tx delay initial (OSPFv3)
- timers spf delay initial (OSPFv3)

Display Commands

- show ipv6 ospf
- show ipv6 ospf border-routers
- show ipv6 ospf database

- show ipv6 ospf database<link-state details>
- show ipv6 ospf database <link state list>
- show ipv6 ospf database link
- show ipv6 ospf database link if-name
- show ipv6 ospf database link if-type
- show ipv6 ospf interface
- show ipv6 ospf lsa-log
- show ipv6 ospf neighbor
- show ipv6 ospf neighbor state
- show ipv6 ospf neighbor summary
- show ipv6 ospf spf-log
- show ospfv3

15.3.5.1 adjacency exchange-start threshold (OSPFv3)

The **`adjacency exchange-start threshold`** command sets the exchange-start options for an OSPF instance.

The **`no adjacency exchange-start threshold`** and **`default adjacency exchange-start threshold`** command resets the default by removing the corresponding **`adjacency exchange-start threshold`** command from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

`adjacency exchange-start threshold` ***peers***

`no adjacency exchange-start threshold`

`default adjacency exchange-start threshold`

Parameters

peers Value ranges from ***1 - 4294967295***. Default value is ***10***.

Example

This command sets the adjacency exchange start threshold to ***156923***.

```
switch(config)# ipv6 router ospf 3
switch(config-router-ospf3)# adjacency exchange-start threshold 156923
switch(config-router-ospf3)#
```

15.3.5.2 area authentication ipsec spi

The **`area authentication ipsec spi`** command configures OSPFv3 authentication on an area.

The **`default area authentication`** and **`no area authentication`** commands delete the OSPFv3 authentication on an area.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

`area` ***area_id*** **`authentication ipsec spi`** ***spi_value*** {md5|sha1} passphrase {***0 unencrypted_key*** | ***7 hidden_key*** | ***LINE***}

no area *area_id* authentication ipsec spi *spi_value* {md5| sha1} passphrase {***0 unencrypted_key*** | ***7 hidden_key*** | ***LINE***}

default area *area_id* authentication ipsec spi *spi_value* {md5| sha1} passphrase {***0 unencrypted_key*** | ***7 hidden_key*** | ***LINE***}

Parameters

- **area *area_id*** configures OSPF area ID in either IP address or decimal formats. The value for decimal format ranges from ***0 to 4294967295***.
- **spi *spi_value*** configures the IPsec Security Parameter Index. The value ranges from ***0 to 4294967295***.
- **md5** configures HMAC-MD5 hash algorithm.
- **sha1** configures HMAC-SHA1 algorithm.
- ***0 unencrypted_key*** configures either a ***192*** bit 3DES key or ***128/192/256*** bit AES key in an unencrypted format.
- ***7 encrypted_key*** configures either a ***192*** bit 3DES key or ***128/192/256*** bit AES key in an encrypted format.
- ***KEY*** configures either a ***128*** bit MD5 key or a ***140*** bit SHA1 key.
- **passphrase** configures passphrase for authentication and encryption. Options include:
  - ***0 unencrypted_passphrase*** configures an unencrypted key.
  - ***7 encrypted_passphrase*** configures an encrypted key.
  - ***LINE*** uses passphrase string to derive keys for authentication and encryption.

Related Commands

- **ospfv3 authentication ipsec spi**
- **area encryption ipsec spi**

Guidelines

Passphrase and key value are exclusive. MD5 and SHA1 keys are derived from the configured passphrase.

Restriction

On the same area, EOS allows security configuration with either AH or ESP but not both. We can have one area configured with AH and another with ESP.

Examples

- This command configures OSPFv3 authentication on an area with MD5 hash algorithm.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# area 0.0.0.0 authentication ipsec spi 34 md5 0
 8FD6158BFE81ADD961241D8E4169D411
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 0.0.0.0 authentication ipsec spi 34 md5 7 $1$cNpcrQl1czqdvKAzKLtYVr6
I7+R3niuWouDKKYCFNs4/XOWG/Iap5Q==
switch(config-router-ospf3)#
```

- This command configures OSPFv3 authentication on an area with SHA1 algorithm.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# area 0.0.0.0 authentication ipsec spi 5789
 sha1 passphrase 7 $1$Ab754G0OHbGllIKqlCl7lyUKscUlpFTpvcQxQIhjJm1OUzGJD
h4bLWxSdKHvWMo6
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 0.0.0.0 authentication ipsec spi 5789 sha1 passphrase 7
Ab754G0OHbGllIKqlCl7lyUKscUlpFTpvcQxQIhjJm1OUzGJDh4bLWxSdKHvWMo6
switch(config-router-ospf3)#
```

- This command deletes the OSPFv3 authentication on an area.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 1.1.1.1 authentication ipsec spi 2437 md5 7 cNpcrQl1czqdv
KAzKLtYVr6I7+R3niuWouDKKYCFNs4/XOWG/Iap5Q==
   area 0.0.0.0 authentication ipsec spi 5789 sha1 passphrase 7
Ab754G0OHbGllIKqlCl7lyUKscUlpFTpvcQxQIhjJm1OUzGJDh4bLWxSdKHvWMo6
switch(config-router-ospf3)#no area 0.0.0.0 authentication
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   area 1.1.1.1 authentication ipsec spi 2437 md5 7 cNpcrQl1czqdv
KAzKLtYVr6I7+R3niuWouDKKYCFNs4/XOWG/Iap5Q==
switch(config-router-ospf3)#
```

15.3.5.3 area default-cost (OSPFv3)

The **area default-cost** command sets the cost for the default summary routes sent into an area. When the **area default-cost** command is not configured for an area, the default-cost of that area is set to ***10***.

The **no area default-cost** and **default area default-cost** command resets the default-cost value of the specified area to ***10*** by removing the corresponding **area default-cost** command from ***running-config***. The **no area (OSPFv3)** command removes all area commands for the specified area from ***running-config***, including the **area default-cost** command.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

area ***area_id*** **default-cost** ***def_cost***

no area ***area_id*** **default-cost**

default area ***area_id*** **default-cost**

Parameters

- ***area_id*** area number. ***0*** to ***4294967295*** or ***0.0.0.0*** to ***255.255.255.255***. ***Running-config*** stores value in dotted decimal notation.
- ***def_cost*** Values range from ***1 to 65535***.

Example

These commands configure a cost of ***15*** for default summary routes that an ABR sends into area ***100***.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# area 100 default 15
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 0.0.0.100 default-cost 15
switch(config-router-ospf3)#
```

15.3.5.4 area encryption ipsec spi

The **area encryption ipsec spi** command configures OSPFv3 security on an area.

The **default area encryption** and **no area encryption** commands delete the OSPFv3 security on an area.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

area ***area_id*** **encryption ipsec spi** ***spi_value*** esp{3des-cbc| aes-128-cbc | aes-192-cbc | aes-256-cbc} { 0 ***unencrypted_key*** | 7 ***encrypted_key***}{ md5| sha1} { 0 ***unencrypted_key*** | 7 ***encrypted_key*** | ***KEY***}

area ***area_id*** **encryption ipsec spi** ***spi_value*** esp null{md5 | sha1} { 0 ***unencrypted_key*** | 7 ***encrypted_key*** | ***KEY***}

area ***area_id*** **encryption ipsec spi** ***spi_value*** esp{3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | null} {md5 | sha1} { 0 ***unencrypted_key*** | 7 ***encrypted_key*** | ***LINE***}

no area ***area_id*** **encryption**

default area ***area_id*** **encryption**

Parameters

- **area** ***area_id*** configures OSPF area ID in either IP address or decimal formats. The value for decimal format ranges from ***0 to 4294967295***.
- **spi** ***spi_value*** configures the value for IPsec Security Parameter Index. The value ranges from ***0 to 4294967295***.
- **3des-cbc** configures ESP with 3DES-CBC encryption.
- **aes-128-cbc** configures ESP with AES-128-CBC encryption.
- **aes-192-cbc** configures ESP with AES-192-CBC encryption.
- **aes-256-cbc** configures ESP with AES-256-CBC encryption.
- **null** configures ESP with null encryption.
- ***0 unencrypted_key*** configures either a ***192*** bit 3DES key or ***128/192/256*** bit AES key in an unencrypted format.
- ***7 encrypted_key*** configures either a ***192*** bit 3DES key or ***128/192/256*** bit AES key in an encrypted format.
- ***KEY*** configures either a ***128*** bit MD5 key or a ***140*** bit SHA1 key.
- **md5** configures HMAC-MD5 hash algorithm.
- **sha1** configures HMAC-SHA1 algorithm.
- **passphrase** configures passphrase for authentication and encryption. Options include:
  - ***0 unencrypted_passphrase*** configures an unencrypted key.
  - ***7 encrypted_passphrase*** configures an encrypted key.
  - ***LINE*** uses passphrase string to derive keys for authentication and encryption.

Related Commands

- **area authentication ipsec spi**
- **ospfv3 encryption ipsec spi**

Guidelines

Passphrase and ***key*** values are exclusive. **MD5** and **SHA1** keys are derived from the configured passphrase.

Restriction

On the same area, EOS allows security configuration with either AH or ESP but not both. We can have one area configured with AH and another with ESP.

Examples

- This command configures OSPFv3 security on an area with 3DES-CBC encryption and MD5 hash algorithm.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# area 0.0.0.0 encryption ipsec spi 5678 esp
 3des-cbc md5 passphrase 0
```

```
8FD6158BFE81ADD961241D8E4169D411
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 0.0.0.0 encryption ipsec spi 5678 esp 3des-cbc md5 passphrase 7
$1$cNpcrQl1czqdvKAzKLtYVr6I7+R3niuWouDKKYCFNs4/XOWG/Iap5Q==
switch (config-router-ospf3)#
```

- This command deletes the OSPFv3 security on an area.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   area 0.0.0.0 encryption ipsec spi 5678 esp 3des-cbc md5 passphrase 7
$1$cNpcrQl1czqdvKAzKLtYVr6I7+R3niuWouDKKYCFNs4/XOWG/Iap5Q==
switch(config-router-ospf3)# no area 0.0.0.0 encryption
switch(config-router-ospf3)# show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

15.3.5.5 area not-so-stubby lsa type-7 convert type-5 (OSPFv3)

The **area not-so-stubby lsa type-7 convert type-5** command configures the switch to always translate Type-7 Link-State Advertisement (LSAs) to Type-5 LSAs.

The **no area not-so-stubby lsa type-7 convert type-5** and **no area not-so-stubby lsa type-7 convert type-5** commands allow LSAs to be translated dynamically by removing the **no area not-so-stubby lsa type-7 convert type-5** command from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

area ***area_id*** **not-so-stubby lsa type-7 convert type-5**

no area ***area_id*** **not-so-stubby lsa type-7 convert type-5**

default area ***area_id*** **not-so-stubby lsa type-7 convert type-5**

Parameters

area_id

- Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
- Area ***0*** (or ***0.0.0.0***) is not configurable; it is always normal.
- The ***running-config*** stores value in dotted decimal notation.

Example

These commands configure the switch to always translate ***Type-7*** Link-State Advertisement (LSAs) to ***Type-5*** LSAs.

```
switch(config)# ipv6 router ospf 3
switch(config-router-ospf3)# area 3 not-so-stubby lsa type-7 convert type-5
switch(config-router-ospf)#
```

15.3.5.6 area nssa (OSPFv3)

The **area nssa** command configures an OSPFv3 area as a Not-So-Stubby Area (NSSA). All routers in an AS must specify the same area type for identically numbered areas.

NSSA ASBRs advertise external LSAs that are part of the area, but do not advertise external LSAs from other areas.

Areas are **normal** by default; area type configuration is required only for stub NSSA areas. Area ***0*** is always a normal area and cannot be configured through this command.

The **no area nssa** command configures the specified area as a normal area by removing the specified **area nssa** command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

area ***area_id*** **nssa** [**TYPE**]

no area ***area_id*** **nssa** [**TYPE**]

default area ***area_id*** **nssa** [**TYPE**]

Parameters

- ***area_id***
  - Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
  - Area ***0*** (or ***0.0.0.0***) is not configurable; it is always normal.
  - The ***running-config*** stores value in dotted decimal notation.
- **TYPE**

  Values include:

  - ***no parameter***.
  - **nssa-only**

Example

This command configures area ***3*** as a NSSA area.

```
switch(config)# ipv6 router ospf 1
switch(config-router-ospf3)# area 3 nssa nssa-only
switch(config-router-ospf3)#
```

15.3.5.7 area nssa default-information-originate (OSPFv3)

The **area nssa default-information-originate** command sets an area as an NSSA and the generation of a type 7 default LSA is created if a default route exists in the routing table.

The switch supports three area types:

Areas are **normal** by default; area type configuration is required only for stub NSSA areas. Area ***0*** is always a normal area and cannot be configured through this command.

The **no area**and **default area** commands remove the specified area from the OSPFv3 instance by deleting all **area** commands from the ***running-config*** for the specified area, including the **area default-cost (OSPFv3)** command.

The **no area stub** and **default area stub** commands configure the specified area as a normal area.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

area ***area_id*** **nssa default-information-originate** [**VALUE**][**TYPE**][**EXCL**]

no area ***area_id*** **nssa default-information-originate** [**VALUE**][**TYPE**][**EXCL**]

default area ***area_id*** **nssa default-information-originate** [**VALUE**][**TYPE**][**EXCL**]

Parameters

All parameters except ***area_id*** can be placed in any order.

- ***area_id*.**
  - Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
  - Area ***0*** (or ***0.0.0.0***) is not configurable; it is always normal.
  - ***Running-config*** stores value in dotted decimal notation.
- **VALUE** Values include:
  - ***no parameter***.
  - **metric *1-65535***.
- **TYPE** Values include:
  - ***no parameter***.
  - **metric-type *1-2***.
- **EXCL** Values include:
  - ***no parameter***.
  - **nssa-only**.

Examples

- These commands sets area ***1*** as NSSA only and generates a ***type 7*** default LSA if a default route exists in the routing table.

```
switch(config-router-ospf3)# area 3 nssa default-information-originate nssa-
only
switch(config-router-ospf3)#
```

- These commands generates a ***type 7*** default route.

```
switch(config-router-ospf3)# area 3 nssa default-information-originate
switch(config-router-ospf3)#
```

15.3.5.8 area range (OSPFv3)

The **area range** command is used by OSPFv3 area border routers to summarize routes.

The **no area range** and **default area range** commands remove the area-range by deleting the corresponding **area range** command from the ***running-config***.

Command Mode

Router-OSPFv3 Configuratio

Command Syntax

area ***area_id*** **range** ***net_addr*** [**ADVERTISE_SETTING**][**COST_SETTING**]

no area ***area_id*** **range** ***net_addr*** [**ADVERTISE_SETTING**][**COST_SETTING**]

default area ***area_id*** **range** ***net_addr*** [**ADVERTISE_SETTING**][**COST_SETTING**]

Parameters

- ***area_id 0*** to ***4294967295*** or ***0.0.0.0*** to ***255.255.255.255***.
- ***net_addr***
- **ADVERTISE_SETTING** specifies the LSA advertising activity. Values include:
  - ***no parameter***
  - **advertise**
  - **not-advertise**

- **COST_SETTING** Values include:
  - ***no parameter***
  - **cost** ***range_cost*** Value ranges from ***1 to 65535***.

Examples

- These commands summarize routes at an area boundary ***1***.

```
switch(config)# ipv6 router ospf 1
switch(config-router-ospf3)# area 1 range 2001:0DB8:0:1::/64
switch(config-router-ospf3)#
```

- These commands modify the address range status to DoNotAdvertise.

```
switch(config)# ipv6 router ospf 1
switch(config-ospf6-router)# area 1 range 2001:0DB8:0:1::/64 not-advertise
switch(config-ospf6-router)#
```

15.3.5.9 area stub (OSPFv3)

The **area stub** command configures the area type of an OSPFv3 area.

Areas are **normal** by default.

The **no area stub** command configures the specified area as a normal area.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

area ***area_id*** **stub**

no area ***area_id*** **stub**

default area ***area_id*** **stub**

Parameters

area_id

- Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
- Area ***0*** (or ***0.0.0.0***) is not configurable; it is always **normal**.
- The ***running-config*** stores value in dotted decimal notation.

Examples

- This command configures area ***45*** as a stub area.

```
switch(config)# ipv6 router ospf 3
switch(config-router-ospf3)# area 45 stub
switch(config-router-ospf3)#
```

- This command configures area ***10.92.148.17*** as a stub area.

```
switch(config-router-ospf3)# area 10.92.148.17 stub
switch(config-router-ospf3)#
```

15.3.5.10 clear ospfv3 ipv6 force-spf

The **clear ospfv3 ipv6 force-spf** command starts the SPF algorithm without clearing the OSPF database.

Command Mode

Privileged EXEC

Command Syntax

`clear ospfv3 ipv6 force-spf` [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE Values include:

- ***no parameter*** Action is performed in the default VRF.
- **vrf *vrf_name*** Action is performed in the specified VRF.

Example

This command restarts the SPF algorithm in the default VRF without first clearing the OSPFv3 database.

```
switch(config)# clear ospfv3 ipv6 force-spf
switch(config)#
```

15.3.5.11 default-information originate (OSPFv3)

The **`default-information originate`** command generates a default external route into an OSPF domain.

The **`no default-information originate`** and **`default default-information originate`** command removes the configuration from the the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

`default-information originate` [**DURATION**][**VALUE**][**TYPE**][**MAP**]

`no default-information originate`

`default default-information originate`

Parameters

All parameters can be placed in any order.

- **DURATION** Values include:
  - ***no parameter***.
  - **always**.
- **VALUE** Values include:
  - ***no parameter***.
  - **metric *1-65535***.
- **TYPE** Values include:
  - ***no parameter***.
  - **metric-type *1-2***.
- **MAP** Values include:
  - ***no parameter***.
  - **route-map *map_name***.

Examples

- These commands will advertise the OSPFv3 default route regardless of whether the switch has a default route configured.

```
switch(config)# ipv6 router ospf 1
switch(config-router-ospf3)# default-information originate always
switch(config-router-ospf3)# show active
ipv6 router ospf 1
   default-information originate always
```

- These commands configures OSPF area ***1*** as metric of ***100*** for the default route with an external metric type of ***Type 1***.

```
switch(config)# ipv6 router ospf 1
switch(config-router-ospf3)# default-information originate metric 100
 metric-type 1
switch(config-router-ospf3)# show active
ipv6 router ospf 1
   default-information originate metric 100 metric-type 1
switch(config-router-ospf3)#
```

15.3.5.12 default-metric (OSPFv3)

The **default-metric** command sets default metric value for routes redistributed into the OSPFv3 domain.

The **no default-metric** and **default default-metric** commands restores the default metric to its default value of ***10*** by removing the **default-metric** command from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

default-metric ***def_metric***

no default-metric

default default-metric

Parameter

def_metric Values range from ***1 to 65535***. Default value is ***10***.

Example

These commands configure a default metric of ***30*** for routes redistributed into OSPFv3.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# default-metric 30
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   default-metric 30
switch(config-router-ospf3)#
```

15.3.5.13 distance ospf intra-area (OSPFv3)

The **distance ospf intra-area** command sets the administrative distance for routes in a single OSPFv3 area. The default is ***110***.

The **no distance ospf intra-area** and **default distance ospf intra-area** commands remove the **distance ospf intra-area** command from the ***running-config***, returning the OSPFv3 intra-area distance setting to the default value of ***110***

Command Mode

Router-OSPFv3 Configuration

Command Syntax

`distance ospf intra-area` ***distance***

`no distance ospf intra-area`

`default distance ospf intra-area`

Parameter

distance Values range from ***1 to 255***. Default is ***110***.

Example

This command configures a distance of ***90*** for all OSPFv3 intra-area routes on the switch.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# distance ospf intra-area 90
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   distance ospf intra-area 90
switch(config-router-ospf3)#
```

15.3.5.14 ipv6 ospf area

The **`ipv6 ospf area`** command enables OSPFv3 on the interface and associates the area to the interface.

OSPFv3 areas are configured in by **`no area (OSPFv3)`** commands in ***router-OSPFv3*** configuration mode.

The **`no ipv6 ospf area`** and **`default ipv6 ospf area`** commands disable OSPFv3 on the configuration mode interface by removing the corresponding **`ipv6 ospf area`** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ipv6 ospf` ***process_id*** [**`area`** ***area_id***]

`no ipv6 ospf` ***process_id*** [**`area`** ***area_id***]

`default ipv6 ospf` ***process_id*** [**`area`** ***area_id***]

Parameters

- ***process_id*** Values range from ***1 to 65535***.
- ***area_id***.
  - Valid formats: integer ***0*** to ***4294967295*** or dotted decimal ***0.0.0.0*** to ***255.255.255.255***.
  - ***Running-config*** stores value in dotted decimal notation.

Example

These commands enable OSPFv3 on VLAN interface ***200*** and associates area ***0*** to the interface.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ipv6 ospf 9 area 0
switch(config-if-Vl200)# show active
```

```
interface Vlan200
   ipv6 ospf 9 area 0.0.0.0
switch(config-if-Vl200)#
```

15.3.5.15 ipv6 router ospf

The **ipv6 router ospf** command places the switch in router-OSPFv3 configuration mode and creates and OSPFv3 instance if one does not already exist. Note that each OSPFv3 instance on the switch must have a unique process ID. A router ID for the new instance will be created if one does not already exist.

The **show ipv6 ospf** command displays the router ID of each OSPFv3 instance configured on the switch.

The **no ipv6 router ospf** and **default ipv6 router ospf** commands delete the OSPFv3 instance.

Refer to the **Router-OSPFv3 Configuration Mode** command for a list of commands available in router-OSPFv3 configuration mode.

Command Mode

Global Configuration

Command Syntax

ipv6 router ospf ***process_id*** [**VRF_INSTANCE**]

no ipv6 router ospf ***process_id*** [**VRF_INSTANCE**]

default ipv6 router ospf ***process_id*** [**VRF_INSTANCE**]

Parameters

- ***process_id*** Values range from ***1 to 65535***.
- **VRF_INSTANCE** Values include:
  - ***no parameter*** OSPF instance is in the default VRF.
  - **vrf** ***vrf_name*** OSPF instance is the specified VRF.

Examples

- This command creates an OSPFv3 instance in the default VRF with process ID ***9***.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

- This command deletes the OSPFv3 instance.

```
switch(config)# no ipv6 router ospf 9
switch(config)#
```

15.3.5.16 log-adjacency-changes (OSPFv3)

The **log-adjacency-changes** command enables syslog messages to be sent when it detects OSPFv3 link state changes or when it detects that a neighbor has gone up or down. Log message sending is enabled by default.

The **default log-adjacency-changes** command restores the default state by removing the **log-adjacency-changes** statement from the ***running-config***.

The default option (sending a message only when a neighbor goes up or down) is active when the ***running-config*** does not contain any form of the command. Entering the command in any form replaces the previous command state in the ***running-config***.

The **no log-adjacency-changes** disables link state change Syslog reporting.

The **default log-adjacency-changes** command restores the default state by removing the **log-adjacency-changes detail** or **no log-adjacency-changes** statement from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

log-adjacency-changes [**INFO_LEVEL**]

no log-adjacency-changes

default log-adjacency-changes

Parameters

INFO_LEVEL Options include:

- ***no parameter*** Sends messages when a neighbor goes up or down.
- **detail** Sends messages for all neighbor state changes.

Example

This command configures the switch to send a Syslog message when a neighbor state changes.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# log-adjacency-changes
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   log-adjacency-changes
switch(config-router-ospf3)#
```

15.3.5.17 maximum-paths (OSPFv3)

The **maximum-paths** command sets the maximum number of parallel routes that OSPFv3 supports on the switch.

The **no maximum-paths** command restores the maximum number of parallel routes that OSPFv3 supports on the switch to the default value of ***16*** by removing the **maximum-paths** command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

maximum-paths ***paths***

no maximum-paths

default maximum-paths

Parameters

paths Value range is platform dependent:

- **Arad:** Value ranges from ***1 to 128***. Default value is ***128***.
- **FM6000:** Value ranges from ***1 to 32***. Default value is ***32***.
- **PetraA:** Value ranges from ***1 to 16***. Default value is ***16***.
- **Trident:** Value ranges from ***1 to 32***. Default value is ***32***.
- **Trident II:** Value ranges from ***1 to 128***. Default value is ***128***.

Example

This command configures the maximum number of OSPFv3 parallel paths to ***12***.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# maximum-paths 12
switch(config-router-ospf3)#
```

15.3.5.18 max-metric router-lsa (OSPFv3)

The **max-metric router-lsa** command configures OSPF to include the maximum value in LSA metric fields to keep other network devices from using the switch as a preferred intermediate SPF hop.

The **no max-metric router-lsa** and **default max-metric router-lsa** commands disable the advertisement of a maximum metric.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

max-metric router-lsa [**EXTERNAL**][**STUB**][**STARTUP**][**SUMMARY**]

no max-metric router-lsa [**EXTERNAL**][**STUB**][**STARTUP**][**SUMMARY**]

default max-metric router-lsa [**EXTERNAL**][**STUB**][**STARTUP**][**SUMMARY**]

Parameters

All parameters can be placed in any order.

- **EXTERNAL** Values include:
  - ***no parameter*** Default value of ***1***.
  - **external-lsa** Range: ***1 to 16777215***. The default value is ***0xFF0000***.
- **STUB** Values include:
  - ***no parameter*** Default value of ***2***.
  - **include-stub**.
- **STARTUP** Values include:
  - ***no parameter***
  - **on-startup**
  - **on-startup wait-for-bgp**
  - **on-startup** Range: ***5 to 86400***.

    wait-for-bgp or an **on-start** time value is not included in **no** and **default** commands.
- **SUMMARY** Values include:
  - ***no parameter*** Metric is set to the default value of ***1***.
  - **summary-lsa**.
  - **summary-lsa** Range: ***1 to 16777215***.

Example

This command configures OSPFv3 to include the maximum value in LSA metric fields until BGP has converged:

```
switch(config-router-ospf3)# max-metric router-lsa on-startup wait-for-bgp
switch(config-router-ospf3)#
```

15.3.5.19 no area (OSPFv3)

The **no area** command removes all area configuration commands for the specified OSPFv3 area. Commands removed by the **no area** command include:

- area
- nssa
- range
- stub

Area settings can be removed individually; refer to the command description page of the desired command for details.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

no area ***area_id*** [***TYPE***]

default area ***area_id*** [***TYPE***]

Parameters

- ***area_id*** area number.
  - Valid formats: integer ***1*** to ***4294967295*** or dotted decimal ***0.0.0.1*** to ***255.255.255.255***.
  - Area ***0*** (or ***0.0.0.0***) is not configurable; it is always **normal**.
  - The ***running-config*** stores value in dotted decimal notation.
- **TYPE** area type. Values include:
  - **nssa**.
  - **nssa translate type7 always**.
  - **stub**.
  - **stub no-summary**.

Example

This command remove the area ***1*** stub configuration.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# no area 1 stub
switch(config-router-ospf3)#
```

15.3.5.20 ospfv3 transmit-delay

The**ospfv3 transmit-delay** command configures the transmission delay for OSPFv3 packets.

The **no ospfv3 transmit-delay** and **default ospfv3 transmit-delay** commands restore the default transmission delay of ***1*** second on the configuration mode interface by removing the corresponding **ospfv3 transmit-delay** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ospfv3 transmit-delay ***trans***

no ospfv3 transmit-delay

default ospfv3 transmit-delay

Parameter

trans Value ranges from ***1 to 65535***; default is ***1***.

Guideline

Arista devices also support the legacy **ipv6 ospf transmit-delay** command in certain software releases of the EOS.

Example

This command configures a transmission delay of ***10*** seconds for ***VLAN 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 transmit-delay 10
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 transmit-delay 10
switch(config-if-Vl200)#
```

15.3.5.21 ospfv3 authentication ipsec spi

The **ospfv3 authentication ipsec spi** command configures OSPFv3 authentication on an interface.

The **default ospfv3 authentication** and **no ospfv3 authentication** commands delete the OSPFv3 authentication on an interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

ospfv3 authentication ipsec spi ***spi_value*** {md5 | sha1}{***0 unencrypted_key*** | ***7 hidden_key*** | ***KEY***}

ospfv3 authentication ipsec spi ***spi_value*** {md5 | sha1} passphrase{***0 unencrypted_passphrase*** | ***7 hidden_passphrase*** | ***LINE***}

no ospfv3 authentication

default ospfv3 authentication

Parameters

- **spi** ***spi_value*** configures IPsec Security Parameter Index. The value ranges from ***0 to 4294967295***.
- **md5** configures HMAC-MD5 hash algorithm.
- **sha1** configures HMAC-SHA1 algorithm.
- ***0 unencrypted_key*** configures either a ***192*** bit 3DES key or ***128/192/256*** bit AES key in an unencrypted format.
- ***7 encrypted_key*** configures either a ***192*** bit 3DES key or ***128/192/256*** bit AES key in an encrypted format.
- ***KEY*** configures either a ***128*** bit MD5 key or a ***140*** bit SHA1 key.
- **passphrase** configures passphrase for authentication and encryption. Options include:
  - ***0 unencrypted_passphrase*** configures an unencrypted passphrase.
  - ***7 encrypted_passphrase*** configures an encrypted passphrase.
  - ***LINE*** uses passphrase string to derive keys for authentication and encryption.

Related Commands

- **area authentication ipsec spi**
- **ospfv3 encryption ipsec spi**

Guidelines

Passphrase and ***key*** values are exclusive. **MD5** and **SHA1** keys are derived from the configured passphrase. Arista devices also support the legacy **ipv6 ospf authentication ipsec spi** command in certain software releases of the EOS.

Restriction

On the same interface, EOS allows security configuration with either AH or ESP but not both. We can have one interface configured with AH and another with ESP.

Examples

- This command configures OSPFv3 authentication on an interface with MD5 hash algorithm.

```
switch(config)# interface ethernet 9
switch(config-if-Et9)# ospfv3 authentication ipsec spi 3456 md5 0
 8FD6158BFE81ADD961241D8E4169D411
switch(config-if-Et9)# show active
interface Ethernet9
   no switchport
ospfv3 authentication ipsec spi 3456 md5 7 $1$xtmcMSPzEn+Njp8Lb4qryVV
OjKcjsrYuv6dx1O+nSwKQdaiRt2RPTQ==
switch(config-if-Et9)#
```

- This command configures OSPFv3 authentication on an interface with SHA1 algorithm.

```
switch(config)# interface ethernet 9
switch(config-if-Et9)# ospfv3 authentication ipsec spi 987 sha1 7
 $1$VmUkWk6IL2S343bR3BbH0RhgvxHhwBpfvB4VXKNOOQF7HJBp5VvXTfBaVYbgCkWU
switch(config-if-Et9)# show active
interface Ethernet9
   no switchport
ospfv3 authentication ipsec spi 987 sha1 7
$1$VmUkWk6IL2S343bR3BbH0RhgvxHhwBpfvB4VXKNOOQF7HJBp5VvXTfBaVYbgCkWU
switch(config-if-Et9)#
```

- This command deletes the OSPFv3 authentication on an interface.

```
switch(config)# interface ethernet 9
switch(config-if-Et9)# show active
interface Ethernet9
   no switchport
ospfv3 authentication ipsec spi 3456 md5 7 $1$xtmcMSPzEn+Njp8Lb4qryVV
OjKcjsrYuv6dx1O+nSwKQdaiRt2RPTQ==
switch(config-if-Et9)#no ospfv3 authentication
switch(config-if-Et9)#show active
interface Ethernet9
   no switchport
switch(config-if-Et9)#
```

15.3.5.22 ospfv3 cost

The **ospfv3 cost** command sets the OSPFv3 cost for the interface. The default OSPFv3 cost is ***10***.

The **no ospfv3 cost** and **default ospfv3 cost** commands restore the default cost of ***10*** for the configuration mode interface by removing the corresponding **ospfv3 cost** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ospfv3 cost` ***interface_cost***

`no ospfv3 cost`

`default ospfv3 cost`

Parameters

interface_cost Value ranges from ***1 to 65535***; default is ***10***.

Guideline

Arista devices also support the legacy **`ipv6 ospf cost`** command in certain software releases of the EOS.

Example

This command configures a cost of ***50*** for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 cost 50
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 cost 50
switch(config-if-Vl200)#
```

15.3.5.23 ospfv3 dead-interval

The **`ospfv3 dead-interval`** command sets the OSPFv3 dead interval.

The **`no ospfv3 dead-interval`** and **`default ospfv3 dead-interval`** commands restore the default dead interval of ***40*** seconds on the configuration mode interface by removing the corresponding **`ospfv3 dead-interval`** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ospfv3 dead-interval` ***time***

`no ospfv3 dead-interval`

`default ospfv3 dead-interval`

Parameter

time Value ranges from ***1 to 65535***; default is ***40***.

Guideline

Arista devices also support the legacy **ipv6 ospf dead-interval** command in certain software releases of the EOS.

Example

This command configures a dead interval of ***75*** seconds for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 dead-interval 75
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 dead-interval 75
switch(config-if-Vl200)#
```

15.3.5.24 ospfv3 encryption ipsec spi

The **ospfv3 encryption ipsec spi** command configures OSPFv3 security on an interface.

The **default ospf3 encryption** and **no ospfv3 encryption** commands delete the OSPFv3 security on an interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

ospfv3 encryption ipsec spi ***spi_value*** esp {3des-cbc | aes-128-cbc | aes-128-cbc | aes-192-cbc}{***0 unencrypted_key*** | 7 ***encrypted_key}*** {md5 | sha1}{***0 unencrypted_key*** | ***7 encrypted_key*** | ***KEY***}

ospfv3 encryption ipsec spi ***spi_value*** esp {3des-cbc | aes-128-cbc | aes-128-cbc | aes-192-cbc}{***0 unencrypted_key*** | ***7 encrypted_key***}{md5 | sha1} passphrase {***0 unencrypted_passphrase*** | ***7 encrypted_passphrase*** | ***LINE***}

ospfv3 encryption ipsec spi ***spi_value*** esp null {md5 | sha1}{***0 unencrypted_key*** | ***7 encrypted_key*** | ***KEY***}

ospfv3 encryption ipsec spi ***spi_value*** esp {md5 | sha1} passphrase {***0 unencrypted_passphrase*** | ***7 encrypted_passphrase*** | ***LINE***}

default ospfv3 encryption

no ospf3 encryption

Parameters

- **spi *spi_value*** configures the value for IPsec Security Parameter Index. The value ranges from ***0 to 4294967295***.
- **3des-cbc** configures ESP with 3DES-CBC encryption.
- **aes-128-cbc** configures ESP with AES-128-CBC encryption.
- **aes-192-cbc** configures ESP with AES-192-CBC encryption.
- **aes-256-cbc** configures ESP with AES-256-CBC encryption.
- **null** configures ESP with null encryption.
- ***0 unencrypted_key*** configures either a 192 bit 3DES key or 128/192/256 bit AES key in an unencrypted format.
- ***7 encrypted_key*** configures either a 192 bit 3DES key or 128/192/256 bit AES key in an encrypted format.
- **md5** configures HMAC-MD5 hash algorithm.
- **sha1** configures HMAC-SHA1 algorithm.
- ***KEY*** configures either a 128 bit MD5 key or a 140 bit SHA1 key.
- **passphrase** configures passphrase for authentication and encryption. Options include:

- ***0 unencrypted_passphrase*** configures an unencrypted passphrase.
- ***7 encrypted_passphrase*** configures an encrypted passphrase.
- ***LINE*** uses passphrase string to derive keys for authentication and encryption.

Related Commands

- **area encryption ipsec spi**
- **ospfv3 authentication ipsec spi**

Guidelines

Passphrase and key value are exclusive. MD5 and SHA1 keys are derived from the configured passphrase. Arista devices also support the legacy **ipv6 ospf encryption ipsec spi** command in certain software releases of the EOS.

Restrictions

On the same interface, EOS allows security configuration with either AH or ESP but not both. We can have one interface configured with AH and another with ESP.

Examples

- This command configures OSPFv3 security on an interface with 3DES-CBC encryption and **SHA1** algorithm.

```
switch(config)# interface ethernet 9
switch(config-if-Et9)# ospfv3 encryption ipsec spi 345 esp 3des-cbc sha1
 passphrase 0 2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
switch(config-if-Et9)# show active
interface Ethernet9
   no switchport
ospfv3 encryption ipsec spi 345 esp 3des-cbc sha1 passphrase 7
$1$VmUkWk6IL2S343bR3BbH0RhgvxHhwBpfvB4VXKNOOQF7HJBp5VvXTfBaVYbgCkWU
switch(config-if-Et9)#
```

- This command configures OSPFv3 security on an interface with 3DES-CBC encryption and **MD5** hash algorithm.

```
switch(config)# interface ethernet 9
switch(config-if-Et9)# ospfv3 encryption ipsec spi 345 esp 3des-cbc
 md5 passphrase 7 $1$VmUkWk6IL2S343bR3BbH0RhgvxHhwBpfvB4VXKNOOQF7HJBp5
VvXTfBaVYbgCkWU
switch(config-if-Et9)# show active
interface Ethernet9
   no switchport
ospfv3 encryption ipsec spi 345 esp 3des-cbc md5 passphrase 7
$1$VmUkWk6IL2S343bR3BbH0RhgvxHhwBpfvB4VXKNOOQF7HJBp5VvXTfBaVYbgCkWU
switch(config-if-Et9)#
```

- This command deletes the OSPFv3 security on an interface.

```
switch(config)# interface ethernet 9
switch(config-if-Et9)# show active
interface Ethernet9
   no switchport
ospfv3 encryption ipsec spi 3456 md5 7 $1$xtmcMSPzEn+Njp8Lb4qryVV
OjKcjsrYuv6dx1O+nSwKQdaiRt2RPTQ==
switch(config-if-Et9)#no ospfv3 encryption
switch(config-if-Et9)#show active
interface Ethernet9
   no switchport
switch(config-if-Et9)#
```

15.3.5.25 ospfv3 hello-interval

The **ospfv3 hello-interval** command sets the OSPFv3 hello interval. The hello interval is the period between the transmission of consecutive hello packets.

Each OSPFv3 neighbor should be the same hello interval and should not be longer than any neighbors dead interval.

The **no ospfv3 hello-interval** and **default ospfv3 hello-interval** commands restore the default hello interval of ***10*** seconds on the configuration mode interface by removing the **ospfv3 hello-interval** command from **running-config**.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ospfv3 hello-interval ***time***

no ospfv3 hello-interval

default ospfv3 hello-interval

Parameter

time Values range from ***1 to 65535***; default is ***10***.

Guideline

Arista devices also support the legacy **ipv6 ospf hello-interval** command in certain software releases of the EOS.

Example

This command configures a hello interval of ***45*** seconds for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 hello-interval 45
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 hello-interval 45
switch(config-if-Vl200)#
```

15.3.5.26 ospfv3 ipv6 retransmit-interval

The **ospfv3 ipv6 retransmit-interval** command configures the link state advertisement retransmission interval.

The **no ospfv3 ipv6 retransmit-interval** and **default ospfv3 ipv6 retransmit-interval** commands restore the default retransmission interval of ***5*** seconds on the configuration mode interface by removing the corresponding **ospfv3 ipv6 retransmit-interval** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ospfv3 ipv6 retransmit-interval ***period***

no ospfv3 ipv6 retransmit-interval

default ospfv3 ipv6 retransmit-interval

Parameter

period Value ranges from ***1 to 65535***; default is ***5***.

Example

This command configures a retransmission interval of ***25*** seconds for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 ipv6 retransmit-interval 25
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 ipv6 retransmit-interval 25
switch(config-if-Vl200)#
```

15.3.5.27 ospfv3 network

The **ospfv3 network** command sets the configuration mode interface as a point-to-point link. By default, interfaces are set as broadcast links.

The **no ospfv3 network** and **default ospfv3 network** commands set the configuration mode interface as a broadcast link by removing the corresponding **ospfv3 network** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ospfv3 network point-to-point

no ospfv3 network

default ospfv3 network

Guideline

Arista devices also support the legacy **ipv6 ospf network** command in certain software releases of the EOS.

Examples

- This command configures ***interface vlan 200*** as a point-to-point link.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 network point-to-point
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 network point-to-point
switch(config-if-Vl200)#
```

- This command restores ***interface ethernet 10*** as a broadcast link.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# no ospfv3 network
switch(config-if-Vl200)# show active
interface Vlan200
switch(config-if-Vl200)#
```

15.3.5.28 ospfv3 priority

The **`ospfv3 priority`** command configures the OSPFv3 router priority.

The **`no ospfv3 priority`** and **`default ospfv3 priority`** commands restore the default priority (***1***) on the interface by removing the corresponding **`ospfv3 priority`** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ospfv3 priority` ***priority_level***

`no ospfv3 priority`

`default ospfv3 priority`

Parameter

priority_level Settings range from ***0 to 255***.

Guideline

Arista devices also support the legacy **`ipv6 ospf priority`**command in certain software releases of the EOS.

Example

This command configures a router priority of ***128*** for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# ospfv3 priority 128
switch(config-if-Vl200)# show active
interface Vlan200
   ospfv3 priority 128
switch(config-if-Vl200)#
```

15.3.5.29 passive-interface (OSPFv3)

The **`passive-interface`** command disables OSPF on an interface range. All interfaces are active by default.

The **`no passive-interface`** and **`default passive-interface`** commands enable OSPFv3 on the specified interface range by removing the corresponding **`passive-interface`** statements from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

`passive-interface` **INTERFACE_NAME**

`no passive-interface` **INTERFACE_NAME**

`default passive-interface` **INTERFACE_NAME**

Parameters

- **INTERFACE_NAME** Options include:
  - **ethernet** ***e_range***
  - **loopback** ***l_range***
  - **management** ***m_range***
  - **port-channel** ***p_range***
  - **vlan** ***v_range***
  - **vxlan** ***vx_range***
  - **default**

Valid ***e_range***, ***l_range***, ***m_range***, ***p_range v_range***, and ***vx_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Example

This command configures VLAN interfaces ***101*** through ***103*** as passive interfaces.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# passive-interface vlan 101-103
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   passive-interface Vlan101
   passive-interface Vlan102
   passive-interface Vlan103
switch(config-router-ospf3)#
```

15.3.5.30 redistribute (OSPFv3)

The **`redistribute`** command enables the advertising of all specified routes into the OSPFv3 domain as external routes.

The **`no redistribute`** and **`default redistribute`** commands remove the corresponding **`redistribute`** command from the ***running-config***, disabling route redistribution for the specified route type.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

`redistribute` **ROUTE_TYPE ROUTE_MAP**

`no redistribute` **ROUTE_TYPE**

`default redistribute` **ROUTE_TYPE**

Parameters

- **ROUTE_TYPE** Options include:
  - **BGP**
  - **connected**
  - **static**
- **ROUTE_MAP** Options include:

- **route-map *map_name***

Example

The **redistribute static** command starts the advertising of static routes as OSPFv3 external routes.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# redistribute static
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   redistribute connected
   redistribute static
switch(config-router-ospf3)#
```

15.3.5.31 router-id (OSPFv3)

The **router-id** command assigns the router ID for an OSPFv3 instance. The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IP address.
3. The highest IP address present on the device.

Note: When configuring VXLAN on an MLAG, always manually configure the OSPFv3 router ID to prevent the switch from using the common VTEP IP address as the router ID.

The **no router-id** and **default router-id** commands remove the router ID command from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

router-id *identifier*

no router-id

default router-id

Parameters

identifier Value ranges from ***0.0.0.0 to 255.255.255.255*** (dotted decimal notation).

Example

This command assigns ***10.10.1.4*** as the router ID for the OSPFv3 instance.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# router-id 10.10.1.4
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   router-id 15.10.1.4
switch(config-router-ospf3)#
```

15.3.5.32 show ipv6 ospf border-routers

The **show ipv6 ospf border-routers** command displays the OSPF routing table entries.

Command Mode

EXEC

Command Syntax

show ipv6 ospf border-routers [**VRF_INSTANCE**]

Parameters

- **VRF_INSTANCE** Values include:
  - ***no parameter*** Displays information for all VRFs.
  - **vrf *vrf_name*** Displays information for the specified VRF.

Example

This command displays the ABRs and ASBRs configured in the switch in all VRFs.

```
switch# show ipv6 ospf border-routers
Routing Process "ospf 9", VRF default
 Router 10.37.0.32 area 0.0.0.0 ASBR
 Router 10.37.0.18 area 0.0.0.0 ASBR
 Router 10.37.0.22 area 0.0.0.0 ASBR ABR
 Router 10.37.0.31 area 0.0.0.0 ASBR ABR
 Router 10.37.0.58 area 0.0.0.0 ASBR
 Router 10.37.0.37 area 0.0.0.0 ASBR
 Router 10.37.0.22 area 0.0.0.2 ASBR ABR
 Router 10.37.0.31 area 0.0.0.2 ASBR ABR switch>
```

15.3.5.33 show ipv6 ospf database link if-name

The **show ipv6 ospf database link** command displays link state advertisement details. The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database link if-name [**INTF_ID**][**LS_ID**][**ROUTER**][**DATA_LEVEL**]

Parameters

- **INTF_ID** Options include:
  - **ethernet *e_range*** Ethernet interface list.
  - **loopback *l_range*** Loopback interface list.
  - **management *m_range*** Management interface list.
  - **port-channel *p_range*** Channel group interface list.
  - **vlan *v_range*** VLAN interface list.
  - **vxlan *vx_range*** VXLAN interface list.

    Valid ***range*** formats include number, range, or comma-delimited list of numbers and ranges.
- **LS_ID** Options include:
  - ***no parameter***
  - ***A.B.C.D***
- **ROUTER** Options include:
  - ***no parameter***
  - **adv-router *a.b.c.d***
  - **self-originate**
- **DATA_LEVEL** Options include:
  - ***no parameter***
  - **detail**

Example

This command displays information for ***ethernet 4/1*** link state advertisements.

```
switch# show ipv6 ospf database link if-name ethernet 4/1
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 1":

switch>
```

15.3.5.34 show ipv6 ospf database link if-type

The **show ipv6 ospf database link** command displays information of the link state advertisements. The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database link if-type [**INTF_TYPE**][**LS_ID**][**ROUTER**][**DATA_LEVEL**]

Parameters

- **INTF_TYPE**
  - **broadcast**
  - **nbma**
  - **p2mp**
  - **p2p**
- **LS_ID** Options include:
  - ***no parameter***
  - ***A.B.C.D***
- **ROUTER** Options include:
  - ***no parameter***
  - **adv-router *a.b.c.d***
  - **self-originate**
- **DATA_LEVEL** Options include:
  - ***no parameter***
  - **detail**

Example

This command displays LSA information for the interfaces configured for broadcast transmissions.

```
switch# show ipv6 ospf database link if-type broadcast
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 1":

  Interface et4 LSDB
```

```
Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.61    10.26.0.491378 0x80000027   0x00f8b0
 LNK       0.0.0.20    10.26.0.231371 0x80000027   0x005423

  Interface et7 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.61    10.26.0.501298 0x80000028   0x005e0d
 LNK       0.0.0.38    10.26.0.231291 0x80000028   0x00ce8d

  Interface vlan3901 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.36    10.26.0.22216 0x800000b0   0x00c2b1
 LNK       0.0.0.19    10.26.0.23231 0x800000b0   0x00cfca switch>
```

15.3.5.35 show ipv6 ospf database <link state list>

The **show ipv6 ospf database** command displays the OSPF link state advertisements that originate on a switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database [**FILTER**][**LINKSTATE_ID**][**ROUTER**][**DATA_LEVEL**]

Parameters

- **FILTER** Filters the output of the command by specifying areas. Options include:
  - ***no parameter***
  - **area *A.B.C.D***
  - **area backbone**
  - **as**
  - **as external**
- **LINKSTATE_ID** Options include:
  - ***no parameter***
  - ***A.B.C.D***
- **ROUTER** Options include:
  - ***no parameter***
  - **adv-router *a.b.c.d***
  - **self-originate**
- **DATA_LEVEL** Options include:
  - ***no parameter***
  - **detail**

Example

This command displays the OSPFv3 database of link state advertisements.

```
switch# show ipv6 ospf database 10.26.0.23
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router
```

```
Routing Process "ospf 9":

  AS Scope LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 AEX        0.0.0.5    10.37.0.37   15 0x80000005   0x00be82
 AEX        0.0.0.9    10.37.0.22 1747 0x8000002b   0x00df56
 AEX        0.0.0.3    10.37.0.46  599 0x8000002d   0x00651d


Area 0.0.0.0 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 RTR        0.0.0.0    10.37.0.32  234 0x80000031   0x00585a
 NTW       0.0.0.26    10.37.0.32  271 0x80000005   0x005609
 NAP       0.0.0.26    10.37.0.32  274 0x80000005   0x00964c


Interface vlan3911 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.38    10.37.0.22  267 0x80000005   0x00a45a
 LNK       0.0.0.23    10.37.0.23  270 0x8000002c   0x005b7e

  Interface vlan3902 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.17    10.37.0.11 1535 0x8000002b   0x007120
 LNK       0.0.0.37    10.37.0.22    7 0x8000002b   0x00ce23
 LNK       0.0.0.22    10.37.0.23  250 0x8000002d   0x00c350 switch>
```

15.3.5.36 show ipv6 ospf database link

The **`show ipv6 ospf database link`** command displays details of the specified link state advertisements. The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

`show ipv6 ospf database link` [**LINKSTATE_ID**][**ROUTER**][**DATA_LEVEL**]

Parameters

- **LINKSTATE_ID** Options include:
  - ***no parameter***
  - ***A.B.C.D***
- **ROUTER** Options include:
  - ***no parameter***
  - **adv-router *a.b.c.d***
  - **self-originate**
- **DATA_LEVEL** Options include:
  - ***no parameter***
  - **detail**

Example

This command displays information about the Open Shortest Path First (OSPF).

```
switch# show ipv6 ospf database link
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 9":

switch>
```

15.3.5.37 show ipv6 ospf database

The **show ipv6 ospf database** command displays data from the OSPF database. The switch can return link state data for a single VRF or for all VRFs on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database [**VRF_INSTANCE**]

Parameters

- **VRF_INSTANCE** Values include:
  - ***no parameter*** Displays information for all VRFs.
  - **vrf *vrf_name*** Displays information for the specified VRF.

Example

This command displays OSPF database information for VRF blue.

```
switch# show ipv6 ospf database vrf blue
Codes: AEX - AS External, GRC - Grace,
 IAP - Inter Area Prefix, IAR - Inter Area Router,
 LNK - Link, NAP - Intra Area Prefix,
 NSA - Not So Stubby Area, NTW - Network,
 RTR - Router
Routing Process "ospf 9", VRF blue
AS Scope LSDB switch>
```

15.3.5.38 show ipv6 ospf database <link-state details>

The **show ipv6 ospf database <link-state details>** command displays detailed information about the specified link state advertisements. The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database [**FILTER**][**LINK_TYPE**][**LINKSTATE_ID**][**ROUTER**][**DATA_LEVEL**]

Parameters

- **FILTER** Filters the output of the command by specifying areas. Options include:
  - **area *A.B.C.D***

    - **area backbone**
- **LINK_TYPE** Parameter options include:
    - **router**
    - **network**
    - **inter-area-prefix**
    - **inter-area-router**
    - **intra-area-prefix**
    - **nssa**
- **LINKSTATE_ID** Options include:
    - ***no parameter***
    - ***A.B.C.D***
- **ROUTER** Options include:
    - ***no parameter***
    - **adv-router *a.b.c.d***
    - **self-originate**
- **DATA_LEVEL** Options include:
    - ***no parameter***
    - **detail**

Example

This command displays the OSPF database summary.

```
switch# show ipv6 ospf database detail
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 9":

  AS Scope LSDB

LSA Type: AEX
    Link State ID: 0.0.0.1
    Advertising Router: 10.21.4.9
    Age: 1123
    Sequence Number: 0x80000001
    Checksum: 0x009c89
    Length: 40
    Metric Type: 2
    Metric: 1
    External Route Tag: 0
  Prefix
      Prefix: fd7a:629f:52a4:1::
      Length: 64
      Options: (null)
      Metric: 0

Area 0.0.1.44 LSDB

LSA Type: LNK
    Link State ID: 0.0.0.14
    Advertising Router: 10.26.0.11
    Age: 1285
    Sequence Number: 0x800000c1
```

```
    Checksum: 0x00629b
    Length: 56
    Option Priority: 16777235
    Link Local Addr: fe80::21c:73ff:fe0b:a80e
    Number of Prefixes: 1

  Prefix
      Prefix: fd7a:629f:52a4:fe08::
      Length: 64
      Options: (null)
      Metric: 0

LSA Type: LNK
    Link State ID: 0.0.0.34
    Advertising Router: 10.26.0.22
    Age: 1042
    Sequence Number: 0x800000c2
    Checksum: 0x00bd9f
    Length: 56
    Option Priority: 16777235
    Link Local Addr: fe80::21c:73ff:fe01:5fe1
    Number of Prefixes: 1

  Prefix
      Prefix: fd7a:629f:52a4:fe08::
      Length: 64
      Options: (null)
      Metric: 0

LSA Type: LNK
    Link State ID: 0.0.0.15
    Advertising Router: 10.26.0.23
    Age: 1128
    Sequence Number: 0x800000c7
    Checksum: 0x00d4ab
    Length: 56
    Option Priority: 16777235
    Link Local Addr: fe80::21c:73ff:fe00:1319
    Number of Prefixes: 1

  Prefix
      Prefix: fd7a:629f:52a4:fe08::
      Length: 64
      Options: (null)
      Metric: 0

  Interface vlan3925 LSDB



LSA Type: LNK
    Link State ID: 0.0.0.153
    Advertising Router: 10.27.0.52
    Age: 1186
    Sequence Number: 0x800009b6
    Checksum: 0x002f27
    Length: 56
    Option Priority: 16777235
    Link Local Addr: fe80::21c:73ff:fe17:3906
    Number of Prefixes: 1

  Prefix
      Prefix: fd7a:629f:52a4:fe67::
      Length: 64
```

```
      Options: (null)
      Metric: 0


  Interface lo0 LSDB switch>
```

15.3.5.39 show ipv6 ospf interface

The **show ipv6 ospf interface** command displays OSPFv3 information on interfaces where OSPFv3 is enabled.

Command Mode

EXEC

Command Syntax

show ipv6 ospf interface [VRF_INSTANCE]

Parameters

- **VRF_INSTANCE** Values include:
  - ***no parameter*** Displays information for all VRFs.
  - **vrf *vrf_name*** Displays information for the specified VRF.

Example

This command displays OSPFv3 information for interfaces where OSPFv3 is enabled.

```
switch# show ipv6 ospf interface
Ethernet17 is up
 Interface Address fe80::48c:73ff:fe00:1319, VRF default, Area 0.0.0.0
 Network Type Broadcast, Cost 10
 Transmit Delay is 1 sec, State Backup DR, Priority 1
 Designated Router is 10.37.0.37
 Backup Designated Router is 10.37.0.23
 Timer intervals configured, Hello 10, Dead 40, Retransmit 5
 Neighbor Count is 1
 Options are R E V6
Vlan31 is up
 Interface Address fe80::48c:73ff:fe00:1319, VRF default, Area 0.0.0.0
 Network Type Broadcast, Cost 10
 Transmit Delay is 1 sec, State Backup DR, Priority 1
 Designated Router is 10.37.0.22
 Backup Designated Router is 10.37.0.23
 Timer intervals configured, Hello 10, Dead 40, Retransmit 5
 Neighbor Count is 1
 Options are R E V6
Vlan32 is up
 Interface Address fe80::48c:73ff:fe00:1319, VRF default, Area 0.0.0.0
 Network Type Broadcast, Cost 10
 Transmit Delay is 1 sec, State DR Other, Priority 1
 Designated Router is 10.37.0.11
 Backup Designated Router is 10.37.0.22
 Timer intervals configured, Hello 10, Dead 40, Retransmit 5
 Neighbor Count is 2
 Options are R E V6 switch>
```

15.3.5.40 show ipv6 ospf lsa-log

The **show ipv6 ospf lsa-log** command displays log entries when LSA update messages are sent or received for OSPFv3.

Command Mode

EXEC

Command Syntax

show ipv6 ospf [**PROCESS_ID**] **lsa-log** [**VRF_INSTANCE**]

Parameters

- **PROCESS_ID** OSPFv3 process ID. Values include:
  - ***no parameter*** Displays information for all process IDs.
  - ***1 to 65535*** Displays information for the specified process ID.
- **VRF_INSTANCE** Values include:
  - ***no parameter*** Displays information for all VRFs.
  - vrf ***vrf_name*** Displays information for the specified VRF.

Example

This command displays log entries when LSA update messages are sent or received for OSPFv3.

```
switch# show ipv6 ospf lsa-log
OSPF3 Process 3.3.3.3, VRF default, LSA Throttling Log:
[04:21:09] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 2000 msecs
[04:21:08] type 1: 3.3.3.3/32 [3.3.3.3], event 2, backoff restarted, new hold
 value 900 msecs
[04:21:00] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 3000 msecs
[04:21:00] type 1: 3.3.3.3/32 [3.3.3.3], event 4, maxwait value changed, new
 hold value 3000
msecs
/* Here the maxwait value was changed to 3000 from earlier 32000, this is not
 part of the log */
[04:20:42] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 32000 msecs
[04:20:10] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 32000 msecs
[04:19:54] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 16000 msecs
[04:19:46] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 8000 msecs
[04:19:42] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 4000 msecs
[04:19:40] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value
 2000 msecs
[04:19:39] type 1: 3.3.3.3/32 [3.3.3.3], event 2, backoff restarted, new hold
 value 900 msecs
[04:19:22] type 1: 4.4.4.4/32 [4.4.4.4], event 3, discarded, was early by 995
 msecs
[04:19:22] type 1: 3.3.3.3/32 [3.3.3.3], event 0, backoff started, new hold
 value 1000 msecs switch>
```

15.3.5.41 show ipv6 ospf neighbor state

The **show ipv6 ospf neighbor state** command displays the state information on OSPF neighbors on a per-interface basis.

Command Mode

EXEC

Command Syntax

show ipv6 ospf neighbor state **STATE_NAME** [**VRF_INSTANCE**]

Parameters

- **STATE_NAME** Values include:
  - **2-ways**
  - **attempt**
  - **down**
  - **exch-start**
  - **exchange**
  - **full**
  - **restart**
  - **init**
  - **loading**
- **VRF_INSTANCE** Values include:
  - ***no parameter*** Displays information for all VRFs.
  - **vrf *vrf_name*** Displays information for the specified VRF.

Example

This command displays OSPF information for neighboring devices that are adjacent.

```
switch# show ipv6 ospf neighbor state full
Routing Process "ospf 3":
switch>
```

15.3.5.42 show ipv6 ospf neighbor summary

The **show ipv6 ospf neighbor** summary command displays a single line of state information for each OSPFv3 neighbor.

Command Mode

EXEC

Command Syntax

show ipv6 ospf neighbor summary [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE Values include:

- ***no parameter*** Displays information for all VRFs.
- **vrf *vrf_name*** Displays information for the specified VRF.

Example

This command shows the summary information for the OSPFv3 neighbors.

```
switch# show ipv6 ospf neighbor summary
Routing Process "ospf 1":
      3 neighbors are in state Down
      3 neighbors are in state Full
      5 neighbors are in state Init
      0 neighbors are in state Loading
      0 neighbors are in state Attempt
      3 neighbors are in state Restarting
      0 neighbors are in state Exchange
      3 neighbors are in state 2 Ways
      0 neighbors are in state Exch Start switch>
```

15.3.5.43 show ipv6 ospf neighbor

The **show ipv6 ospf neighbor** command displays OSPFv3 neighbor information.

Command Mode

EXEC

Command Syntax

show ipv6 ospf neighbor [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE Values include:

- ***no parameter*** Displays information for all VRFs.
- **vrf *vrf_name*** Displays information for the specified VRF.

Example

This command displays the switch's neighbors.

```
switch# show ipv6 ospf neighbor
Routing Process "ospf 9":
Neighbor 10.37.0.37 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface et12
 DR is 10.37.0.37 BDR is 10.37.0.23
 Options is 0
 Dead timer is due in 37 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3911
 DR is 10.37.0.22 BDR is 10.37.0.23
 Options is 0
 Dead timer is due in 31 seconds
Neighbor 10.37.0.11 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3902
 DR is 10.37.0.11 BDR is 10.37.0.22
 Options is 0
 Dead timer is due in 33 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3902
 DR is 10.37.0.11 BDR is 10.37.0.22
 Options is 0
 Dead timer is due in 31 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3923
 DR is 10.37.0.22 BDR is 10.37.0.46
 Options is 0
 Dead timer is due in 31 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3908
 DR is 10.37.0.22 BDR is 10.37.0.21
 Options is 0
 Dead timer is due in 39 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.2 interface vlan3992
 DR is 10.37.0.22 BDR is 10.37.0.23
 Options is 0
 Dead timer is due in 39 seconds switch>
```

15.3.5.44 show ipv6 ospf spf-log

The **show ipv6 ospf spf-log** command displays when and how long the switch took to run a full SPF calculation for OSPFv3.

Command Mode

EXEC

Command Syntax

show ipv6 ospf [**PROCESS_ID**] **spf-log** [**VRF_INSTANCE**]

Parameters

- **PROCESS_ID** OSPFv3 process ID. Values include:
  - ***no parameter*** Displays information for all process IDs.
  - ***1 to 65535*** Displays information for the specified process ID.
- **VRF_INSTANCE** Values include:
  - ***no parameter*** Displays information for all VRFs.
  - **vrf *vrf_name*** Displays information for the specified VRF.

Example

This command displays the SPF information for OSPFv3 in all VRFs.

```
switch# show ipv6 ospf spf-log
OSPF3 Process 172.26.0.22, VRF default
TIME          EVENT                     REASON
04:54:52.070  SPF ran for 0.70 ms
04:54:52.070  Scheduled after 0 ms     Router LSA generation
04:54:39.151  SPF ran for 0.71 ms
04:54:39.151  Scheduled after 0 ms     Router LSA generation
04:54:12.071  SPF ran for 0.56 ms
04:54:12.070  Scheduled after 0 ms     Router LSA generation
04:54:04.153  SPF ran for 0.29 ms
04:53:59.153  Scheduled after 4999 ms  Router LSA generation
04:53:59.153  SPF ran for 0.25 ms
04:53:59.151  Scheduled after 0 ms     Router LSA generation
04:53:33.081  SPF ran for 0.3 ms
04:53:33.081  Scheduled after 0 ms     ECMP max nexthop cfg change switch>
```

15.3.5.45 show ipv6 ospf

The **show ipv6 ospf** command displays information about OSPFv3 routing.

Command Mode

EXEC

Command Syntax

show ipv6 ospf [access-list | border-routers | database | interface | lsa-log | neighbor | request-list | retransmission-list | spf-log | vrf] ***Process ID***

Parameters

- ***no parameters*** displays the complete configuration of OSPFv3 address family and routing process.
- **access-list** displays the information of configured OSPFv3 access-list. Options include:
  - ***no parameters*** displays the information of all configured OSPFv3 access lists.
  - **WORD** displays the information of the specified access list.
  - **summary** displays the summary of all configured access lists.

- **border-routers** displays the information of configured OSPFv3 border and boundary routers. Options include:
  - ***no parameters*** displays the information of all configured OSPFv3 borders and boundary routers.
  - **vrf** displays the OSPFv3 borders and boundary routers information of the specified Virtual Routing and Forwarding (VRF).
- **database** displays the summary of database. Options include:
  - ***no parameters*** displays the complete summary of database.
  - **ipv4** displays the database information of link state ID.
  - **adv-router** displays the database information of advertising router link states.
  - **area** displays the database information filtered by area scope LSAs.
  - **as** displays the database information filtered by AS scope LSAs.
  - **database-summary** displays the count of LSAs in OSPFv3 database.
  - **detail** displays the detailed information of LSA.
  - **link** displays the database information filtered by link scoped LSAs.
  - **self-originate** displays the database information of self-originated link states.
  - **vrf** displays the VRF information in OSPFV3 database.
- **interface** displays the information of OSPFv3 interfaces. Options include:
  - ***no parameters*** displays the information of all OSPFv3 interfaces.
  - **Ethernet *eth_num*** displays the information of the specified Ethernet interface. The value ranges from ***1 to 24***.
  - **Loopback *lb_num*** displays the information of the specified loop back interface. The value ranges from ***0 to 1000***.
  - **Port-Channel *pc_num*** displays the interface or sub-interface information of the specified port channel. The interface and sub-interface values of port channel ranges from ***1-1000*** and ***1-2000***. ***1-4094*** respectively.
  - **Tunnel *t_num*** displays the information of the specified tunnel. The value ranges from ***0 to 255***.
  - **Vlan *vlan_num*** displays the information of the specified VLAN interface. The value ranges from ***1 to 4094***.
  - **vrf *vrf_name*** displays the information of the specified VRF.
- **lsa-log** displays the log entries of OSPFv3 LSA updates.
- **neighbor** displays the list of OSPFv3 neighbors.
- **request-list** displays the list of all OSPFv3 LSAs requested by a router.
- **retransmission-list** displays the list of all OSPFv3 LSAs waiting to be re-sent.
- **spf-log** displays the start-time, duration of completion, and reason of delay to calculate the OSPFv3 Sender Policy Framework (SPF).
- **vrf *vrf_name*** displays the information of specified VRF.
- ***Process ID*** displays the OSPFv3 configuration of the specified process ID. The value ranges from ***1 to 65535***.

Examples

- This command displays OSPFv3 routing information for all VRFs.

```
switch# show ipv6 ospf
Routing Process "ospfv3 0" with ID 11.1.11.1 and Instance 0 VRF default
  FIPS mode disabled
  It is not an autonomous system boundary router and is not an area border
 router
  Minimum LSA arrival interval 1000 msecs
  Initial LSA throttle delay 1000 msecs
  Minimum hold time for LSA throttle 5000 msecs
  Maximum wait time for LSA throttle 5000 msecs
  It has 0 fully adjacent neighbors
  Number of areas in this router is 0. 0 normal, 0 stub, 0 nssa
```

```
   Number of LSAs 0
   Initial SPF schedule delay 0 msecs
   Minimum hold time between two consecutive SPFs 5000 msecs
   Current hold time between two consecutive SPFs 5000 msecs
   Maximum wait time between two consecutive SPFs 5000 msecs
   SPF algorithm last executed 00:07:13 ago
   No scheduled SPF
   Adjacency exchange-start threshold is 20
   Maximum number of next-hops supported in ECMP is 32
   Number of backbone neighbors is 0
   Graceful-restart is not configured
   Graceful-restart-helper mode is enabled
```

- This command displays the log entries of OSPFv3 LSA updates.

```
switch# show ipv6 ospf lsa-log
[22:11:02] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[21:31:02] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[20:56:22] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[20:18:12] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[19:47:22] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[19:13:22] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[18:39:32] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[18:06:32] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[17:26:42] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[16:48:42] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[16:13:12] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[15:36:52] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[15:03:32] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[14:27:52] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[13:52:02] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[13:15:02] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[12:39:42] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[12:00:02] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[11:27:22] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[10:53:22] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[10:17:12] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
[09:42:42] type RTR: 0.0.0.0 [13.13.13.13], event 2, backoff restarted, new hold value 1000 msecs
```

15.3.5.46 show ospfv3

The **show ospfv3** command displays the OSPFv3 configuration of OSPFv3 address family and routing process.

Command Mode

EXEC **Command Syntax**

show ospfv3 [access-list | border-routers | database | interface | ipv4 | ipv6 | lsa-log | neighbor | request-list | retransmission-list | spf-log | vrf]

Parameters

- ***no parameters*** displays the complete configuration of OSPFv3 address family and routing process.
- **access-list** displays the information of configured OSPFv3 access-list. Options include:
  - ***no parameters*** displays the information of all configured OSPFv3 access lists.
  - **WORD** displays the information of the specified access list.
  - **summary** displays the summary of all configured access lists.
- **border-routers** displays the information of configured OSPFv3 border and boundary routers. Options include:
  - ***no parameters*** displays the information of all configured OSPFv3 borders and boundary routers.
  - **vrf** displays the OSPFv3 borders and boundary routers information of the specified Virtual Routing and Forwarding (VRF).
- **database** displays the summary of database. Options include:
  - ***no parameters*** displays the complete summary of database.
  - **ipv4** displays the database information of link state ID.

  - **adv-router** displays the database information of advertising router link states.
  - **area** displays the database information filtered by area scope LSAs.
  - **as** displays the database information filtered by AS scope LSAs.
  - **database-summary** displays the count of LSAs in OSPFv3 database.
  - **detail** displays the detailed information of LSA.
  - **link** displays the database information filtered by link scoped LSAs.
  - **self-originate** displays the database information of self-originated link states.
  - **vrf** displays the VRF information in OSPFV3 database.
- **interface** displays the information of OSPFv3 interfaces. Options include:
  - ***no parameters*** displays the information of all OSPFv3 interfaces.
  - **Ethernet *eth_num*** displays the information of the specified Ethernet interface. The value ranges from ***1 to 24***.
  - **Loopback *lb_num*** displays the information of the specified loop back interface. The value ranges from ***0 to 1000***.
  - **Port-Channel *pc_num*** displays the interface or sub-interface information of the specified port channel. The interface and sub-interface values of port channel ranges from ***1-1000*** and ***1-2000***. ***1-4094*** respectively.
  - **Tunnel *t_num*** displays the information of the specified tunnel. The value ranges from ***0 to 255***.
  - **Vlan *vlan_num*** displays the information of the specified VLAN interface. The value ranges from ***1 to 4094***.
  - **vrf *vrf_name*** displays the information of the specified VRF.
- **ipv4** displays the IPv4 address family information.
- **ipv6** displays the IPv6 address family information.
- **lsa-log** displays the log entries of OSPFv3 LSA updates.
- **neighbor** displays the list of OSPFv3 neighbors.
- **request-list** displays the list of all OSPFv3 LSAs requested by a router.
- **retransmission-list** displays the list of all OSPFv3 LSAs waiting to be re-sent.
- **spf-log** displays the start-time, duration of completion, and reason of delay to calculate the OSPFv3 Sender Policy Framework (SPF).
- **vrf *vrf_name*** displays the information of specified VRF.

Examples

- This command displays the complete configuration of OSPFv3 address family and routing process.

```
switch# show ospfv3
OSPFv3 address-family ipv6
Routing Process "ospfv3" with ID 13.13.13.13 and Instance 0 VRF default
  FIPS mode disabled
  It is not an autonomous system boundary router and is not an area border
 router
  Minimum LSA arrival interval 1000 msecs
  Initial LSA throttle delay 1000 msecs
  Minimum hold time for LSA throttle 5000 msecs
  Maximum wait time for LSA throttle 5000 msecs
  Interface flood pacing timer 50 msecs
  It has 0 fully adjacent neighbors
  Number of areas in this router is 1. 1 normal, 0 stub, 0 nssa
  Number of LSAs 1
  Initial SPF schedule delay 0 msecs
  Minimum hold time between two consecutive SPFs 5000 msecs
  Current hold time between two consecutive SPFs 5000 msecs
  Maximum wait time between two consecutive SPFs 5000 msecs
  SPF algorithm last executed 3d23h ago
  No scheduled SPF
  Adjacency exchange-start threshold is 20
```

```
  Maximum number of next-hops supported in ECMP is 32
  Number of backbone neighbors is 0
  Graceful-restart is not configured
  Graceful-restart-helper mode is enabled
  Area 0.0.0.0
    Number of interface in this area is 0
    It is a normal area
    SPF algorithm executed 2 times
```

- This command displays the count of LSAs in OSPFv3 database.

```
switch# show ospfv3 database database-summary
OSPFv3 address-family ipv4
Routing Process "ospfv3" Instance 64 VRF default

LSA Type               Count
Router                     1
Network                    0
Inter Area Prefix          0
Inter Area Router          0
Summary Asex               0
Nssa                       0
Link                       0
Intra Area Prefix          0
Grace                      0
Total                      1

OSPFv3 address-family ipv6
Routing Process "ospfv3" Instance 0 VRF default

LSA Type               Count
Router                     0
Network                    0
Inter Area Prefix          0
Inter Area Router          0
Summary Asex               0
Nssa                       0
Link                       0
Intra Area Prefix          0
Grace                      0
Total                      0

ro301.02:05:02(config-router-ospfv3-af)#
```

- This command displays the start-time, duration of completion, and reason of delay to calculate the OSPFv3 SPF.

```
switch# show ospfv3 spf-log
OSPFv3 address-family ipv4
Routing Process "ospfv3" with ID 11.1.11.1 and Instance 64, VRF default
TIME          EVENT                    REASON
02:00:13.495  SPF ran for 0.064 ms
02:00:13.335  Scheduled after 0.000 ms Router LSA generation
01:59:55.499  SPF ran for 0.061 ms
01:59:54.604  Scheduled after 0.000 ms ECMP max nexthop cfg change
OSPFv3 address-family ipv6
Routing Process "ospfv3" with ID 11.1.11.1 and Instance 0, VRF default
TIME          EVENT                    REASON
02:00:13.495  SPF ran for 0.064 ms
02:00:13.335  Scheduled after 0.000 ms OSPF3 re-initialisation
01:59:55.499  SPF ran for 0.089 ms
01:59:54.603  Scheduled after 0.000 ms ECMP max nexthop cfg change
ro301.02:04:06(config-router-ospfv3-af)#
```

15.3.5.47 shutdown (OSPFv3)

The **shutdown** command disables OSPFv3 on the switch.

OSPFv3 is disabled by default on individual interfaces and enabled through **ipv6 ospf area** commands.

The **no shutdown** and **default shutdown** commands enable the OSPFv3 instance by removing the **shutdown** statement from the OSPFv3 block in ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Example

This command disables OSPFv3 activity on the switch.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# shutdown
switch(config-router-ospf3)# show active
ipv6 router ospf 9
   shutdown
switch(config-router-ospf3)#
```

15.3.5.48 timers lsa rx min interval (OSPFv3)

The **timers lsa rx min interval** command sets the minimum interval for accepting identical Link-State Advertisements (LSAs) from OSPFv3 neighbors.

The **no timers lsa rx min interval** and **default timers lsa rx min interval** commands restore the minimum interval to the default value of one second by removing the **timers lsa rx min interval** command from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Router-OSPFv3 Address-Family

IPv4/IPv6 Configuration

Command Syntax

timers lsa rx min interval ***lsa_time***

no timers lsa rx min interval

default timers lsa rx min interval

Parameter

lsa_time Minimum time (in milliseconds) after which the switch accepts an identical LSA from OSPFv3 neighbors. Value ranges from ***0 to 600000*** (ms). Default value is ***1000*** milliseconds (***1*** second).

Example

This command sets the minimum LSA arrival interval to ***10*** milliseconds.

```
switch(config)# router ospfv3
switch(config-router-ospfv3)# timers lsa rx min interval 10
```

```
switch(config-router-ospfv3)#
```

15.3.5.49 timers lsa tx delay initial (OSPFv3)

The **`timers lsa tx delay initial`** command sets the rate-limiting values for OSPFv3 Link-State Advertisement (LSA) generation.

The **`no timers lsa tx delay initial`** and **`default timers lsa tx delay initial`** commands restore the default LSA rate-limiting values by removing the **`timers lsa tx delay initial`** command from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Router-OSPFv3 Address-Family

IPv4/IPv6 Configuration

Command Syntax

`timers lsa tx delay initial` ***initial_delay min_hold max_wait***

`no timers lsa tx delay initial`

`default timers lsa tx delay initial`

Parameters

- ***initial_delay*** Initial delay in milliseconds to generate the first instance of LSAs. Value ranges from ***0 to 600000*** (ms). The default value is ***1000*** ms.
- ***min_hold*** Minimum hold interval availed in milliseconds between the generation of same LSA. Value ranges from ***1 to 600000*** (ms). The default interval is ***5000*** ms.
- ***max_wait*** Maximum hold interval availed in milliseconds between the generation of same LSA. Value ranges from ***1 to 600000*** (ms). The default interval is ***5000*** ms.

Example

These commands set the LSA transmission timers on the switch.

```
switch(config)# router ospfv3
switch(config-router-ospfv3)#timers lsa tx delay initial 5 100 20000
switch(config-router-ospfv3)#
```

15.3.5.50 timers spf delay initial (OSPFv3)

The purpose of SPF throttling is to delay Shortest Path First (SPF) calculations when network topology is changing rapidly. The **`timers spf delay initial`** command controls the intervals of SPF calculations in a switch. The command sets three values:

- **Initial delay:** Initial wait by a switch to calculate SPF after a topology change in a network that has been stable throughout the hold interval. Because a topology change often requires several link state updates to be sent, the initial delay is configured to allow the network to settle before the switch calculates SPF. If an additional topology change occurs during the initial interval, the SPF calculation still takes place after the initial delay period has expired and no other change is made to the throttle timers.
- **Hold interval:** This is an additional wait timer that reduces the frequency of SPF calculations during periods of network instability. If a network change occurs during the hold period, an SPF calculation is scheduled to occur when the hold interval expires. Subsequent hold intervals are doubled if further topology changes occur during a hold interval until either the hold interval reaches its configured maximum or no topology change occurs during the interval. If the next topology change occurs after the hold interval expires, the hold interval is reset to its configured value and the SPF calculation is scheduled to take place after the initial delay.

- **Maximum interval:** The maximum wait time of a switch after a topology change before performing an SPF calculation.

The **no timers spf delay initial** and **default timers spf delay initial** commands restore the default OSPFv3 SPF calculation intervals by removing the **timers spf delay initial** command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Router-OSPFv3 Address-Family

IPv4/IPv6 Configuration

Command Syntax

timers spf delay initial ***initial_delay hold_interval max_interval***

no timers spf

default timers spf

Parameters

- ***initial_delay*** Initial delay between a topology change and SPF calculation. Value ranges from ***0 to 65535000*** (ms). The default value is ***0*** ms.
- ***hold_interval*** Additional wait time after SPF calculation to allow the network to settle. If a topology change occurs during the hold interval, another SPF calculation is scheduled to occur after the hold interval expires. The next hold interval is doubled if topology changes occur during the hold interval. If doubling exceeds the maximum value, the maximum value is used instead. Value ranges from ***0 to 65535000*** (ms). The default value is ***5000*** ms.
- ***max_interval*** The maximum hold interval before a switch calculates SPF. Value ranges from **0 to 65535000** (ms). The default value is ***5000*** ms.

Example

These commands set the SPF timers on the switch.

```
switch(config)#router ospfv3
switch(config-router-ospfv3)#timers spf delay initial 5 100 20000
switch(config-router-ospfv3)#
```

15.3.5.51 timers

The **timers** command configures the minimum interval between the transmission of consecutive LS update packets in a network.

The **no timers** and **default timers** commands set the configured timer value to its default.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

timers {lsa | out-delay| pacing | throttle}

no timers {lsa | out-delay| pacing | throttle}

deault timers {lsa | out-delay| pacing | throttle}

Parameters

- **lsa** configures threshold for the retransmission of LSA. Option includes:
  - **arrival** configures the OSPF LSA arrival timer.

- **out-delay** configures the delay to flood router LSA in milliseconds. Option includes:
  - ***out-delay_time*** minimum interval in milliseconds between accepting the same LSAs. The value ranges from ***0 to 65000*** milliseconds. The default value is ***0***.
- **pacing** configures the OSPF packet pacing. Option includes:
  - **flood** configures the OSPF flood pacing.
- **throttle** configures OSPF throttle timers. Options include:
  - **lsa** configures threshold for the retransmission of LSA.
  - **spf** configures the time between SPF calculations.

Examples

- This command configures OSPFv3 flood pacing timer to ***50*** ms in the global OSPFv3 instance.

```
switch(config)# ipv6 router ospf 9
switch(config-router-ospf3)# timers pacing flood 50
switch(config-router-ospf3)# show ospfv3
Routing Process "ospfv3 9" with ID 13.13.13.13 and Instance 0 VRF default
  FIPS mode disabled
  It is not an autonomous system boundary router and is not an area border
 router
  Minimum LSA arrival interval 1000 msecs
  Initial LSA throttle delay 1000 msecs
  Minimum hold time for LSA throttle 5000 msecs
  Maximum wait time for LSA throttle 5000 msecs
  Interface flood pacing timer 50 msecs
  It has 0 fully adjacent neighbors
  Number of areas in this router is 1. 1 normal, 0 stub, 0 nssa
  Number of LSAs 1
  Initial SPF schedule delay 0 msecs
  Minimum hold time between two consecutive SPFs 5000 msecs
  Current hold time between two consecutive SPFs 5000 msecs
  Maximum wait time between two consecutive SPFs 5000 msecs
  SPF algorithm last executed 21d19h ago
  No scheduled SPF
  Adjacency exchange-start threshold is 20
  Maximum number of next-hops supported in ECMP is 32
  Number of backbone neighbors is 0
  Graceful-restart is not configured
  Graceful-restart-helper mode is enabled
  Area 0.0.0.0
    Number of interface in this area is 0
    It is a normal area
    SPF algorithm executed 2 times
```

- This command configures the OSPFv3 flood pacing timer to ***50*** ms in IPv4 address family.

```
switch(config)# router ospfv3
switch(config-router-ospfv3)# address-family ipv4
switch(config-router-ospfv3-af)# timers pacing flood 50
switch(config-router-ospfv3-af)# show ospfv3
OSPFv3 address-family ipv4
Routing Process "ospfv3" with ID 11.1.11.1 and Instance 64 VRF default
  FIPS mode disabled
  It is not an autonomous system boundary router and is not an area border
 router
  Minimum LSA arrival interval 1000 msecs
  Initial LSA throttle delay 1000 msecs
  Minimum hold time for LSA throttle 5000 msecs
  Maximum wait time for LSA throttle 5000 msecs
  Interface flood pacing timer 50 msecs
  It has 0 fully adjacent neighbors
```

```
Number of areas in this router is 1. 1 normal, 0 stub, 0 nssa
Number of LSAs 1
Initial SPF schedule delay 0 msecs
Minimum hold time between two consecutive SPFs 5000 msecs
Current hold time between two consecutive SPFs 5000 msecs
Maximum wait time between two consecutive SPFs 5000 msecs
SPF algorithm last executed 00:10:38 ago
No scheduled SPF
Adjacency exchange-start threshold is 20
Maximum number of next-hops supported in ECMP is 32
Number of backbone neighbors is 0
Graceful-restart is not configured
Graceful-restart-helper mode is enabled
Area 0.0.0.0
  Number of interface in this area is 0
  It is a normal area
  SPF algorithm executed 2 times
```

15.4 IS-IS

Intermediate System-to-Intermediate System (IS-IS) intra-domain routing information exchange protocol is designed by the International Organization for Standardization to support connectionless networking. This protocol is a dynamic routing protocol.

This chapter contains the following sections.

- IS-IS Introduction
- IS-IS Segment Routing
- IS-IS Graceful Restart
- IS-IS Dynamic Flooding
- IS-IS Configuration
- IS-IS Commands

15.4.1 IS-IS Introduction

IS-IS is a link-state protocol, which uses the Shortest Path First (SPF) algorithm. IS-IS and the OSPF protocol are similar in many aspects. As an Interior Gateway Protocol (IGP), IS-IS runs inside an Autonomous System (AS).

To enable IS-IS, you must instantiate an IS-IS routing instance and assign it to an interface. Arista IS-IS support includes IS-IS segment routing and IS-IS graceful restart.

15.4.2 IS-IS Segment Routing

Segment Routing (SR) provides a mechanism to simplify the definition of end-to-end paths within IGP topologies by encoding paths as sequences of topological sub-paths, called segments. The IS-IS protocol advertises these segments in four different ways: node segments, prefix segments, proxy-node segments, and adjacency segments.

Node segments represent a node in an IGP topology. A proxy segment are generally associated with an IP(v6) address received from a router that does not support IS-IS SR. Prefix segments represent an ECMP-aware shortest path to a prefix (or a node), as per the state of the IGP topology. Adjacency segments represent a hop over a specific adjacency between two nodes in IGP.

15.4.2.1 TI-LFA FRR using IS-IS Segment-Routing

Topology Independent Fast Reroute, or TI-LFA, uses IS-IS SR to build loop-free alternate paths along the post-convergence path. These loop-free alternates provide fast convergence in the range of sub-50 ms.

This section describes TI-LFA FRR using II-IS SR, including configuration instructions and command descriptions. Topics covered by this chapter include:

- TI-LFA FRR using IS-IS Segment-Routing Configuration
- TI-LFA FRR using IS-IS SR Commands
- Limitations

The (Point of Local Repair (PLR)- the router where TI-LFA is configured) PLR switches to these loop-free alternate backup paths in the event of a link down (link-protection) or BFD neighbor down (node-protection) event, protecting traffic destined to IS-IS SR node segments, adjacency segments, and anycast segments while the IGP converges and the post-convergence paths are computed. Anycast segment protection is restricted to those segments which are attached to prefixes with host mask (***/32*** for V4 address and ***/128*** for v6 address).



Note: Unlike node segments, anycast segments do not have the 'N' flag set described in section ***2.1.1.2*** of ***RFC8667***.

The following enhancements are available by release:

- ***EOS Release 4.22.1F*** adds support for TI-LFA backup paths that protect IS-IS SR labeled traffic corresponding to a node segment or adjacency segment on a transit router.
- ***EOS Release 4.23.1F*** adds support for TI-LFA backup paths that protect IS-IS SR tunnels.
- ***EOS Release 4.24.1*** adds support for protecting IS-IS SR labeled traffic corresponding to anycast segments.
- ***EOS Release 4.24.2F*** adds support for calculating TI-LFA backup paths that exclude the SRLG configured on the failing link.

Backup paths are only installed for IS-IS SR labeled routes and tunnels corresponding to node segments, adjacency segments, and anycast segments. When requesting node-protection, and no node-protecting LFAs are available, a link-protecting LFA is computed instead. TI-LFA FRR using IS-IS Segment-Routing is available with the multi-agent routing protocol model and the ribd routing protocol model.

Other traffic that resolves over IS-IS SR tunnels, such as LDP pseudowires, BGP LU tunnels, BGP IP routes, L2 EVPN, MPLS L3 VPN, and so on, are also protected by the TI-LFA tunnel that protects the resolving IS-IS SR tunnel.

15.4.2.1.1 TI-LFA FRR using IS-IS Segment-Routing Configuration

The following configuration tasks can be performed by Topology Independent Fast Reroute (TI-LFA FRR) using IS-IS Segment-Routing.

- Configuring Link or Node Protection on a Specific Interface
- Configuring a Local LFIB Convergence Delay for Protected Node or Adjacency Segments
- Making Locally-originated Adjacency Segments Backup Eligible
- Enabling SRLG Protection
- show ip route

15.4.2.1.2 TI-LFA FRR using IS-IS SR Commands

TI-LFA FRR using IS-IS SR Show Commands

- show ip route
- show isis interface
- show isis local-convergence-delay
- show isis segment-routing prefix-segments
- show isis segment-routing adjacency-segments
- show isis segment-routing tunnel
- show isis ti-lfa path
- show isis ti-lfa tunnel

- show mpls lfib route
- show tunnel fib

15.4.2.1.3 Limitations

- Backup paths are not computed for prefix segments that do not have a host mask (***/32*** for v4 and ***/128*** for v6).
- When TI-LFA is configured, the number of anycast segments generated by a node cannot exceed ***10***.
- Computing TI-LFA backup paths for proxy node segments is not supported.
- Backup paths are not computed for node segments corresponding to multi-homed prefixes. The multi-homing could be the result of them being anycast node segments, loopback interfaces on different routers advertising SIDs for the same prefix, node segments leaked between levels, and thus being seen as originated from multiple L1-L2 routers.
- Backup paths are only computed for segments that are non-ECMP.
- Only IS-IS interfaces that are using the point-to-point network type are eligible for protection.
- Link/node protection is only supported in the default VRF owing to the lack of non-default VRF support for IS-IS segment-routing.
- Backup paths are computed in the same IS-IS level topology as the primary path.
- Even with IS-IS GR configured, SSU, SSO, agent restart are not hitless events for IS-IS SR LFIB routes or tunnels being protected by backup paths.

15.4.3 IS-IS Graceful Restart

IS-IS Graceful Restart (GR) is a mechanism to prevent routing protocol re-convergence during a processor switchover or device downtime. Normally, when a router restarts, all the neighboring routers associated with that router detect that the device has gone down and remove routes from that neighbor. When the router restarts, the session is re-established and data transfer continues. During the restart, the removal and re-insertion of routes will result in data loss. This can be prevented by configuring graceful restart on the device.

When IS-IS is used as the interior gateway protocol (IGP), the following EOS features require nonstop forwarding (NSF) and support for the graceful restart from IS-IS:

- Smart Software Upgrade (SSU).
- planned Stateful SwitchOver (SSO) initiated by an operator for maintenance, or unplanned SSO due to failures on the active supervisor.
- RIB agent restart due to software failures.

With IS-IS Graceful Restart (GR) configured, a redundancy switchover from active to standby supervisor, or SSU, or restart of the IS-IS software (the RIB agent) should be a hitless event if the GR completes successfully. Neighboring routers will continue to forward traffic to the restarting router and traffic forwarding through the restarting router continues without loss. If GR is successful, the failure of a router should be completely transparent to network applications.

ISIS Graceful Restart (GR) is compatible with the following platforms:

- IS-IS GR with unplanned software restart is supported on all platforms.
- IS-IS GR with SSO is supported on modular dual-supervisor platforms.
- IS-IS GR with SSU is supported on platforms that support SSU.

15.4.4 IS-IS Dynamic Flooding

Dynamic Flooding allows IS-IS to scale to large, dense topologies such as Leaf-Spine topologies. In such topologies, legacy IS-IS can exhibit a congestive collapse due to the control plane load created by excessively redundant flooding.

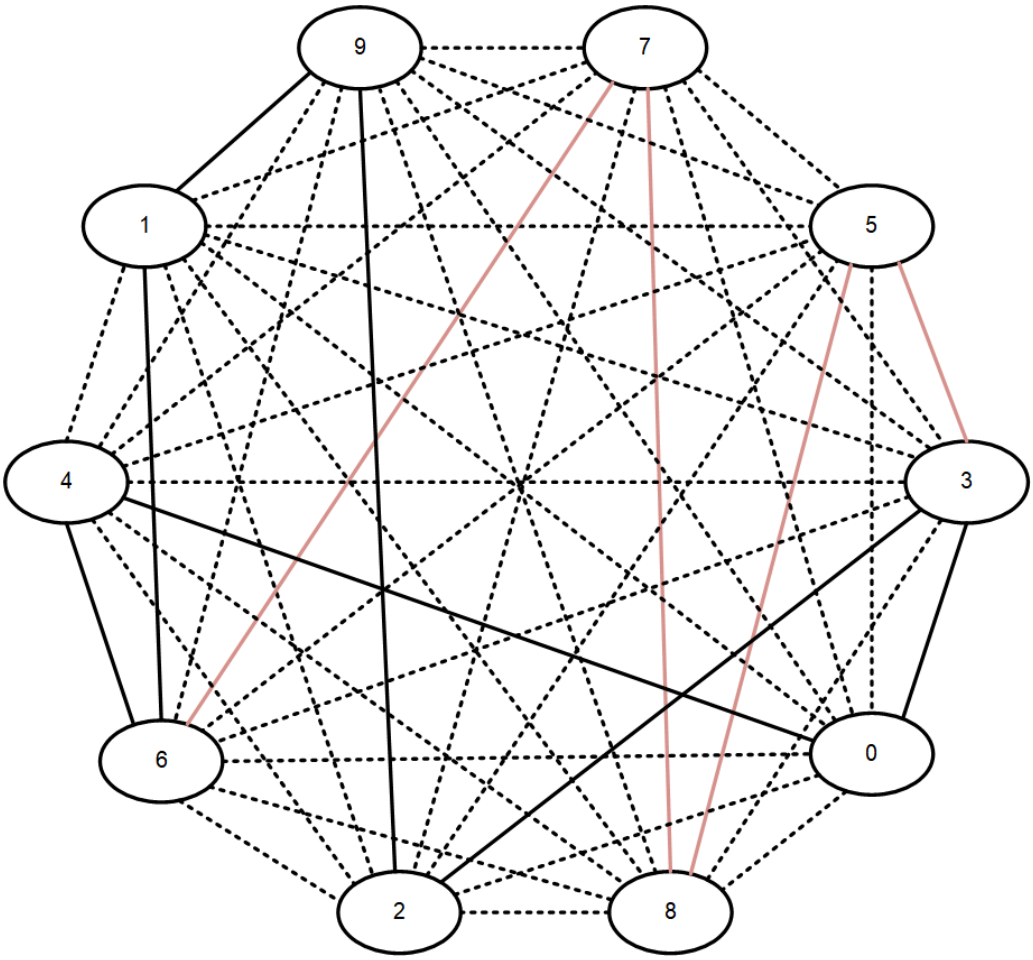
The concept in Dynamic Flooding is to dynamically compute a restricted topology for flooding (the **flooding topology**). Since this can be much smaller than the full physical topology, this can reduce the redundancy seen by each node, thereby reducing the control plane load and avoiding a congestive collapse.

To do this, first select one node within the IS-IS area as the **area leader**. Leverage the Designated Intermediate System (DIS) election algorithm for this, except instead of applying it to the neighbors on an interface, compute it across all of the nodes within the area.

The area leader is responsible for computing the flooding topology. This is distributed to the other nodes in the area through the Area System IDs TLV and the Flooding Path TLV.

All nodes within the area then flood only on the flooding topology.

Figure 15-11: Flooding Topology



A flooding topology on a dense graph. The flooding topology is shown by the solid lines. Dotted lines indicate non-flooding links.

In a dense topology, this can reduce the amount of flooding by an order of magnitude or more, with a resulting increase in scalability.

15.4.5 IS-IS Configuration

These sections describe IS-IS configuration tasks:

- Enabling IS-IS
- Configuring IS-IS Optional Global Parameters
- Configuring Optional IS-IS Interface Parameters
- Configuring IS-IS Segment Routing
- Configuring Redistribution of DHCP for IS-IS Agent (IPv6)
- Disabling IS-IS

- Configuring IS-IS Graceful Restart (GR)
- TI-LFA FRR using IS-IS Segment-Routing Configuration
- IS-IS Dynamic Flooding Configuration
- Relax Address-Family Check for IS-IS Adjacency
- Displaying IS-IS Information

15.4.5.1 Enabling IS-IS

To enable IS-IS, each of the following tasks must be performed.

- Enabling IS-IS Globally and Specifying an IS-IS Instance
- Configuring the Network Entity Title (NET)
- Setting the Address Family Configuration
- Enabling IS-IS on a Specified Interface

15.4.5.1.1 Enabling IS-IS Globally and Specifying an IS-IS Instance

The switch supports only one IS-IS routing instance per VRF. The routing instance uniquely identifies the switch to other devices. IS-IS configuration commands apply globally to the IS-IS instance.

The switch must be in router IS-IS configuration mode to run IS-IS configuration commands. The **router isis** command places the switch in router IS-IS configuration mode.

Example

These commands create an IS-IS routing instance named Osiris in the default VRF and place the switch in IS-IS configuration mode for that instance.

```
switch(config)# router isis Osiris
switch(config-router-isis)#
```

15.4.5.1.2 Configuring the Network Entity Title (NET)

After creating an IS-IS routing instance, configure the Network Entity Title (NET) with the **net** command. The NET defines the IS-IS area address and the system ID of the device.

Example

These commands configure the NET by specifying the IS-IS area address and the system ID of the device.

```
switch(config)# router isis Osiris
switch(config-router-isis)# net 49.0001.1010.1040.1030.00
```

15.4.5.1.3 Setting the Address Family Configuration

The **address-family** command enables the address families that IS-IS will route and places the switch in the configuration mode for that address family. The address families supported are IPv4 unicast and IPv6 unicast.

Example

These commands enable and enter the address family mode for IPv4 unicast.

```
switch(config)# router isis Osiris
switch(config-router-isis)# address-family ipv4 unicast
switch(config-router-isis-af)#
```

15.4.5.1.4 Enabling IS-IS on a Specified Interface

After enabling IS-IS globally, enable it on an interface with the **isis enable** command.

Example

These commands enable IS-IS on ***interface ethernet 4***.

```
switch(config-router-isis)# interface ethernet 4
switch(config-if-Eth4)#isis enable Osiris
```

15.4.5.2 Configuring IS-IS Optional Global Parameters

After globally enabling IS-IS, the following global parameters may be configured.

- Setting the Router Type
- Configuring Redistribution of Connected or Static Non-ISIS Routes
- Configuring Redistribution of Connected or Static non-ISIS Routes into Level-1 or Level-2
- Configuring Redistribution of BGP Routes into ISIS
- Setting the Overload Bit
- Configuring IS-IS MD5 Authentication
- Setting the SPF Interval
- Configuring IS-IS Segment Routing Global Adjacency-SID
- Enabling Logging for Peer Changes
- Setting the IS-IS hostname
- Configuring IS-IS Multi-Topology

15.4.5.2.1 Setting the Router Type

The **is-type** command sets the routing level for an IS-IS instance.

Example

These commands specify Level-2 for the IS-IS instance.

```
switch(config)# router isis Osiris
switch(config-router-isis)# is-type level-2
switch(config-router-isis)#
```

15.4.5.2.2 Configuring Redistribution of Connected or Static Non-ISIS Routes

The **redistribute (IS-IS)** command configures redistribution of connected or static non-ISIS routes.

Example

These commands redistribute connected routes into the IS-IS domain.

```
switch(config)# router isis Osiris
switch(config-router-isis)# redistribute connected
switch(config-router-isis)#
```

15.4.5.2.3 Configuring Redistribution of Connected or Static Non-ISIS Routes into Level-1 or Level-2

Non-ISIS routes can be exported into Level-1, Level-2, or both using a route map. By default, the routes are exported only to Level-2; to export to Level-1 or to both levels, configure the route map using the **set isis level** command. The Level-1 or Level-2 routes can also be filtered using the route maps match statement. The route map is then used when redistributing routes in ISIS with the **redistribute (IS-IS)** command.

Use the **show isis database detail** command to make sure that the route shows up in the exported level.

Examples

- The following commands configure a route map called ***rm*** to set the IS-IS level to Level-1, then use it to redistribute connected routes.

```
switch(config)# route-map rm
switch(config-route-map-rm)# set isis level level-1
switch(config-route-map-rm)# router isis osiris
switch(config-router-isis)# redistribute connected route-map rm
switch(config-router-isis)#
```

- The following command displays IS-IS database information and confirms that the level has been set to Level-1.

```
switch# show isis database detail
ISIS Instance: inst1 VRF: default
 ISIS Level 1 Link State Database
 LSPID Seq Num Cksum Life IS Flags
 1111.1111.1001.00-00 10 63306 751 L2 <>
 NLPID: 0xCC(IPv4) 0x8E(IPv6)
 Area address: 49.0001
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

15.4.5.2.4 Configuring Redistribution of BGP Routes into ISIS

The **redistribute bgp route-map** command redistributes the BGP routes from the specified route map into IS-IS. Only one route map can be specified; reissuing the command overrides any previous configuration.

The **no redistribute bgp** and **default redistribute bgp** commands disable BGP route redistribution from the specified domain by removing the **redistribute bgp** statement from ***running-config***.

The command is available in both ***router IS-IS*** configuration mode and the ***address-family*** submode. The command is rejected if configured in both modes at the same time. Issuing the **no** or **default** command in ***router IS-IS*** configuration mode has no effect on redistribution configured in the ***address-family*** submode.

Note: If the command is configured in an ***address-family*** submode, it only redistributes routes from that address family. If it is configured in router-ISIS mode, it applies to all enabled address families.

Examples

- These commands redistribute IPv4 BGP routes from the route map called ***bgp-to-isis-v4*** into the ISIS domain.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv4
switch(config-router-isis-af)# redistribute bgp route-map bgp-to-isis-v4
switch(config-router-isis-af)#
```

- These commands redistribute all BGP routes from the route map ***bgp-to-isis*** into ISIS.

```
switch(config)# router isis 1
switch(config-router-isis)# redistribute bgp route-map bgp-to-isis
```

15.4.5.2.5 Setting the Overload Bit

The overload bit is set in link state packets (LSPs) to signal that the switch is not available for forwarding transit traffic (for instance, during startup or when the switch is being taken down for maintenance). To set the

overload bit manually, use the **set-overload-bit** command without the **on-startup** option. To configure the switch to set the overload bit after a reboot, allowing routing protocols to converge before the switch is used for forwarding traffic, use the **set-overload-bit** command with the **on-startup** option. The overload bit will remain set set for the interval specified after startup.

Note: When using the **on-startup** option, the overload bit will remain set in LSPs until the IS-IS agent has been up for the configured interval. If the configured **on-startup** time is less than the actual IS-IS agent uptime, the command will be applied immediately.

In scenarios when Border Gateway Protocol (BGP) routes are resolved using an Interior Gateway Protocol (IGP), if the transit router reboots and becomes available again, the IGP will consider the transit router as an optimal path again. After rebooting, the transit router will blackhole traffic until the transit router learns the external destination reachability information via BGP.

Examples

- These commands configure the switch to set the overload bit in LSPs sent for ***120*** seconds after startup.

```
switch(config)# router isis Osiris
switch(config-router-isis)# set-overload-bit on-startup 120
switch(config-router-isis)#
```

- These commands configure the overload bit until BGP converges. If BGP fails to converge within the set timeout default period, then the overload bit gets cleared.

```
switch(config)# router isis Osiris
switch(config-router-isis)# set-overload-bit on-startup wait-for-bgp
switch(config-router-isis)# set-overload-bit on-startup wait-for-bgp timeout
 750
switch(config-router-isis)#
```

15.4.5.2.6 Configuring IS-IS MD5 Authentication

To configure authentication for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated, use the **authentication mode** and **authentication key** commands. To configure authentication on the interface, causing IS-IS Hellos to be authenticated, use the **isis authentication mode** and **isis authentication key** commands on the interface.

Two forms of authentication are supported by the IS-IS routing protocol: Clear-text authentication and MD5 authentication. The difference between the two forms of authentication is in the level of security provided. In the case of clear-text authentication, the password is specified as text in the authentication TLV, making it possible for an attacker to break authentication by sniffing and capturing IS-IS PDUs on the network. Arista recommends using the MD5 authentication.

HMAC MD5 authentication provides much stronger authentication by computing the message digest (on the IS-IS PDU contents) using the secret key to produce a hashed message authentication code (HMAC). Different modes of authentication can be specified on the interface, which authenticates IIH PDUs (IS-IS hello PDUs), and globally in the router IS-IS mode, in which the LSPs, CSNPs and PSNPs are authenticated. Area-wide and domain-wide authentication can be specified for L1 and L2 routers respectively.

Example

- These commands configure authentication for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated.

```
switch(config)# router isis 1
switch(config-router-isis)# authentication mode md5
switch(config-router-isis)# authentication key secret
switch(config-router-isis)#
```

- These commands configure authentication on the interface causing IS-IS hellos to be authenticated.

```
switch(config)# interface Ethernet 3/6
switch(config-if-Et3/6)# isis authentication mode text
switch(config-if-Et3/6)# isis authentication key 7 cAm28+9a/xPi04o7hjd8Jw==
switch(config-if-Et3/6)#
```

To maximize interoperability, Arista recommends using the same key in both interface mode and in the ***router isis*** mode.

15.4.5.2.7 Setting the SPF Interval

The SPF timer interval defines the maximum interval between two successive SPF calculations. IS-IS runs SPF calculations following a change in the network topology or the link-state database. The **spf-interval** command defines the following intervals:

- **Maximum wait interval:** The maximum time a switch will wait before running an SPF after a topology change.
- **Initial wait interval:** In a network that has been stable throughout the hold interval, this interval defines the initial wait time of a switch for performing an SPF calculation after a topology change. As several link-state updates must be sent after a topology change, the initial wait interval allows the network to settle before a switch computes an SPF. If the topology changes during an initial wait interval, an SPF is calculated after the initial wait interval expires and no further changes are made to throttle timers.
- **Hold time:** This interval delays SPF calculations during network instability. If the topology changes during a hold time, an SPF is computed when the hold time expires. Subsequent hold intervals are doubled up to the configured maximum wait interval for continuous topology changes. If the next topology change occurs after the hold interval expires, the hold interval is reset to its configured value and the SPF is computed after the initial wait interval.

Note: EOS does not support configuring topology-specific SPF timers in multi-topology deployments and IS-IS level-specific SPF timers.

Example

This command configures maximum wait interval, initial wait interval, and hold time to ***10*** seconds, ***2000*** ms, and ***1000*** ms respectively.

```
switch(config)# router isis inst1
switch(config-router-isis)# spf-interval 10 2000 1000
```

15.4.5.2.8 Configuring IS-IS Segment Routing Global Adjacency-SID

IS-IS Segment Routing (SR) supports global adjacency SIDs for point-to-point interfaces. The adjacency SID is configured as an index using the adjacency-segment command.

Global adjacency segments are represented using an index instead of actual MPLS labels. The index is an offset into the Segment Routing Global Block (SRGB) advertised by a router, resulting in an MPLS label. The default value of SRGB in EOS is Base: ***900000*** and Size: ***65536***.

The same index may be used to configure multiple interfaces so that MPLS forms an ECMP group, and the same index may be applied to IPv4 and IPv6 adjacencies.

Example

In this example, the global adjacency is configured on a point-to-point interface ethernet ***Et1***, with an index value ***10***.

```
switch(config-if-Et1)# adjacency-segment ipv4 p2p index 10 global
```

Displaying Adjacency SID Information

The command **show isis segment-routing adjacency-segments** displays the global adjacency SID value and other related information.

Examples

- In this example an interface is configured as follows:

```
interface ethernet1
 ip address 1.1.1.1/24
 ipv6 address 1000::1/64
 isis enable isis1
 isis network point-to-point
 adjacency-segment ipv4 p2p index 1 global
 adjacency-segment ipv6 p2p index 2 global
```

- The show output for the above interface configuration:

```
switch# show isis segment-routing adjacency-segments

System ID: 1000.0000.0002                       Instance: isis1
SR supported Data-plane: MPLS                   SR Router ID: 1.1.1.4
Adj-SID allocation mode: SR-adjacencies
Adj-SID allocation pool: Base: 100000     Size: 16384
Adjacency Segment Count: 2
Flag Descriptions: F: Ipv6 address family, B: Backup, V: Value
                   L: Local, S: Set

Segment Status codes: L1 - Level-1 adjacency, L2 - Level-2 adjacency, P2P -
Point-to-Point adjacency, LAN - Broadcast adjacency

Locally Originated Adjacency Segments
Adj IP Address     Local Intf  SID     SID Source   Flags                Type
-----------------  ----------  ------  -----------  ---------------      -------
1.1.1.2            Et1         1       Configured    F:0 B:0 V:0 L:0 S:0  P2P L1
fe80::1:ff:fe65:0  Et1         2       Configured    F:1 B:0 V:0 L:0 S:0  P2P L1


Received Global  Adjacency Segments
SID              Originator              Neighbor          Flags
---------        --------------------    ----------------  ---------
    0                 rtrmpls1           1000.0000.0002     F:0 B:0 V:0 L:0 S:0
```

15.4.5.2.9 Enabling Logging for Peer Changes

The **log-adjacency-changes (IS-IS)** command configures the switch to send syslog messages when it detects IS-IS neighbor adjacency state changes.

Example

These commands configure the switch to send a Syslog message when a neighbor goes up or down.

```
switch(config)# router isis Osiris
switch(config-router-isis)# log-adjacency-changes
switch(config-router-isis)#
```

15.4.5.2.10 Setting the IS-IS hostname

The **is-hostname** command configures the use of a human-readable string to represent the symbolic name of an IS-IS router. It also changes the output of IS-IS show commands, to show the IS-IS hostname in place of system IDs if the corresponding IS-IS hostname is known. However, syslogs still use IS-IS system IDs and not the IS-IS hostname.

By default if theres a hostname configured on the switch, it is used as the IS-IS hostname. It is also possible to deconfigure an assigned hostname for IS-IS using the **no is-hostname** command. When the IS-IS hostname is removed, the switch goes back to using the switchs hostname as the IS-IS hostname.

Examples

- These commands configure the IS-IS hostname to the symbolic name ***ishost1*** for the IS-IS router.

```
switch(config)# router isis inst1
switch(config-router-isis)# is-hostname ishost1
switch(config-router-isis)#
```

- These commands unconfigure the IS-IS hostname of the symbolic name ***ishost1*** for the IS-IS router.

```
switch(config)# router isis inst1
switch(config-router-isis)# no is-hostname ishost1
switch(config-router-isis)#
```

15.4.5.2.11 Configuring IS-IS Multi-Topology

The **multi-topology** command configures IS-IS Multi-Topology (MT) support (disabled by default), enabling an IS-IS router to compute a separate topology for IPv4 and IPv6 links in the network. With MT configured, not all the links in a network need to support both IPv4 and IPv6. Some can support IPv4 or IPv6 individually. The IPv4 SPF will install IPv4 routes using the IPv4 topology, and similarly, the IPv6 SPF will install IPv6 routes using the IPv6 topology. Without MT support, all links in an IS-IS network need to support the same set of address families.

When MT is enabled, and each link has a separate IPv4 metric and IPv6 metric.

The **isis ipv6 metric** command configures the IPv6 metric.

The **isis multi-topology** command configures the IPv4 or IPv6 address family individually on an interface with both IPv4 and IPv6 addresses.

The address families that are enabled on an interface are based on the global address families enabled in router IS-IS configuration mode, and the addresses configured on the interface. To enable a particular address family on an interface, it needs to have an address configured in that address family. In the case where both IPv4 and IPv6 address families are enabled in router IS-IS configuration mode, then if an interface has IPv4 and IPv6 addresses, both IPv4 and IPv6 address families are enabled on that interface. In the case of an interface with only an IPv4 address family, the IPv4 address family is enabled on that interface. Where an interface only has an IPv6 address family, the IPv6 address family is enabled on that interface. Finally, where only the IPv6 address family is enabled in router IS-IS config mode and MT is enabled, then the IPv6 address family is enabled on all interfaces which have an IPv6 address configured.

Examples

- These commands configure MT for the IS-IS router.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv6 unicast
switch(config-router-isis-af)# multi-topology
switch(config-router-isis-af)#
```

- These commands unconfigure MT for the IS-IS router.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv6 unicast
switch(config-router-isis-af)# no multi-topology
switch(config-router-isis-af)#
```

- These commands configure the IPv6 metric.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# isis ipv6 metric 30
switch(config-if-Et5/6)#
```

- These commands configure the IPv4 address family on an interface with both IPv4 and IPv6 addresses.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# isis multi-topology address-family ipv4 unicast
switch(config-if-Et1)#
```

- These commands configure the IPv6 address family on an interface with both IPv4 and IPv6 addresses.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# isis multi-topology address-family ipv6 unicast
switch(config-if-Et1)#
```

- These commands configure both the IPv4 and IPv6 address families on an interface.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# no isis multi-topology address-family unicast
switch(config-if-Et1)#
```

15.4.5.3 Configuring Optional IS-IS Interface Parameters

After globally enabling IS-IS, the following parameters may be configured on individual interfaces.

- Setting the Hello Packet Interval
- Configuring the Hello Multiplier for the Interface
- Configuring the IS-IS Metric
- Setting the LSP Transmission Interval
- Setting the IS-IS Priority
- Configuring an Interface as Passive
- Configuring BFD support for IS-IS for IPv4

15.4.5.3.1 Setting the Hello Packet Interval

The **isis hello-interval** command sets the time interval between the hello packets that maintain an IS-IS adjacency.

Example

These commands configure a hello interval of ***60*** seconds for ***interface ethernet 4***.

```
switch(config)# interface ethernet 4
switch(config-if-Et4)# isis hello-interval 60
switch(config-if-Et4)#
```

15.4.5.3.2 Configuring the Hello Multiplier for the Interface

The switch maintains the adjacency by sending/receiving hello packets. When receiving no hello packets from the peer within a time interval, the local switch considers the neighbors invalid.

The **isis hello-multiplier** command calculates the hold time announced in hello packets by multiplying this number with the configured **isis hello-interval**.

Example

- These commands configure a hello multiplier of ***5*** for ***interface ethernet 4***.

```
switch(config)# interface ethernet 4
switch(config-if-Et4)# isis hello-interval 60
switch(config-if-Et4)# isis hello-multiplier 5
switch(config-if-Et4)#
```

15.4.5.3.3 Configuring the IS-IS Metric

The **isis metric** command sets the cost for sending information over a specific interface. At present only wide metrics are supported.

Example

These commands configure a metric cost of ***30*** for sending information over ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# isis metric 30
switch(config-if-Et5)#
```

15.4.5.3.4 Setting the LSP Transmission Interval

The **isis lsp tx interval** command configures the minimum interval between successive LSP transmissions on an interface.

Example

This command sets the LSP transmission interval on interface ***interface ethernet 5*** to ***50*** milliseconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# isis lsp tx interval 50
switch(config-if-Et5)#
```

15.4.5.3.5 Setting the IS-IS Priority

The **isis priority** command determines which device will be the Designated Intermediate System (DIS). The device with the highest priority on the LAN will become the DIS.

Example

These commands configure a device priority of ***60*** on interface ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# isis priority 60
switch(config-if-Et5)#
```

15.4.5.3.6 Configuring an Interface as Passive

A passive IS-IS interface does not send or receive IS-IS packets and will not form adjacencies, but is still included in LSP advertisements, making its IP address visible to the IS-IS domain. To configure an IS-IS interface as passive, use the **isis passive** command in interface configuration mode or the **passive (IS-IS)** command in router IS-IS configuration mode.

Examples

- These commands configure ***interface ethernet 10*** as a passive interface.

```
switch(config)# interface ethernet 10
switch(config-if-Et10)# isis passive
switch(config-if-Et10)#
```

- These commands also configure ***interface ethernet 10*** as a passive interface.

```
switch(config)# router isis Osiris
switch(config-router-isis)# passive ethernet 10
switch(config-router-isis)#
```

15.4.5.3.7 Configuring BFD support for IS-IS for IPv4

The **isis bfd** and **bfd all-interfaces** commands configure Bidirectional Forwarding Detection (BFD). BFD is supported for both IS-IS IPv4 and IPv6 routes.

Examples

- These commands enable BFD (for the IPv4 address family) for all the interfaces on which IS-IS is enabled. By default, BFD is disabled on all interfaces.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv4
switch(config-router-af)# bfd all-interfaces
switch(config-router-af)#
```

- These commands enable BFD on an IS-IS interface.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# isis bfd
switch(config-if-Et5/6)#
```

15.4.5.4 Configuring IS-IS Segment Routing

Global IS-IS Segment Routing (IS-IS SR) commands are accessed in Segment-Routing MPLS mode, under the router IS-IS configuration mode. Interface-specific IS-IS SR commands are accessed in interface configuration mode.

15.4.5.4.1 Starting the MPLS Agent

The Routing Information Base (RIB) or IS-IS agent provides IS-IS segment routing, but the actual installation of LFIB entries pertaining to SR information provided by IS-IS is handled by the MPLS agent in EOS, which is disabled by default. To enable the MPLS agent, use the following commands.

Note: IP(v6) routing must be enabled as a prerequisite.

Example

The following commands enable IP routing and the MPLS agent on the switch.

```
switch(config)# ip routing
switch(config)# mpls ip
switch(config)#
```

15.4.5.4.2 Enabling IS-IS SR

By default, IS-IS SR is disabled. You must enable it explicitly by issuing the **no** form of the **shutdown (IS-IS SR)** command in Segment-Routing MPLS configuration mode.

Example

The following commands enable IS-IS SR.

```
switch(config)#router isis instance1
switch(config-router-isis)#segment-routing mpls
switch(config-router-isis-sr-mpls)#no shutdown
switch(config-router-isis-sr-mpls)#
```

15.4.5.4.3 Disabling IS-IS Segment Routing

To administratively disable IS-IS SR, issue the **shutdown (IS-IS SR)** command in Segment-Routing MPLS configuration mode. To disable ***isis sr*** and delete all ***isis sr*** configuration, issue the **no segment-routing mpls** command in ***router isis*** configuration mode.

Example

- The following commands administratively disable ***router isis***.

```
switch(config)# router isis instance1
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)# shutdown
switch(config-router-isis-sr-mpls)#
```

- The following commands disable ***router isis*** and delete all ***router isis*** configuration.

```
switch(config)# router isis instance1
switch(config-router-isis)# no segment-routing mpls
switch(config-router-isis)#
```

15.4.5.4.4 SRGB (Segment Routing Global Range)

The global segments such as Prefix-SID, Node-SID, Proxy-node-SID are represented using indices of actual MPLS labels. These indices are offset on the SRGB advertised by a router to derive the respective MPLS label. The default value of SRGB in EOS is Base: ***900000***, Size: ***65536***. In other words, the labels that any global segment could represent is between ***900000***-***965535***. The MPLS label range is categorized and reserved into pools based on the applications using these labels. The default values of label ranges in these pools are:

- Dynamic Global Range--(***100000***) (***262144***)
- IS-IS SR Global Range -- (***900000***) (***65536***)
- Static Global Range -- (***16***) (***99984***)

Note: SRGB can be configured to fit in different MPLS ranges as long as it does not fall under an MPLS range already assigned for usage by other applications.

Example

```
switch(config)# mpls label range isis-sr 900000 65536
```

15.4.5.4.5 IS-IS Maximum LSP Size

The IS-IS maximum LSP size provides the ability to configure the maximum LSP size that the IS-IS protocol accepts and sends.The default value of LSP size is ***9000***. The **lsp size maximum** command configures maximum size of an LSP that is sent or received. The default LSP maximum size is ***9000***. The minimum value is ***512***.

Example

```
switch(config)# lsp size maximum 400
```

The **no lsp size maximum** and **default lsp size maximum** commands remove the specified **lsp size maximum** command from ***running-config***.

```
switch(config)# no lsp size maximum
```

```
switch(config)# default lsp size maximum
```

15.4.5.4.6 Configuring Node-SID

Node segments are indices associated with routers within an IS-IS SR domain. This is done by associating node-segments with prefix mask length ***/32*** (IPV4) or ***/128*** (IPV6) addresses. Node segments are carried as sub-TLVs (type-length-value) in IP reachability TLVs for the prefixes with which these segments are associated. Node segments are configured on IS-IS enabled Loop-back interface(s) as shown in the example below.

Examples

- The following commands are used to associate a node-segment with an IPv4 address.

```
switch(config)# int loopback 1
switch(config-if-Lo1)# ip address 21.1.1.1/32
switch(config-if-Lo1)# node-segment ipv4 index 5
```

- The following commands are used to associate a node-segment with an IPv6 address.

```
switch(config)# int loopback 1
switch(config-if-Lo1)# ipv6 add 2000::24/128
switch(config-if-Lo1)# node-segment ipv6 index 5
```

- The following example shows a warning thrown at the CLI when a ***/32*** or ***/128*** address is not configured on the interface.

```
switch(config)# int loopback 1
switch(config-if-Lo1)# ip address 21.1.1.1/24
switch(config-if-Lo1)# node-segment ipv4 index 1
! /32 IPv4 address is not configured on the interface
```

- The following command removes the node-segment from IS-IS SR from an interface.

```
switch(config-if-Lo1)# no node-segment ipv4 index 1
```

15.4.5.4.7 Configuring Prefix-SIDs

Prefix segments are associated with any IS-IS prefix a router is originating an IP Reachability TLV for. These segments are carried as sub-TLVs in IP Reachability TLVs of the prefixes with which these segments are associated. Prefix segments are configured under segment-routing MPLS configuration mode in IS-IS.

Note: The configured prefix segment is effective, only if, the prefix for which a prefix-SID configured becomes a part of IS-IS by enabling IS-IS on interfaces, or by redistribution from other protocols etc.

Example

The following commands are used to associate a prefix segment with an IPv4 address with index value of ***50***.

```
switch(config)# router isis instance1
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)# prefix-segment 1.1.1.0/24 index 50
```

15.4.5.4.8 Configuring Proxy-Node SIDs

Node segments represent a device (node) by attaching a segment (index) with a /32, /128 prefix which generally is configured on a loopback interface. There are routers which do not support segment routing, and there might be a situation where it is required to assign node identifiers on such routers. To overcome this shortfall, a router that supports IS-IS SR is made to proxy by configuring a proxy-node-SID for a IS-IS prefix originating from the router that does not support IS-IS SR.

Example

A proxy-node SID associates a ***/32*** or a ***/128*** route with an SID as shown below.

```
switch(config)# router isis instance1
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)# proxy-node-segment 1.1.1.0/32 index 50
```

Although the general use case is to configure a proxy node segment on a router that is not originating the prefix with which we want to associate the proxy-node SID, it is not prohibited to configure one for self-originated prefixes.

Configuring proxy-node-SIDs enables a router to send out a Binding-SID TLV with details pertaining to the prefix and SID.

Note: A Binding-SID can carry a range of prefixes and an associated range of SIDs, but at present the EOS does not support to configure such ranges with one binding segment TLV in IS-IS SR. However, we do process ranges of prefixes and SIDs if received from devices that support such configurations.

15.4.5.4.9 Configuring Anycast-SID

An Anycast-SID is a prefix segment that identifies a set of routers and not a specific router. It enforces the ECMP-aware shortest-path forwarding towards the closest node of the anycast set.

An example of such an anycast group could be a set of routers A1, A2, A3, and A4 where at least one router of A1, A2, A3, and A4 advertises the prefix SID corresponding to the anycast address (which can be a prefix originating on all of A1, A2, A3 and A4 a loop-back address, maybe).

In general use case, all the routers of the anycast group would have the same prefix-SID configured for the anycast IP address present on them.

Note: That for Anycast-SID to work as expected, the SRGB on the members of the anycast group should be same.

15.4.5.4.10 Configuring router-ID

A router that support IS-IS SR need to advertise its SR data-plane capability and the range of MPLS label values it uses for segment routing, this is advertised by inserting SR-Capability sub-TLV in the Router Capabilities TLV.

A Router Capability TLV is now sent in IS-IS LSPs when Segment routing is enabled and it is necessary for a Router Capability TLV to carry a router-ID. This router-ID could be configured in EOS under the segment routing MPLS configuration mode. If no router-ID is configured, the router automatically picks up the highest IPv4 address configured on the router for an router-ID.

15.4.5.4.11 Configuring IS-IS Static Adjacency SID

Adjacency segments for IS-IS adjacencies are statically configured on the switch, so that these values are preserved even when the switch restarts. Static adjacency segments are configured per address family on any interface (including Port-Channel, VLANs and SVIs). They are configured and advertised as labels.

These are the few points to be considered while configuring the static adjacency SIDs:

- The same label can be configured on multiple interfaces so that MPLS can form ECMP, the same value can be applied to IPv4 and IPv6 adjacency.
- Static adjacency SID is applied only to p2p interface, and has local scope. When interface type changes to LAN, then dynamic adjacency SID is assigned.
- When Static adjacency SIDs are configured, then simply replace dynamic adjacency SIDs which are advertised to other routers and installed in the local LFIB.
- Static adjacency SID is applied regardless of Adjacency Segment Allocation Mode.

- When Static adjacency SID is disabled, then normal rules for dynamic adjacency SID is applied (it automatically applies a value based on Adjacency Segment Allocation Mode as described in IS-IS Segment Routing TOI document).

Example

```
switch(config-if-Et1)# adjacency-segment ipv4 p2p index 50 global
```

They can be a label (local) or index (global) and we can assign multiple adjacency segments per link.

Where label-value must be within the SR Local Block (SRLB) that can be found in the output of **show mpls label range** command as shown.

```
switch# show mpls label range
Start     End       Size      Usage
------------------------------------------------
0         15        16        reserved
16        99999     99984     static mpls
100000    362143    262144    free (dynamic)
362144    899999    537856    unassigned
900000    965535    65536     isis-sr
900000    965535    65536     bgp-sr
965536    1031071   65536     srlb
1031072   1036287   5216      unassigned
1036288   1048575   12288     l2evpn
```

15.4.5.4.12 Configuring Adjacency Segment Label Range

Adjacency Segments are MPLS labels assigned to IS-IS adjacencies.These labels are shared with other routers in the domain by adding them in adjacency-SID sub-TLVs which are inserted in neighbor Reachability TLVs in IS-IS.

The MPLS labels (adjacency segments) are incrementally allocated to adjacencies, as the transition to Up state, from a adjacent set of MPLS labels pre-allocated by MPLS agent. This label range extends from ***100000*** to ***116383*** (base: ***100000***, size: ***16384***) by default. This could be changed by the following configuration:

Example

```
switch(config)# mpls label range dynamic 200000 131072
```

The dynamic label pool is shared between LDP and IS-IS SR Adjacency Segments.

15.4.5.4.13 Configuring Adjacency Segment Allocation Mode

Adjacency Segments are allocated to all IS-IS adjacencies based on the IS-IS routers that have advertised IS-IS SR capability or to none of the adjacencies. The command adjacency-segment allocation is used to configure this under the ***segment-routing mpls*** configuration mode.

The default behavior is to allocate adjacency segments to adjacencies of SR supporting devices.

Example

```
switch(config-router-isis-sr-mpls)# adjacency-segment allocation sr-peer
```

15.4.5.4.14 Adjacency Segment Persistence across Link Flaps

Adjacency segments are allocated to IS-IS adjacencies based on configured adjacency segment allocation mode mentioned above.

If an adjacency that has been allocated label L goes down, L is reserved for this adjacency for a duration of ***3600*** seconds from the time of the adjacency down event. Only the adjacency that owned this label before going down could reclaim label L in this duration.

15.4.5.4.15 Troubleshooting IS-IS Segment Routing

- The **`show tech-support ribd`** command has a section starting with the string SR Book Keeper which has extensive information on state of IS-IS SR on the router.
- In-case, if IS-IS SR is configured but SR related TLVs/sub, but, TLVs are not being sent in IS-IS LSPs.
  - Ensure that MPLS has been enabled (MPLD IP) enabled.
  - Check if segment routing is administratively shut down.
  - A segment might have been configured for a prefix not yet being advertised in IS-IS.
- In case, if Adjacency Segments are not being advertised.
  - Check if the adjacency segment mode is correctly set.
  - Adjacency Mode is set to all SR supported interfaces (default setting) and the peer does not support SR.
- Generally, it is good to not have same prefix with different indices or same index with different prefixes. There are CLI prohibitions that ensure that a router is not sending out conflicting sets of prefixes and associated SIDs. As there is possibility of receiving conflicting prefix-segments from other devices, there are ways to resolve the following three types of conflicts: prefix+SID conflict, SID conflict and prefix conflict.
  - Prefix+SID Conflict: When there are two prefix segments which have both the prefix and SID have same values, the one from the higher system ID is chosen for LFIB processing.
  - Prefix Conflict: If the two prefix segments which have same Prefix are from two different system than the one from higher system ID is chosen. If they are originated from same system ID than we choose the prefix segment of smaller SID.
  - SID Conflict: If the two prefix segments which have same SID are from two different system than the one from higher system ID is chosen. If they are originated from same system than the one which is of smaller prefix length is chosen. If prefix length is also same than the one with smaller address is chosen.

For a given prefix, if both a proxy-node segment and prefix-SID are received, the prefix-SID advertised is preferred while the proxy-node segment is ignored.

The **`show tech-support ribd`** displays detail information about IS-IS SRs internal state, and more information on conflicts and chosen active segments could be found under the SR Book Keeper section of **`show tech-support ribd`** command as shown.

```
Received Prefix Segments:
-----------------------------------------------------------------
 Prefix    | Value | Index/Label | Type   | SystemID      | spfgen
 * - Active, # - Duplicate pfx, + - duplicate SID
-----------------------------------------------------------------
*1.0.3.0/24    3      Index        Prefix   1111.1111.1002   0
*1.0.5.1/32    0      Index        Node     1111.1111.1002   0
*1.0.6.1/32    2      Index        Node     1111.1111.1003   39
*1.0.7.1/32    14     Index        Node     1111.1111.1001   39
#1.0.7.1/32
```

10 Index Proxy ***1111.1111.1003 39***

15.4.5.5 Configuring Redistribution of DHCP for IS-IS Agent (IPv6)

The **`redistribute dhcp`** command redistributes DHCPv6 routes in IS-IS when using multi-agent routing protocol mode.

The **redistribute dhcp** command enables DHCP route redistribution in IS-IS when using the multi-agent routing protocol mode.

- These commands redistribute IPv6 DHCP routes into the ISIS domain.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv6
switch(config-router-isis-af)# redistribute dhcp
switch(config-router-isis-af)#
```

- The following command shows the DHCPv6 routes distributed into IS-IS.

```
switch(config)# show isis database detail
IS-IS Instance: inst1 VRF: default
  IS-IS Level 1 Link State Database
    LSPID                 Seq Num   Cksum  Life  IS Flags
    1111.1111.1001.00-00  10        19778  1101  L1 <>
      ...
      Reachability (MT-IPv6): 3ffe:701:ffff:101::10/128 Metric: 0 Type: 1 Up
      ...
```

15.4.5.6 Disabling IS-IS

An IS-IS instance can be shut down globally or can be disabled on individual interfaces.

The **shutdown (IS-IS)** command shuts down an IS-IS instance globally.

Example

These commands disable IS-IS globally without modifying the IS-IS configuration.

```
switch(config)# router isis Osiris
switch(config-router-isis)# shutdown
switch(config-router-isis)#
```

The **no isis enable** command disables IS-IS on an interface.

Example

These commands disable IS-IS on interface ***interface ethernet 4***.

```
switch(config-router-isis)# interface ethernet 4
switch(config-if-Eth4)# no isis enable
```

15.4.5.7 Configuring IS-IS Graceful Restart (GR)

By default, IS-IS graceful restart is disabled. Use the graceful-restart command to configure graceful restart on an IS-IS router. By default IS-IS graceful-restart-helper functionality is enabled, and to disable it use no graceful-restart-helper command.

Examples

In this example IS-IS graceful restart is configured with t2 wait time of ***30*** seconds for ***level-1*** routes.

```
switch(config)# router isis 1
switch(config-router-isis)# graceful-restart t2 level-1 30
```

t2 is the maximum wait time for the LSP database to synchronize (SPF computation is not done while t2 is running). t2 can be configured for either Level-1 or Level-2 through the CLI. The default value is ***30*** seconds, and the allowed configuration range is ***5*** to ***300*** seconds.

Example

In this example an ISIS graceful restart is configured with restart-hold-time of ***50*** seconds.

```
switch(config)# router isis 1
switch(config-router-isis)# graceful-restart restart-hold-time 50
```

In case of a planned restart, the hold time advertised by the IS-IS router prior to restart should be greater than the time for which the router is expected to be offline. Otherwise, neighboring routers will bring down the adjacency before the restarting router has a chance to send a restart request in its hello packet, which may result in traffic loss.

In case of ASU2, the IS-IS router instance will advertise a hello hold time of ***restart-hold-time*** on those interfaces for which the configured hold time is less than ***restart-hold-time***. This is done just before the router restarts.

Note: Once the router has restarted, the routers advertised hello hold time will depend on the hello-interval and hello-multiplier configuration on each interface as before. By default, the ***restart-hold-time*** is disabled.

For Graceful Restart to be successful, the hold time advertised by the router should be greater than the time it takes for Graceful Restart to complete. If the restarting router is DIS, hold time advertised is 1/3rd of the configured value (default is 9s). We recommend increasing the hold time for the DIS to a higher value before a planned restart; otherwise, it may result in traffic loss.

15.4.5.8 TI-LFA FRR using IS-IS Segment-Routing Configuration

The following configuration tasks can be performed by Topology Independent Fast Reroute (TI-LFA FRR) using IS-IS Segment-Routing.

- Configuring Link or Node Protection on a Specific Interface
- Configuring a Local LFIB Convergence Delay for Protected Node or Adjacency Segments
- Making Locally-originated Adjacency Segments Backup Eligible
- Enabling SRLG Protection
- show ip route

15.4.5.8.1 Configuring Link or Node Protection on a Specific Interface

To enable link or node protection for node segments and Adjacency segments learned on a specific IS-IS interface, use the following command in the interface configuration mode.

```
switch(config-if-Et1)# [no|default] isis fast-reroute ti-lfa mode
 {link-protection|node-protection|disabled} [level-1|level-2]
```

The interface TI-LFA configuration inherits the address-family sub-mode configuration by default.

On an L1-L2 router, the [**level-1**|**level-2**] optional keyword in both the router IS-IS address-family sub-mode and interface configuration mode CLIs is used to restrict protection to node segments and Adjacency segments learned through either Level-1 or Level-2 topologies only.

15.4.5.8.2 Configuring a Local LFIB Convergence Delay for Protected Node or Adjacency Segments

The Point of Local Repair (PLR) switches to the TI-LFA backup path on link failure or BFD neighbor failure but switches back to the post-convergence path once the PLR computes SPF and updates its LFIB. This sequence of events can lead to micro-loops in the topology if the PLR converges faster than other routers along the post-convergence path. So a configuration option is provided to apply a delay, after which the

LFIB route being protected by the TI-LFA loop-free repair path will be replaced by the post-convergence LFIB route.

To configure a convergence delay only to LFIB routes that are being protected, the following command is used either in the router IS-IS mode or the ***router isis address-family*** sub-mode. A default of ***10*** seconds is used when using the command without an explicitly specified delay.

```
switch (config-router-isis-af)# timers local-convergence-delay [delay_in_milliseconds]
protected-prefixes
```

15.4.5.8.3 Making Locally-originated Adjacency Segments Backup Eligible

The PLR computes backup paths for an adjacency segment only if the Adjacency SID sub-TLV has the B-flag (backup flag) set.

To set the B-flag in originated Adjacency SID sub-TLVs corresponding to adjacency segments dynamically allocated on the router, the following command is used in the ***segment-routing mpls*** sub-mode in the ***router isis*** mode.

```
switch(config-router-isis-sr-mpls)# adjacency-segment allocation [all-interfaces | sr-peers]
```

To set the B-flag in originated Adjacency SID sub-TLVs corresponding to adjacency segments statically configured on the router, the following command is used in the interface configuration mode.

```
switch(config-if-Et1)# adjacency-segment [ipv4 | ipv6] p2p [multiple][label label | index index] backup-eligible
```

backup-eligible is the newly introduced optional keyword in both the CLIs mentioned above that controls the setting of the B-flag in the Adjacency SID sub-TLV.

15.4.5.8.4 Enabling SRLG Protection

To enable SRLG protection on all interfaces, use the **`fast-reroute ti-lfa srlg`** command. This command is used in addition to configuring link-protection or node-protection. If SRLG protection is enabled, the backup paths are computed after excluding all the links that share the same SRLG with the active link that is being used by all prefix segments and adjacency segments.

```
switch(config-router-isis-af)# fast-reroute ti-lfa srlg [strict]
```

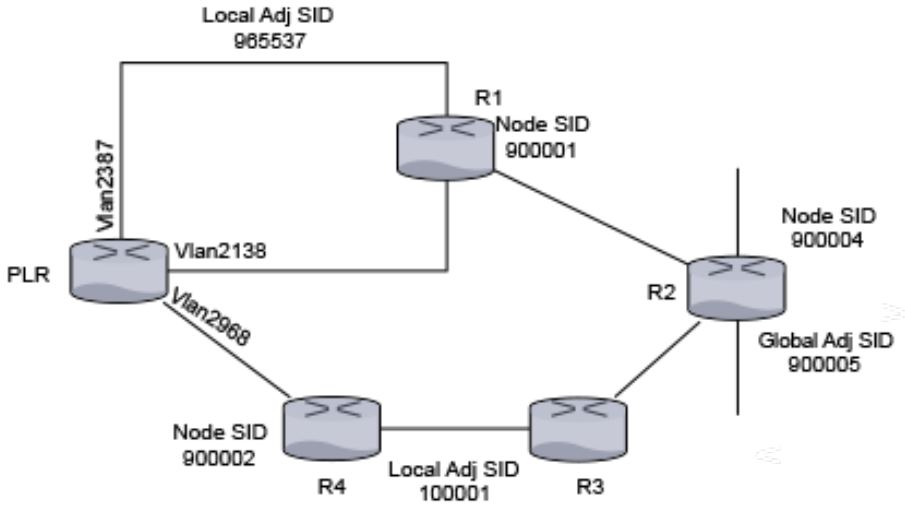
If the optional argument **strict** is configured, the backup path is only programmed if a backup path that excludes all the SRLGs configured on the primary interface. If the keyword is not provided and an SRLG excluding path is not available, TI-LFA programs the backup path that excluded the maximum number of SRLGs possible.

To selectively disable SRLG protection on an interface, use the **`isis [ipv4|ipv6] fast-reroute ti-lfa srlg disabled`** command. This is useful if SRLG protection is enabled globally for all interfaces but needs to be selectively disabled for a specific interface.

```
switch(config-intf-et1)# isis [ipv4 | ipv6] fast-reroute ti-lfa srlg disabled
```

Sample Configuration

Figure 15-12: Sample Configuration



The above topology will be used to demonstrate the configuration and show command output.

The above topology is used to demonstrate the configuration and show command output. You will see the backup paths that the PLR computes to protect the node segments of R1 and R2, the global adjacency segment on R2, and the local adjacency segment on the ***vlan 2387*** on the PLR.

Here is a snippet of the configuration on the PLR.

```
switch(config)# interface vlan 2138
switch(config-if-Vl2138)# ip address 10.1.1.1/24
switch(config-if-Vl2138)# isis enable inst1
switch(config-if-Vl2138)# isis metric 11
switch(config-if-Vl2138)# isis network point-to-point

switch(config)# interface vlan2387
switch(config-if-Vl2138)# ip address 10.1.2.1/24
switch(config-if-Vl2138)# isis enable inst1
switch(config-if-Vl2138)# isis network point-to-point
switch(config-if-Vl2138)# adjacency-segment ipv4 p2p label 965537
 backup-eligible

switch(config)# interface vlan2968
switch(config-if-Vl2968)# ip address 10.1.3.1/24
switch(config-if-Vl2968)# isis enable inst1
switch(config-if-Vl2968)# isis network point-to-point
switch(config-if-Vl2968)# isis fast-reroute ti-lfa mode disabled

…

switch(config)# router isis inst1
switch(config-isis)# net 49.0001.1111.1111.1001.00
switch(config-isis)# router-id ipv4 252.252.1.252
switch(config-isis)# is-type level-2
```

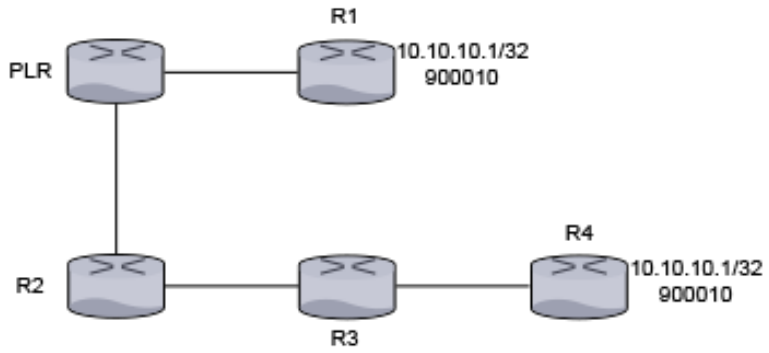
```
switch(config-isis)# timers local-convergence-delay 5000 protected-pre
fixes
   !
switch(config-isis)# address-family ipv4 unicast
switch(config-isis-af)# fast-reroute ti-lfa mode node-protection
   !

switch(config-isis)# segment-routing mpls
switch(config-isis-sr-mpls)# no shutdown
switch(config-isis-sr-mpls)# adjacency-segment allocation sr-peers
 backup-eligible
!
end
```

The protection of anycast segments does not need any new configuration. The above configuration enables protection of anycast segments.

To demonstrate the protection of anycast segments consider the following topology.

Figure 15-13: Topology Number 2



R1 and R4 are originators of the host prefix 10.10.10.1/32 and advertise prefix segment 900010. Note that this should be configured as a prefix segment and not a node segment.

R1 and R4 are originators of the host prefix ***10.10.10.1/32*** and advertise prefix segment ***900010***. This must be configured as a prefix segment and not a node segment.

R1 and R4's configuration should look similar to the following:

```
switch(confg)# router isis inst1
switch(config-router-isis)# interface Loopback0
switch(config-if-Lo0)# ip address 10.10.10.1/32
switch(config-if-Lo0)# isis enable inst1

!
...
switch(confg)# router isis inst1
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)# prefix-segment 10.10.10.1/32 index
 10
!
```

The prefix in the prefix-segment command must belong to an interface enabled with IS-IS or must be an active route in the RIB of another protocol redistributed into IS-IS.

When the link or node protection is configured on the PLR, then the primary path to the segment ***900010*** is PLR - R1 and the backup path is PLR - R2 - R3 - R4. In other words, the destination in the backup path is the segment originated by R4 as the segment originated by R1 and is not reachable when link PLR-R1 or the node R1 goes down.

15.4.5.8.5 show ip route

When services like LDP pseudowires, BGP LU, L2 EVPN, or L3 MPLS VPN use IS-IS SR tunnels as an underlay, these services are automatically protected by TI-LFA tunnels that protect the IS-IS SR tunnels. The **`show ip route`** command displays the hierarchy of the overlay-underlay-TI-LFA tunnels.

```
switch# show ip route
 B        2001:db8:3::/48 [200/0]
           via 2002::b00:301/128, IS-IS SR tunnel index 3, label
 122697
  via TI-LFA tunnel index 5, label imp-null(3)
  via fe80::200:76ff:fe03:0, Ethernet26/1, label imp-null(3)
  backup via fe80::200:76ff:fe01:0, Ethernet30/1, label 900002 900003
```

15.4.5.9 IS-IS Dynamic Flooding Configuration

Configure Dynamic flooding using the **`lsp flooding dynamic`** command under the ***router config*** mode. For example:

```
switch(config)# router isis Amun
switch(config-router-isis)# net 49.0000.0000.3333.00
switch(config-router-isis)# is-hostname ip3
switch(config-router-isis)# lsp flooding dynamic
```

Dynamic flooding should be enabled on all routers in the area. To enable Dynamic Flooding on all routers, use the following command:

`lsp flooding dynamic` [level-1 | level-2]

`nolsp flooding dynamic` [level-1 | level-2]

`default lsp flooding dynamic` [level-1 | level-2]

If necessary, the area leader election process can be tuned or disabled with the commands:

`area leader` [level-1 | level-2] priority ***0-255*** area leader [level-1 | level-2] disabled

`no area leader` [level-1 | level-2] priority ***0-255*** area leader [level-1 | level-2] disabled

`default area leader` [level-1 | level-2] priority ***0-255*** area leader [level-1 | level-2] disabled

15.4.5.9.1 Limitations

On a sparse topology, Dynamic Flooding is not effective and only adds overhead. Leaf-spine and Clos networks are appropriate dense topologies.

15.4.5.10 Relax Address-Family Check for IS-IS Adjacency

Address-Family Check for IS-IS creates the adjacency between devices with different address famiies. For example, a router supporting IPv4 and IPv6 is connected to a IPv4 only router, **Address-Family Check** is

verified by comparing the NLPID TLV (Type #129) advertised in IIH hellos exchanged between peers. It is useful in following scenarios.

Incrementally Enable IPv6 in an Existing IPv4 Network

Relaxing the **Address-Family Check** is useful to gradually add IPv6 support in an IPv4 network, without disturbing the IPv4 connectivity.

IPv4 Controller Peering IPv4/v6 Dual Stack Router

A controller forms an IS-IS adjacency with a router and uses the IS-IS database for topology discovery. If the controller only supports IPv4 IS-IS or only IPv4 tunnels, to relax the **Address-Family Check** on the dual stack IPv4/v6 router for adjacency is useful in establishment.

Disabling the Address-family Check

Under IS-IS instance, configure the following to disable the **Address-Family Check** during IIH processing.

```
switch(config-router-isis)#?
   adjacency Configure parameters for adjacency formation
switch(config-router-isis)# adjacency?
   address-family Configure address-family related parameters for adjacency
 formation
switch(config-router-isis)# adjacency address-family?
   match Configure address-family match check related parameters for adjacency
 formation
switch(config-router-isis)# adjacency address-family match?
   disabled Relax address-family match check for bringing up adjacency
switch(config-router-isis)# adjacency address-family match disabled?
```

Show Command

The **show isis neighbor detail** command displays address family details at each end of the adjacency.

```
switch# show isis neighbor detail
Instance  VRF      System Id   Type Interface       SNPA        State Hold time
   Circuit Id
inst1     default  1111.1111.1002   L2   Vlan2116           P2P            UP
  24         06
  Area Address(es): 49.0001
  SNPA: P2P
  Router ID: 1.0.0.2
  Advertised Hold Time: 30
  State Changed: 00:04:18 ago at 2020-11-01 22:28:35
  IPv4 Interface Address: 1.0.0.2
  IPv6 Interface Address: none
  Interface name: Vlan2116
  Graceful Restart: Supported
  Segment Routing Enabled
    SRGB Base: 900000 Range: 65536
    Adjacency Label IPv4: 149152
  Supported Address Families: IPv4, IPv6
  Neighbor Supported Address Families: IPv4
```

The **show isis interface detail** command shows the details of supported protocols on the interface and the neighbors connected to it. The state of **Address-Family match check** is also displayed.

```
switch(config-router-isis)# show isis interface detail
IS-IS Instance: inst1 VRF: default
```

```
 Interface Vlan2116:
   Index: 35 SNPA: P2P
   MTU: 1497 Type: point-to-point
  Supported Address Families: IPv4, IPv4
   Area Proxy Boundary is Disabled
   BFD IPv4 is Disabled
   BFD IPv6 is Disabled
   Hello Padding is Enabled
   Level 2:
     Metric: 10, Number of adjacencies: 1
     Link-ID: 23
     Authentication mode: None
     TI-LFA link protection is enabled for the following IPv4 segments: node
segments, adjacency segments
     TI-LFA protection is disabled for IPv6
   Adjacency 1111.1111.1002:
     State: UP, Level: 2 Type: Level 2 IS
     Advertised Hold Time: 30
    Neighbor Supported Address Families: IPv4
    Address Family Match: Disabled
     IPv4 Interface Address: 1.0.0.2
     Areas:
       49.0001
```

Usage Guidelines

For IPv6 network upgrade, ensure the knob is incrementally configured on a contiguous section of the network, at any point the choice of routers for upgrade should not bisect the upgraded (supporting IPv4/v6) part of the network. All the routers bordering the upgraded network should always have the knob enabled.

When a proper set of router is established, the following steps are carried on each router.

1. Enable the CLI knob.
2. Enable IPv6 address family in the IS-IS instance.
3. Configure IPv6 on all the IS-IS interfaces.

15.4.5.11 Displaying IS-IS Information

The following sections describe display and verification of IS-IS settings and of peer and connection configuration:

- Displaying the Link State Database
- Displaying the Interface Information for the IS-IS Instance
- Displaying IS-IS Neighbor Information
- Displaying IS-IS Instance Information
- Displaying IS-IS Segment Routing Information
- Displaying show isis local-convergence-delay
- Verifying IS-IS Graceful Restart (GR) Information
- IS-IS Dynamic Flooding Show Commands

15.4.5.11.1 Displaying the Link State Database

To display the link state database of IS-IS, use the **`show isis database`** command.

Example

This command displays the IS-IS link state database.

```
switch# show isis database
ISIS Instance: Osiris
```

```
  ISIS Level 2 Link State Database
    LSPID                 Seq Num   Cksum  Life  IS Flags
    1212.1212.1212.00-00  4         714    1064  L2 <>
    1212.1212.1212.0a-00  1         57417  1064  L2 <>
    2222.2222.2222.00-00  6         15323  1116  L2 <>
    2727.2727.2727.00-00  10        15596  1050  L2 <>
    3030.3030.3030.00-00  12        62023  1104  L2 <>
    3030.3030.3030.c7-00  4         53510  1104  L2 <>
switch>
```

15.4.5.11.2 Displaying the Interface Information for the IS-IS Instance

To display interface information related to the IS-IS instance, use the **show isis interface** command.

Example

This command displays IS-IS interface information.

```
switch# show isis interface

ISIS Instance: Osiris
  Interface Vlan20:
    Index: 59 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 2
      LAN-ID: 1212.1212.1212, Priority: 64
      DIS: 1212.1212.1212, DIS Priority: 64
  Interface Ethernet30:
    Index: 36 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 1
      LAN-ID: 3030.3030.3030, Priority: 64
      DIS: 3030.3030.3030, DIS Priority: 64 switch>
```

15.4.5.11.3 Displaying IS-IS Neighbor Information

To display general information for IS-IS neighbors that the device sees, use **show isis neighbors**.

Example

This command displays information for IS-IS neighbors that the device sees.

```
switch# show isis neighbor
Inst Id   System Id       Type  Interface   SNPA          State  Hold time
10        2222.2222.2222  L2    Vlan20       2:1:0:c:0:0   UP    30
10        1212.1212.1212  L2    Vlan20       2:1:0:d:0:0   UP     9
10        3030.3030.3030  L2    Ethernet30   2:1:0:b:0:0   UP     9 switch>
```

15.4.5.11.4 Displaying IS-IS Instance Information

To display the system ID, Type, Interface, IP address, State and Hold information for IS-IS instances, use the **show isis summary** command. The command is also used to verify the configured maximum wait interval, initial wait interval, and hold time of SPF timers in IS-IS instances. This command also displays values of the current SPF interval, last Level-1 SPF run, and last Level-2 SPF run.

Example

- This command displays general information about IS-IS instances.

```
switch# show isis summary
ISIS Instance: Osiris
  System ID: 1010.1040.1030, administratively enabled, attached
  Internal Preference: Level 1: 115, Level 2: 115
  External Preference: Level 1: 115, Level 2: 115
  IS-Type: Level 2, Number active interfaces: 1
  Routes IPv4 only
  Last Level 2 SPF run 2:32 minutes ago
  Area Addresses:
    10.0001
  level 2: number dis interfaces: 1, LSDB size: 1 switch>
```

- This command displays the SPF interval information about IS-IS instances.

```
switch(config-router-isis-af)# show isis summary

IS-IS Instance: 1 VRF: default
 System ID: 0000.0000.0001, administratively enabled
 Multi Topology disabled, not attached
 IPv4 Preference: Level 1: 115, Level 2: 115
 IPv6 Preference: Level 1: 115, Level 2: 115
 IS-Type: Level 1 and 2, Number active interfaces: 0
 Routes both IPv4 and IPv6
                           Max wait(s) Initial wait(ms) Hold interval(ms)
 LSP Generation Interval:     5              50              50
 SPF Interval:                2            1000            1000
 Current SPF hold interval(ms): Level 1: 1000, Level 2: 1000
 Last Level 1 SPF run 1 seconds ago
 Last Level 2 SPF run 1 seconds ago
 Authentication mode: Level 1: None, Level 2: None
 Graceful Restart: Disabled, Graceful Restart Helper: Enabled
 Area Addresses:
   49.0001
 level 1: number dis interfaces: 0, LSDB size: 1
 level 2: number dis interfaces: 0, LSDB size: 1
```

15.4.5.11.5 Displaying IS-IS Segment Routing Information

IS-IS Segment Routing information is displayed using the following commands:

- **show isis database detail**
- **show isis segment-routing**
- **show isis segment-routing global-blocks**
- **show isis segment-routing prefix-segments**
- **show isis segment-routing prefix-segments self-originated**
- **show isis segment-routing adjacency-segments**
- **show mpls label ranges**
- **show mpls segment-routing bindings**
- **show mpls lfib route**
- **show mpls lfib route <label value>**

show isis database detail

The **show isis database detail** command provides a view of LSPDB of different devices in the IS-IS domain. The output displays the TLVs and sub-TLVs that are being self-originated or the ones that have been received from other routers.

Example

```
switch# show isis database detail
ISIS Instance: inst1 VRF: default
 ISIS Level 2 Link State Database
 LSPID Seq Num Cksum Life IS Flags
 1111.1111.1001.00-00 10 63306 751 L2 <>
 NLPID: 0xCC(IPv4) 0x8E(IPv6)
 Area address: 49.0001
 Interface address: 1.0.7.1
 Interface address: 1.0.0.1
 Interface address: 2000:0:0:47::1
 Interface address: 2000:0:0:40::1
 IS Neighbor : lf319.53 Metric: 10
    LAN-Adj-sid: 100000 flags: [ L V ] weight: 0 system ID: 1111.1111.1002
 IS Neighbor (MT-IPv6): lf319.53 Metric: 10
    LAN-Adj-sid: 100001 flags: [ L V F ] weight: 0 system ID: 1111.1111.1002
 Reachability : 1.0.11.0/24 Metric: 1 Type: 1 Up
    SR Prefix-SID: 10 Flags: [ R ] Algorithm: 0
 Reachability : 1.0.3.0/24 Metric: 1 Type: 1 Up
 Reachability : 1.0.7.1/32 Metric: 10 Type: 1 Up
    SR Prefix-SID: 2 Flags: [ N ] Algorithm: 0
 Reachability : 1.0.0.0/24 Metric: 10 Type: 1 Up
 Reachability (MT-IPv6): 2000:0:0:4b::/64 Metric: 1 Type: 1 Up
    SR Prefix-SID: 11 Flags: [ R ] Algorithm: 0
 Reachability (MT-IPv6): 2000:0:0:43::/64 Metric: 1 Type: 1 Up
 Reachability (MT-IPv6): 2000:0:0:47::1/128 Metric: 10 Type: 1 Up
    SR Prefix-SID: 3 Flags: [ N ] Algorithm: 0
 Reachability (MT-IPv6): 2000:0:0:40::/64 Metric: 10 Type: 1 Up
 Router Capabilities: 252.252.1.252 Flags: [ ]
    SR Capability: Flags: [ I V ]
       SRGB Base: 900000 Range: 65536
 Segment Binding: Flags: [ F ] Weight: 0 Range: 1 Pfx 2000:0:0:4f::1/128
    SR Prefix-SID: 19 Flags: [ ] Algorithm: 0
 Segment Binding: Flags: [ ] Weight: 0 Range: 1 Pfx 1.0.15.1/32
    SR Prefix-SID: 18 Flags: [ ] Algorithm: 0
```

show isis segment-routing

The **show isis segment-routing** command displays the summary information on IS-IS SR status.

Example

```
switch(config)# show isis segment-routing
System ID: 1111.1111.1002               Instance: inst1
SR supported Data-plane: MPLS           SR Router ID: 252.252.2.252
SR Global Block( SRGB ): Base: 900000   Size: 65536
Adj-SID allocation mode: SR-adjacencies
Adj-SID allocation pool: Base: 100000     Size: 16384
All Prefix Segments have    : P:0 E:0 V:0 L:0
All Adjacency Segments have : F:0 B:0 V:1 L:1 S:0
ISIS Reachability Algorithm : SPF (0)
Number of ISIS segment routing capable peers: 3
Self-Originated Segment Statistics:
Node-Segments       : 2
Prefix-Segments     : 2
Proxy-Node-Segments : 0
Adjacency Segments  :
```

About the Output

The first line of the output shows the IS-IS system ID of this device and the name of the instance with which IS-IS is configured.

The supported data plane is shown against the SR supported Data-plane field whereas the Router ID being advertised in the Router Capability is mentioned in the SR Router ID Field.

The SRGB in use and the MPLS label pool being used for adjacency segment allocation are mentioned in this output. The current adjacency allocation mode which refers to whether we are allocating adjacency segments to all IS-IS adjacencies or only those adjacencies which support SR or None of the adjacencies is shown in the Adj-SID allocation mode field.

Flag contents of All Prefix Segments originated on this router, Flag contents of All Adjacency Segments originated on this router and supported IS-IS Reachability Algorithm have been provided through this command output and they carry the meaning as per the IS-IS SR IETF draft.

This show command provides a statistics related to IS-IS SR in terms of various counters ranging from number of IS-IS SR enabled peers, number of Node-SIDs, prefix-SIDs, proxy-node-segments and adjacency segments being originated on this router in IS-IS.

The **`show isis segment-routing`** command also provides information if segment routing has been administratively disabled as shown.

```
switch(config-router-isis-sr-mpls)# show isis segment-routing
! IS-IS (Instance: inst1) Segment Routing has been administratively shutdown
```

show isis segment-routing global-blocks

The **`show isis segment-routing global-blocks`** command lists the SRGBs in use by all SR supporting devices in IS-IS domain including the SRGB in use by IS-IS SR on this device.

Example

```
switch# show isis segment-routing global-blocks
System ID: 1111.1111.1002              Instance: inst1
SR supported Data-plane: MPLS          SR Router ID: 252.252.2.252
SR Global Block( SRGB ): Base: 900000  Size: 65536
Number of ISIS segment routing capable peers: 3
SystemId             Base         Size
-------------------- ------------ -----
1111.1111.1002        900000      65536
1111.1111.1001        900000      65536
```

show isis segment-routing prefix-segments

The **`show isis segment-routing prefix-segments`** command provides the details of all prefix segments being originated as well the segments received from IS-IS SR speakers in the domain.

Example

```
switch# show isis segment-routing prefix-segments
System ID: 1111.1111.1002         Instance: inst1
SR supported Data-plane: MPLS     SR Router ID: 252.252.2.252
Node: 2 Proxy-Node: 2 Prefix: 2 Total Segments: 6
Flag Descriptions: R: Re-advertised, N: Node Segment, P: no-PHP
                   E: Explicit-NULL, V: Value, L: Local
Segment status codes: * - Self originated Prefix, L1 - level 1, L2 - level 2
Prefix        SID     Type        Flags                    SystemID        Type
------------------- --------- ---------------         --------------------
 1.0.7.1/32    2      Node        R:0 N:1 P:0 E:0 V:0 L:0   1111.1111.1001  L1
* 1.0.8.1/32   4      Node        R:0 N:1 P:0 E:0 V:0 L:0   1111.1111.1002  L2
 1.0.11.0/24   10     Prefix      R:1 N:0 P:0 E:0 V:0 L:0   1111.1111.1001  L2
* 1.0.12.0/24  12     Prefix      R:1 N:0 P:0 E:0 V:0 L:0   1111.1111.1002  L2
 1.0.15.1/32   18     Proxy-Node  R:0 N:0 P:0 E:0 V:0 L:0   1111.1111.1001  L2
 1.0.16.1/32   20     Proxy-Node  R:0 N:0 P:0 E:0 V:0 L:0   1111.1111.1003  L2
```

About the Output

After the usual output header that represents the system ID, instance name, etc and parameters of a router, there is a line depicting prefix segment counters. Each field in this line relates to the number of segments that are present in this routers IS-IS instance. For example, the above example shows that this device has 2 Node Segments (Self originated as well as the ones received from other IS-IS SR devices).

The main section of this show commands output is the section that lists all the prefix segments and related information like prefix, SID, type of segment (Prefix, Node, Proxy-Node), the flag values being carried in the sub-TLVs of these prefix segments and the system ID of the originating router. The Type field will be useful on a IS type level-1-2 router. It shows whether the installed prefix segment is from a level-1 prefix or a level-2 prefix.

show isis segment-routing prefix-segments self-originated

The **`show isis segment-routing prefix-segments self-originated`** command output is identical to show isis segment-routing prefix-segments except, the fact that the former lists only self-originated prefix segments.

show isis segment-routing adjacency-segments

The **`show isis segment-routing adjacency-segments`** displays list of all the adjacency segments that are being originated by IS-IS SR on a router.

Example

```
switch# show isis segment-routing adjacency-segments
System ID: 1111.1111.1002           Instance: inst1
SR supported Data-plane: MPLS       SR Router ID: 252.252.2.252
Adj-SID allocation mode: SR-adjacencies
Adj-SID allocation pool: Base: 100000     Size: 16384
Adjacency Segment Count: 4
Adj IP-address     Local     Intf    Label    SID  Source     Flags      Type
-----------------  --------  ------  ------   ---  ---------  ---------  --------
1.0.0.1            Vlan2472  100000  Dynamic  F:0   B:0 V:1    L:1 S:0   LAN L2
1.0.1.2            Vlan2579  100001  Dynamic  F:0   B:0 V:1    L:1 S:0   P2P L2
fe80::1:ff:fe01:0  Vlan2472  100002  Dynamic  F:0   B:0 V:1    L:1 S:0   LAN L2
fe80::1:ff:fe02:0  Vlan2579  100003  Dynamic  F:0   B:0 V:1    L:1 S:0   P2P L2
```

About the Output

It consists allocation mode, MPLS label pool from which labels would be allocated to adjacencies, total count of adjacency segments allocated so far and the default flag values carried in all adj-SID sub-TLVs originating from this device.

The main section of the output lists all the adjacency segments allocated so far in six columns each pertaining to Adjacency IP address, local interface name, MPLS label value, SID source, flags in the sub-TLV and the type of adj-SID respectively. The type of the adjacency segments depends on the IS-IS type of adjacency and the IS level.

show mpls label ranges

The **`show mpls label ranges`** command displays the MPLS label range available on a router is categorized into different pools which cater to different applications running on the router.

The isis-sr refers to the SRGB use-case in IS-IS, and isis (dynamic) refers to the label pool that is used for dynamic allocation of adjacency segments in IS-IS.

Example

```
switch# show mpls label ranges
Start    End      Size     Usage
-----------------------------------------
0        15       16       reserved
```

```
16       99999    99984    static mpls
100000   116383   16384    isis (dynamic)
116384   362143   245760   free (dynamic)
362144   899999   537856   unassigned
900000   965535   65536    isis-sr
```

show mpls segment-routing bindings

The **`show mpls segment-routing bindings`** command displays the local label bindings and label bindings on the peer routers for each prefix that has a segment advertised. Peer ID here represents the IS-IS system ID of the peer.

Example

```
switch# show mpls segment-routing bindings
1.0.7.1/32
 Local binding: Label: 900002
 Remote binding: Peer ID: 1111.1111.1001, Label: imp-null
 Remote binding: Peer ID: 1111.1111.1003, Label: 900002
1.0.8.1/32
 Local binding: Label: imp-null
 Remote binding: Peer ID: 1111.1111.1001, Label: 900004
 Remote binding: Peer ID: 1111.1111.1003, Label: 900004
1.0.9.1/32
 Local binding: Label: 900006
 Remote binding: Peer ID: 1111.1111.1001, Label: 900006
 Remote binding: Peer ID: 1111.1111.1003, Label: imp-null
```

show mpls lfib route

The **`show mpls lfib route`** command displays the LFIB. Each LFIB entry has In-Label, Out-Label, metric, payload type, nexthop information, etc. fields. The source column depicts the MPLS control plane protocol that is responsible for the label binding that resulted in this LFIB route.

Example

```
switch# show mpls lfib route
MPLS forwarding table (Label [metric] Vias) - 7 routes
MPLS next-hop resolution allow default route: False
Via Type Codes:
        M - Mpls Via, P - Pseudowire Via,
        I - IP Lookup Via, V - Vlan Via,
        VA - EVPN Vlan Aware Via, ES - EVPN Ethernet Segment Via,
        VF - EVPN Vlan Flood Via, AF - EVPN Vlan Aware Flood Via,
        NG - Nexthop Group Via
Source Codes:
        S - Static MPLS Route, B2 - BGP L2 EVPN,
        B3 - BGP L3 VPN, R - RSVP,
        P - Pseudowire, L - LDP,
        IP - IS-IS SR Prefix Segment, IA - IS-IS SR Adjacency Segment,
        IL - IS-IS SR Segment to LDP, LI - LDP to IS-IS SR Segment,
        BL - BGP LU, ST - SR TE Policy,
        DE - Debug LFIB

 IA 100000 [1]
            via M, 1.0.1.2, pop
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930
 IA 100001 [1]
            via M, fe80::200:eff:fe02:0, pop
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930
```

```
 IP 900008 [1]
            via M, 1.0.1.2, swap 900008
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930
 IP 900009 [1]
            via M, fe80::200:eff:fe02:0, swap 900009
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930
```

show mpls lfib route <label value>

The **show mpls lfib route <label value>** command provides information relevant to just the label value passed as an extension to the show command.

Example

```
switch# show mpls lfib route 900008
MPLS forwarding table (Label [metric] Vias) - 7 routes
MPLS next-hop resolution allow default route: False
Via Type Codes:
        M - Mpls Via, P - Pseudowire Via,
        I - IP Lookup Via, V - Vlan Via,
        VA - EVPN Vlan Aware Via, ES - EVPN Ethernet Segment Via,
        VF - EVPN Vlan Flood Via, AF - EVPN Vlan Aware Flood Via,
        NG - Nexthop Group Via
Source Codes:
        S - Static MPLS Route, B2 - BGP L2 EVPN,
        B3 - BGP L3 VPN, R - RSVP,
        P - Pseudowire, L - LDP,
        IP - IS-IS SR Prefix Segment, IA - IS-IS SR Adjacency Segment,
        IL - IS-IS SR Segment to LDP, LI - LDP to IS-IS SR Segment,
        BL - BGP LU, ST - SR TE Policy,
        DE - Debug LFIB
 IP 900008 [1]
            via M, 1.0.1.2, swap 900008
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930
```

15.4.5.11.6 Displaying show isis local-convergence-delay

The **show isis local-convergence-delay** command shows the current or last attempt at delaying the convergence of protected routes on a link down/BFD neighbor down event. If the timer aborts for some reason (such as a topology change causing a new SPF), the attempt fails.

```
switch# show isis local-convergence-delay

IS-IS Instance: inst1 VRF: default
  System ID: 1111.1111.1001
  IPv4 local convergence delay configured, 5000 msecs
  IPv6 local convergence delay configured, 5000 msecs
  Level 1 attempts 0, failures 0
  Level 2 attempts 3, failures 1

  Level 2 in progress due to LINK DOWN on Vlan2138
    TI-LFA node protection is enabled for IPv4
    IPv4 Routes delayed: 0
      Delay timer started at: 2019-07-25 23:16:33
      Delay timer expires in 2 secs
    TI-LFA protection is disabled for IPv6
```

```
  Level 2 last attempt due to LINK DOWN on Vlan2138, Succeeded
    TI-LFA node protection is enabled for IPv4
    IPv4 Routes delayed: 3
      Delay timer started at: 2019-07-25 23:14:51
      Delay timer stopped at: 2019-07-25 23:14:56
    TI-LFA protection is disabled for IPv6
```

The **detail** keyword also lists all the routes that have been delayed.

```
switch# show isis local-convergence-delay detail
  ...
  Level 2 last attempt due to LINK DOWN on Vlan2138, Succeeded
    TI-LFA node protection is enabled for IPv4
    IPv4 Routes delayed: 3
      Delay timer started at: 2019-07-25 23:14:51
      Delay timer stopped at: 2019-07-25 23:14:56
      Delayed routes:
        10.0.7.1/32
        10.0.9.1/32
        10.0.10.1/32
    TI-LFA protection is disabled for IPv6
```

15.4.5.11.7 Verifying IS-IS Graceful Restart (GR) Information

GR State can be one of the following:

- Last Start/Restart was completed successfully.
- Last Start/Restart exited after t2 (level-1/level-2) expiry.
- Last Restart exited after t3 expiry.
- Start/Restart in progress.
- Graceful Restart was disabled during startup.

The following show commands are used to display the IS-IS graceful restart information.

- The **`show isis graceful-restart vrf [vrf-name]`** command displays the GR configuration and graceful-restart related state of the IS-IS instance as well as its neighbors.

  Example

```
switch# show isis graceful-restart vrf default
IS-IS Instance: 1 VRF: default
 System ID: 0000.0000.0001
 Graceful Restart: Enabled, Graceful Restart Helper: Enabled
 State: Last Start exited after T2 (level-1) expiry
 T1 : 3s
 T2 (level-1) : 30s/20s remaining
 T2 (level-2) : 30s/not running
 T3 : not running

System ID       Type   Interface     Restart Capable  Status
is-hostname-1   L1L2   Ethernet1     Yes              Running
is-hostname-2   L1     Ethernet2     Yes              Restarting
```

- The **`show isis summary vrf [vrf-name]`** command displays the graceful restart state and helper configuration.

Example

```
switch# show isis summary vrf default
IS-IS Instance: 1 VRF: default
 System ID: 0000.0000.0001, administratively enabled
 ....
 Graceful Restart: Enabled, Graceful Restart Helper: Enabled
```

- The **show isis neighbors detail vrf [vrf-name]** command displays the helpers view of a restarting router.

  Example

```
switch# show isis neighbors detail vrf default
Instance  VRF      System Id  Type Interface  SNPA     State  Hold time   Circuit Id
1         default  OT1        L1   Ethernet1  2:1:0:b  4:0:0  UP          29839   OT3.05
  Area Address(es): 49.0001
  SNPA: 2:1:0:b4:0:0
 ....
 Graceful Restart: Supported, Status: Restarting (RR rcvd, RA sent, CSNP sent)
```

- The **show isis interface detail vrf [vrf-name]** command displays the graceful restart related stats for that interface.

  Example

```
switch# show isis interface detail vrf default
ISIS Instance: ISISQ VRF: default
  Interface Ethernet1:
    Index: 2 SNPA: P2P
    ...
    Level 1:
      Graceful Restart Status: RR sent, SA sent, RA rcvd, CSNP rcvd
```

15.4.5.11.8 IS-IS Dynamic Flooding Show Commands

Several show commands are available to monitor Dynamic Flooding. To see the flooding topology, use the **show isis dynamic flooding topology** command:

```
switch# show isis dynamic flooding topology

IS-IS Instance: Amun VRF: default
  Level 1:
    Path: ip6.00 ip4.00 ip2.00 ip1.00 ip3.00 ip5.00 ip6.00
```

This command displays a list of paths that describe the flooding topology. Each path is a list of nodes in the network.

To see which interfaces dynamic flooding will use, use the **show isis dynamic flooding interfaces** command:

```
switch# show isis dynamic flooding interfaces

IS-IS Instance: Amun VRF: default
  Level 1:
    Ethernet5
    Ethernet4
```

This shows that the system is currently flooding only on ***ethernet4*** and ***ethernet5***. Normally at least two interfaces are selected.

15.4.6 IS-IS Commands

Global Configuration Commands

- router isis

Clear Commands

- clear isis database
- clear isis neighbor

Interface Configuration Commands

- adjacency-segment
- adjacency-segment (allocation)
- adjacency-segment (static)
- area leader
- authentication key
- authentication mode
- bfd all-interfaces
- isis [ipv4|ipv6] fast-reroute ti-lfa srlg
- isis authentication key
- isis authentication mode
- isis bfd
- isis enable
- isis fast-reroute ti-lfa mode
- isis hello-interval
- isis hello-multiplier
- isis ipv6 metric
- isis lsp tx interval
- isis metric
- isis multi-topology
- isis network
- isis passive
- isis priority

Router IS-IS Configuration Mode (Includes Address-Family Mode)

- address-family
- fast-reroute ti-lfa mode
- fast-reroute ti-lfa srlg
- graceful-restart (IS-IS)
- is-hostname
- is-type
- log-adjacency-changes (IS-IS)
- lsp dynamic flooding
- match isis level
- multi-topology
- net
- passive (IS-IS)
- redistribute (IS-IS)
- redistribute bgp route-map

- set isis level
- set-overload-bit
- shutdown (IS-IS)
- spf-interval

IS-IS Segment Routing Commands

- adjacency-segment (static)
- mpls label range
- node-segment
- prefix-segment
- proxy-node-segment
- segment-routing mpls
- shutdown (IS-IS SR)

Display Commands EXEC Mode

- show ip route
- show isis database
- show isis database detail
- show isis dynamic flooding
- show isis graceful-restart vrf
- show isis hostname
- show isis interface
- show isis local-convergence-delay
- show isis neighbors
- show isis network topology
- show isis segment-routing
- show isis segment-routing prefix-segments
- show isis segment-routing adjacency-segments
- show isis segment-routing global-blocks
- show isis segment-routing tunnel
- show isis summary
- show isis ti-lfa path
- show isis ti-lfa tunnel
- show mpls label ranges
- show mpls segment-routing bindings
- show mpls lfib route
- show tunnel fib

15.4.6.1 address-family

The **`address-family`** command places the switch in address-family configuration mode.

Address-family configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **`exit`** command does not affect the configuration.

The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

The **`no address-family`** and **`default address-family`** commands delete the specified address-family from ***running-config*** by removing all commands previously configured in the corresponding address-family mode.

The **exit** command returns the switch to the ***isis*** configuration mode.

Command Mode

Router-IS-IS Configuration

Command Syntax

address-family [ipv4 | ipv6][MODE]

no address-family [ipv4 | ipv6][**MODE]**

default address-family [ipv4 | ipv6][**MODE**]

Parameters

- **address_family** Options include:
  - **ipv4** IPv4 unicast
  - **ipv6** IPv6 unicast
- **MODE** Options include:
  - ***no parameter*** Defaults to unicast.
  - **unicast** All IPv4 or IPv6 addresses are active.

Example

- These commands enter the ***address family*** mode for IPv4 unicast.

```
switch(config)# router isis Osiris
switch(config-router-isis)# address-family ipv4 unicast
switch(config-router-isis-af)#
```

- To exit from the IPv4 IS-IS unicast ***address family*** configuration mode, enter the following command.

```
switch(config)# router isis Osiris
switch(config-router-isis)# address-family ipv4 unicast
switch(config-router-isis-af)# exit
switch(config-router-isis)#
```

15.4.6.2 adjacency-segment

Use the **adjacency-segment** command in the interface configuration mode to have the PLR computes backup paths for an adjacency segment only if the Adjacency SID sub-TLV has the B-flag (backup flag) set.

Command Mode

Interface configuration mode

Command Syntax

adjacency-segment [ipv4|ipv6] p2p [multiple][label ***label*** | index***index***] backup-eligible

no adjacency-segment [ipv4 | ipv6]p2p [multiple][label ***label***|index ***index***] backup-eligible

default adjacency-segment [ipv4 | ipv6]p2p multiple][label ***label***|index ***index***] backup-eligible

Parameters

- **ipv4** IPv4 related.
- **ipv6** IPv6 related.
- **p2p** P2P interface type.
- **multiple** Configure multiple Adj-SIDs.
- **label *label*** Label value to be assigned as Adj-SID for adjacency on this interface. ***label*** range ***16 -1048575***.
- **index *index*** Prefix segment identifier. ***index*** range ***0-65535***.

- **backup-eligible** Eligible for protection.

15.4.6.3 adjacency-segment (allocation)

The **adjacency-segment** command allocates adjacency segments to all IS-IS adjacencies, or only those adjacencies which are to IS-IS routers that have advertised IS-IS SR capability, or to none of the adjacencies.

Command Mode

Segment-Routing MPLS Configuration

Command Syntax

adjacency-segment allocation [all-interface |none | sr-peers]

Parameters

- **allocation** Allocation of Adjacency Segments.
- **all-interface** Allocates adjacency segments to all IS-IS adjacencies.
- **none** Disable automatic adjacency segment allocation.
- **sr-peers** Allocate adjacency segments to IS-IS adjacencies with SR peers.

Example

This command allocates the adjacency segment to an sr-peer.

```
switch(config-router-isis-sr-mpls)# adjacency-segment allocation sr-peer
```

15.4.6.4 adjacency-segment (static)

The **adjacency-segment** command configures IS-IS adjacencies statically on the switch, so that these values are preserved even when the switch restarts. The no and the default form of the command places the switch back to the global configuration mode.

Command Mode

Interface Ethernet Configuration

Command Syntax

adjacency-segment ipv4 | ipv6 p2p [[label ***label-value***]|[index ***index-value*** global]]

Parameters

- **ipv4** IS-IS SR adjacency segment IPv4 interface configuration.
- **ipv6** IS-IS SR adjacency segment IPv6 interface configuration.
- ***label*** Label value to be assigned as Adj-SID for adjacency on this interface. Value ranges from ***16*** to ***1048575***.
- ***index*** Index to be assigned as Adj-SID for adjacency on this interface. Value ranges from ***0*** to ***65535***.
- **global** global adjacency SID.

Example

This command allocates the adjacency segment to an IPv4 p2p interface with a index value ***50***.

```
switch(config-if-Et1)# adjacency-segment ipv4 p2p index 50 global
```

15.4.6.5 area leader

Use the **area leader** command to tune or disable the area leader election process.

Command Mode

Router configuration mode

Command Syntax

area leader [disabled | level-1 [disabled] | level-2 [disabled] | priority [***num*** [level-1 | level-2]]]

no area leader

default area leader

Parameters

- **disabled** Disables becoming the are leader.
- **level-1** Configure at Level 1.
  - **disabled** Disables becoming the are leader.
- **level-2** Configure at Level 2.
  - **disabled** Disables becoming the are leader.
- **priority** Sets the area leader priority.
  - **level-1** Configure at Level 1.
  - **level-2** Configure at Level 2.

15.4.6.6 authentication key

The**authentication key** command configures the authentication key for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated.

The **no authentication key** and **default authentication key** commands disables the authentication key for the IS-IS instance.

Command Mode

ISIS-Router Configuration

Command Syntax

authentication key [***0*** | ***7***] **[LAYER_VALUE]**

no authentication key [***0*** | ***7***] **[LAYER_VALUE]**

default authentication key [***0*** | ***7***] **[LAYER_VALUE]**

Parameters

LAYER_VALUE layer value. Options include:

- **level-1**
- **level-2**

Example

These commands configure authentication for the IS-IS instance causing LSPs, CSNPs, and PSNPs to be authenticated.

```
switch(config)# router isis 1
switch(config-router-isis)# authentication key secret
switch(config-router-isis)#
```

15.4.6.7 authentication mode

The **authentication mode** command configures authentication for the IS-IS instance causing LSPs, CSNPs, and PSNPs to be authenticated.

The **no authentication mode** and **default authentication mode** commands disables authentication for the IS-IS instance.

Command Mode

ISIS-Router Configuration

Command Syntax

authentication mode [md5 | *text*] [LAYER_VALUE]

no authentication mode [md5 | *text*] [LAYER_VALUE]

default authentication mode [md5 | *text*] [LAYER_VALUE]

Parameters

- **LAYER_VALUE** Layer value. Options include:
  - **level-1**
  - **level-2**

Example

These commands configure authentication for the IS-IS instance causing LSPs, CSNPs, and PSNPs to be authenticated.

```
switch(config)# router isis 1
switch(config-router-isis)# authentication mode md5
switch(config-router-isis)#
```

15.4.6.8 bfd all-interfaces

The **bfd all-interfaces** command enables Bidirectional Forwarding Detection (BFD) for all IS-IS-enabled interfaces in the IPv4 or IPv6 address family.

Use the **isis bfd** command to configure BFD on a specific interface.

Command Mode

Router-IS-IS Address-Family Configuration

Command Syntax

bfd all-interfaces

Example

These commands enable BFD for all the interfaces on which IS-IS is enabled. By default, BFD is disabled on all the interfaces.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv4
switch(config-router-af)# bfd all-interfaces
switch(config-router-af)#
```

15.4.6.9 clear isis database

The **clear isis database** command clears a specific LSP with a predefined LSP ID, or LSPs at a given level, or all LSPs in the database. Additionally, the command sends purge LSPs throughout the network to clear LSPs from all devices.



Note: Exercise caution while using this command since it can be disruptive to the network.

Command Mode

Privileged Exec

Command Syntax

`clear isis [INSTANCE] database` {*LSPID* | all | level-1 | level-2}

Parameters

- ***INSTANCE*** Clears all LSPs from a specific LSP instance.
- ***LSPID*** Clears an LSP based on the specific LSP ID.
- **all** Clears all LSPs from the LSP database.
- **level-1** Clears LSPs at level 1 only.
- **level-2** Clears LSPs at level 2 only.

Examples

- This command clears all LSPs for the specific LSP ID of ***1111.1111.1002.00-00***.

```
switch(config)# clear isis database 1111.1111.1002.00-00
1 LSPs cleared on instance 1.
switch(config)#
```

- This command clears all LSPs from the LSP database.

```
switch(config)# clear isis database all
3 LSPs cleared on instance 1.
switch(config)#
```

- This command clears all LSPs from the level 1 LSP database.

```
switch(config)# clear isis database level-1
3 LSPs cleared on instance 1.
switch(config)#
```

- This command clears all LSPs from a specific LSP instance *instance2*.

```
switch(config)# clear isis instance2 database all
3 LSPs cleared on instance instance 2.
switch(config)#
```

15.4.6.10 clear isis neighbor

The **`clear isis neighbor`** command clears IS-IS adjacencies that exist on an interface, or at a specific level, or the adjacencies formed with a given neighbor (either with a system ID or a hostname).

Command Mode

Privileged EXEC

Command Syntax

`clear isis neighbor` {*Neighbor-ID* | all | *interface*} [level-1 | level-2 | level-1-2]

Parameters

- ***Neighbor-ID*** Clears adjacencies based on the system ID or the hostname of a neighbor.
- **all** Clears all adjacencies.
- ***interface*** Clears adjacencies for a specific interface.
- **level-1** level 1 only.
- **level-1-2** level 1-2 point-to-point only.
- **level-2** level 2 only.

Examples

- This command clears IS-IS adjacencies with a neighbor ***af86.3032.1a0f***.

```
switch# clear isis neighbor af86.3032.1a0f
2 neighbors cleared on instance 1 switch#
```

- This command clears all IS-IS adjacencies on an interface ***et1***.

```
switch# clear isis neighbor interface et1
4 neighbors cleared on instance 1 switch#
```

- This command clears IS-IS adjacencies with a neighbor ***af86.3032.1a0f*** and on interface ***et1***.

```
switch# clear isis neighbor af86.3032.1a0f interface et1
2 neighbors cleared on instance 1 switch#
```

- This command clears all IS-IS adjacencies at Level 1 and on interface ***et1***.

```
switch# clear isis neighbor interface et1 level-1
2 neighbors cleared on instance 1 switch#
```

- This command clears Level 1-2 point-to-point adjacencies only.

```
switch# clear isis neighbor all level-1-2
0 neighbors cleared on instance 1 switch#
```

15.4.6.11 fast-reroute ti-lfa mode

Use the **`fast-reroute ti-lfa mode`** to enable link or node protection for node segments and adjacency segments of a specific address-family learned on all IS-IS interfaces.

Command Mode

address-family sub-mode of the router isis mode (config-router-isis-af)

Command Syntax

`fast-reroute ti-lfa mode` [[[link-protection | node-protection][level-1 | level-2]] | disabled]

Parameters

- **link-protection** Protects against the failure of the link.
- **node-protection** Protects against the failure of the neighbor mode.
- **level-1** Protects prefixes only in level-1.
- **level-2** Protects prefixes on in level-2. Disables the fast-reroute TI-LFA mode.

Guidelines

FRR using TI-LFA is disabled globally by default in the router IS-IS address-family sub-modes.

The interface TI-LFA configuration inherits the address-family sub-mode configuration by default.

15.4.6.12 fast-reroute ti-lfa srlg

Use the **`fast reroute ti-lfa srlg`** command to enable SRLG protection on all interfaces. This command is used in addition to configuring link-protection or node-protection. When SRLG protection is enabled, the backup paths are computed after excluding all the links that share the same SRLG with the active link that is being used by all prefix segments and adjacency segments.

Command Mode

IS-IS router address-family configuration mode

Command Syntax

fast-reroute ti-lfa srlg [strict]

Parameters

strict The backup path is only programmed if a backup path that excludes all the SRLGs configured on the primary interface.

15.4.6.13 graceful-restart (IS-IS)

The **graceful-restart** command configures IS-IS graceful-restart. The command provides options to configure the t2 time or the restart-hold-time.

t2 is the maximum wait time for the LSP database to synchronize (SPF computation is not done while t2 is running). t2 can be configured for either Level-1 or Level-2 routes.

restart-hold-time is the hold time advertised by the router to its neighbors before undergoing ASU2 fast reboot.

The **no graceful-restart** and **default graceful-restart** commands disables the IS-IS graceful-restart configuration from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

graceful-restart ***t2 | restart-hold-time value***

no graceful-restart ***t2 | restart-hold-time value***

default graceful-restart ***t2 | restart-hold-time value***

Parameters

- ***value*** The time in seconds. Value ranges from ***5*** to ***300*** seconds.
- ***restart-hold-time*** Sets the hold time when restarting.
- ***t2*** Sets the LSP database sync wait time.

Examples

- In this example an ISIS graceful restart is configured with t2 wait time of ***30*** seconds for Level-1 routes.

```
switch(config)# router isis 1
switch(config-router-isis)# graceful-restart t2 level-1 30
```

- In this example an ISIS graceful restart is configured with restart-hold-time of ***50*** seconds.

```
switch(config)# router isis 1
switch(config-router-isis)# graceful-restart restart-hold-time 50
```

15.4.6.14 is-hostname

The **is-hostname** command configures the use of a human-readable string to represent the symbolic name of an IS-IS router. It also changes the output of IS-IS show commands, to show the IS-IS hostname in place of system IDs if the corresponding IS-IS hostname is known. However, syslogs still use IS-IS system IDs and not the IS-IS hostname.

By default, if a hostname is configured on the switch, it is used as the IS-IS hostname. It is also possible to unconfigure an assigned hostname for IS-IS using the **no is-hostname** command. When the IS-IS hostname is removed, the switch goes back to using the switchs hostname as the IS-IS hostname.

Command Mode

Router-IS-IS Configuration

Command Syntax

is-hostname ***string***

no is-hostname

Examples

- These commands configure the IS-IS hostname to the symbolic name ***ishost1*** for the IS-IS router.

```
switch(config)# router isis inst1
switch(config-router-isis)# is-hostname ishost1
switch(config-router-isis)#
```

- These commands unconfigure the IS-IS hostname of the symbolic name ***ishost1*** for the IS-IS router.

```
switch(config)# router isis inst1
switch(config-router-isis)# no is-hostname ishost1
switch(config-router-isis)#
```

15.4.6.15 isis authentication key

The **isis authentication key** command configures the authentication key on the interface causing IS-IS Hellos to be authenticated.

The **no isis authentication mode** and **default isis authentication mode** commands disables the authentication key for the IS-IS instance.

Command Mode

Interface-Ethernet Configuration

Command Syntax

isis authentication key [***0*** | ***7***] **[LAYER_VALUE]**

no isis authentication key [***0*** | ***7***] **[LAYER_VALUE]**

default isis authentication key [***0*** | ***7***] **[LAYER_VALUE]**

Parameters

LAYER_VALUE Layer value. Options include:

- **level-1**
- **level-2**

Example

These commands configure authentication on the interface causing IS-IS Hellos to be authenticated.

```
switch(config)# interface Ethernet 3/6
switch(config-if-Et3/6)# isis authentication mode text
switch(config-if-Et3/6)# isis authentication key 7 cAm28+9a/xPi04o7hjd8Jw==
switch(config-if-Et3/6)#
```

15.4.6.16 isis authentication mode

The **isis authentication mode** command configures authentication on the interface causing IS-IS Hellos to be authenticated.

The **no isis authentication mode** and **default isis authentication mode** commands disables authentication for the IS-IS instance.

Command Mode

Interface-Ethernet Configuration

Command Syntax

isis authentication mode [md5 | ***text***][**LAYER_VALUE**]

no isis authentication mode [md5 | ***text***][**LAYER_VALUE**]

default isis authentication mode [md5 | ***text***][**LAYER_VALUE**]

Parameters

LAYER_VALUE Layer value. Options include:

- **level-1**
- **level-2**

Example

These commands configure authentication on the interface causing IS-IS Hellos to be authenticated.

```
switch(config)# interface Ethernet 3/6
switch(config-if-Et3/6)# isis authentication mode text
switch(config-if-Et3/6)# isis authentication key 7 cAm28+9a/xPi04o7hjd8Jw==
switch(config-if-Et3/6)#
```

15.4.6.17 isis bfd

The **isis bfd** command activates the corresponding IS-IS routing instance on the configuration mode interface. By default, the IS-IS routing instance is not enabled on an interface.

The **no isis enable** and **default isis enable** commands disable IS-IS on the configuration mode interface by removing the corresponding **isis enable** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

isis bfd

no isis bfd

default isis bfd

Example

These commands enable BFD on IS-IS interfaces.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# isis bfd
switch(config-if-Et5/6)#
```

15.4.6.18 isis enable

The **isis enable** command activates the corresponding IS-IS routing instance on the configuration mode interface. By default, the IS-IS routing instance is not enabled on an interface.

The **no isis enable** and **default isis enable** commands disable IS-IS on the configuration mode interface by removing the corresponding **isis enable** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`isis enable` ***instance_id***

`no isis enable`

`default isis enable`

Parameters

instance_id IS-IS instance name.

Examples

- These commands enable the IS-IS protocol on the ***interface ethernet 4***.

```
switch(config)# router isis Osiris
switch(config-router-isis)# net 49.0001.1010.1040.1030.00
switch(config-router-isis)# interface ethernet 4
switch(config-if-Eth4)# isis enable Osiris
```

- These commands disable the IS-IS protocol on the ***interface ethernet 4***.

```
switch(config)# interface ethernet 4
switch(config-if-Eth4)# no isis enable
```

15.4.6.19 isis fast-reroute ti-lfa mode

Use the **`isis fast-reroute ti-lfa mode`** command to enable link or node protection for node segments and adjacency segments learned on a specific IS-IS interface. By default, the interface TI-LFA configuration inherits the address-family sub-mode configuration.

The **`no isis fast-reroute ti-lfa mode`** and **`default isis fast-reroute ti-lfa mode`** commands disable link or node protection for node segments and adjacency segments learned on a specific IS-IS interface.

Command Mode

Interface configuration mode.

Command Syntax

`isis fast-reroute ti-lfa mode` [link-protection | node-protection | disabled][level-1 | level-2]

`no isis fast-reroute ti-lfa mode` [link-protection | node-protection | disabled][level-1 | level-2]

`default isis fast-reroute ti-lfa mode` [link-protection | node-protection | disabled][level-1 | level-2]

Parameters

- **link-protection** Configures link-protection.
- **node-protection** Configures node-protection.
- **disabled** Disables protection over the link.
- **level-1** Optional keyword in both the ***router isis address-family*** sub-mode and interface configuration mode CLIs is used to restrict protection to node segments and adjacency segments learned through either Level-1 topologies only.
- **level-2** Optional keyword in both the ***router isis address-family*** sub-mode and interface configuration mode CLIs is used to restrict protection to node segments and adjacency segments learned through Level-2 topologies only.

15.4.6.20 isis hello-interval

The **isis hello-interval** command sends Hello packets from applicable interfaces to maintain the adjacency through the transmitting and receiving of Hello packets. The Hello packet interval can be modified.

The **no isis hello-interval** and **default isis hello-interval** commands restore the default hello interval of ***10*** seconds on the configuration mode interface by removing the **isis hello-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

isis hello-interval ***time***

no isis hello-interval

default isis hello-interval

Parameters

time Values range from ***1*** to ***300***; default is ***10***.

Examples

- These commands configure a hello interval of ***45*** seconds for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# isis hello-interval 45
switch(config-if-Vl200)#
```

- These commands remove the configured hello interval of ***45*** seconds from ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# no isis hello-interval
switch(config-if-Vl200)#
```

- These commands configure a hello interval of ***60*** seconds for ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# isis hello-interval 60
switch(config-if-Et5)#
```

- These commands remove the configured hello interval of ***60*** seconds from ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no isis hello-interval
switch(config-if-Et5)#
```

15.4.6.21 isis hello-multiplier

The **isis hello-multiplier** command specifies the number of IS-IS hello packets missed by a neighbor before the adjacency is considered down.

The **no isis hello-multiplier** and **default isis hello-multiplier** commands restore the default hello interval of ***3*** on the configuration mode interface by removing the **isis hello-multiplier** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

isis hello-multiplier ***factor***

no isis hello-multiplier

default isis hello-multiplier

Parameters

factor Values range from ***3*** to ***100***; default is ***3***.

Examples

- These commands configure a hello multiplier of ***4*** for ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# isis hello-multiplier 4
switch(config-if-Vl200)#
```

- These commands remove the configured hello multiplier of ***4*** from ***vlan 200***.

```
switch(config)# interface vlan 200
switch(config-if-Vl200)# no isis hello-multiplier
switch(config-if-Vl200)#
```

- These commands configure a hello multiplier of ***45*** for ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# isis hello-multiplier 45
switch(config-if-Et5)#
```

- These commands remove the configured hello multiplier of ***45*** from ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no isis hello-multiplier
switch(config-if-Et5)#
```

15.4.6.22 isis [ipv4|ipv6] fast-reroute ti-lfa srlg

Use the **isis [ipv4|ipv6] fast-reroute ti-lfa srlg** command to enable protection selectively on a specific interface. This command only enables SRLG protection for prefix segments and adjacency segments enabled on the interface.

Command Mode

Interface configuration mode

Command Syntax

isis [ipv4 | ipv6][fast-reroute ti-lfa srlg][strict | disabled]

no isis [ipv4 | ipv6][fast-reroute ti-lfa srlg][strict | disabled

default isis [ipv4 | ipv6][fast-reroute ti-lfa srlg][strict | disabled]

Parameters

- **ipv4** IS-IS IPv4 interface configuration.
- **ipv6** IS-IS IPv6 interface configuration.
- **fast-reroute** Configures fast reroute.
- **ti-lfa** Configures TI-LFA FRR.
- **srlg** Excludes same SRLG links from backup path.
- **strict** The backup path is only programmed only if a backup path that excludes all the SRLGs configured on the primary interface. If **strict** is not provided and an SRLG excluding path is not available, TI-LFA programs the backup path that excluded the maximum number of SRLGs possible.
- **disabled** Use to selectively disable SRLG protection on an interface. This is useful when SRLG protection is enabled globally for all interfaces but needs to be selectively disabled for a specific interface.

15.4.6.23 isis ipv6 metric

The **`isis ipv6 metric`** command configures the IPv6 metric.

The **`no isis ipv6 metric`** and **`default isis ipv6 metric`** commands restore the default metric of 10 on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`isis ipv6 metric` ***metric_value***

`no isis ipv6 metric`

`default isis ipv6 metric`

Parameters

metric_value Values range from ***1*** to ***16777214***; default is ***10***.

Example

These commands configure the IPv6 metric.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# isis ipv6 metric 30
switch(config-if-Et5/6)#
```

15.4.6.24 isis lsp tx interval

The **`isis lsp tx interval`** command sets the interval at which IS-IS sends link-state information on the interface.

The **`no isis lsp tx interval`** and **`default isis lsp tx interval`** commands restores the default setting of ***33*** ms. by removing the **`isis lsp tx interval`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

`isis lsp tx interval` ***period***

`no isis lsp tx interval`

default isis lsp tx interval

Parameters

period Value ranges from ***1*** through ***3000***. Default interval is ***33*** ms.

Examples

- This command sets the LSP interval on interface ***interface ethernet 5*** to ***600*** milliseconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# isis lsp tx interval 600
switch(config-if-Et5)#
```

- This command removes the LSP interval on ***interface ethernet 5***.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no isis lsp tx interval
switch(config-if-Et5)#
```

15.4.6.25 isis metric

The **isis metric** command sets cost for sending information over an interface.

The **no isis metric** and **default isis metric** commands restore the metric to its default value of ***10*** by removing the **isis metric** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

isis metric ***metric_cost***

no isis metric

default isis metric

Parameters

metric_cost Values range from ***1*** to ***1677214***. Default value is ***10***.

Examples

- These commands configure a metric cost of ***30*** for sending information over ***interface ethernet 5***.

```
switch(config)# router isis Osiris
switch(config-router-isis)# interface ethernet 5
switch(config-if-Et5)# isis metric 30
switch(config-if-Et5)#
```

- These commands remove the configured metric cost of ***30*** from ***interface ethernet 5***.

```
switch(config)# router isis Osiris
switch(config-router-isis)# interface ethernet 5
switch(config-if-Et5)# no isis metric
switch(config-if-Et5)#
```

15.4.6.26 isis multi-topology

The **isis multi-topology** command configures the IPv4 or IPv6 address family individually on an interface with both IPv4 and IPv6 addresses.

The **no isis multi-topology** and **default isis multi-topology** commands restores the default interface to both IPv4 and IPv6 address families.

Command Mode

Interface-Ethernet Configuration

Command Syntax

isis multi-topology address-family ipv4 unicast

no isis multi-topology address-family ipv4 unicast

default isis multi-topology address-family ipv4 unicast

Examples

- These commands configure the IPv4 address family on an interface with both IPv4 and IPv6 addresses.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# isis multi-topology address-family ipv4 unicast
switch(config-if-Et5/6)#
```

- These commands configure the IPv6 address family on an interface with both IPv4 and IPv6 addresses.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# isis multi-topology address-family ipv6 unicast
switch(config-if-Et5/6)#
```

- These commands configure both the IPv4 and IPv6 address families on an interface.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# no isis multi-topology address-family unicast
switch(config-if-Et5/6)#
```

15.4.6.27 isis network

The **isis network** command sets the configuration mode interface as a point-to-point link. By default, interfaces are configured as broadcast links.

The **no isis network** and **default isis network** commands set the configuration mode interface as a broadcast link by removing the corresponding **isis network** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

isis network point-to-point

no isis network

default isis network

Examples

- These commands configure ***interface ethernet 10*** as a point-to-point link.

```
switch(config)# interface ethernet 10
switch(config-if-Et10)# isis network point-to-point
switch(config-if-Et10)#
```

- This command restores ***interface ethernet 10*** as a broadcast link.

```
switch(config-if-Et10)# no isis network
switch(config-if-Et10)#
```

15.4.6.28 isis passive

The **isis passive** command configures the configuration-mode interface as passive. The switch will continue to advertise the IP address in the LSP, but the interface will not send or receive IS-IS control packets.

The **no isis passive** command removes the passive configuration, allowing the interface to send and receive IS-IS control packets. The **default isis passive** command sets the interface to the default interface activity setting by removing the corresponding **isis passive** or **no isis passive** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

isis passive

no isis passive

default isis passive

Examples

- These commands configure ***interface ethernet 10*** as a passive interface.

```
switch(config)# interface ethernet 10
switch(config-if-Et10)# isis passive
switch(config-if-Et10)#
```

- These commands restore ***interface ethernet 10*** as an active interface.

```
switch(config)# interface ethernet 10
switch(config-if-Et10)# no isis passive
switch(config-if-Et10)#
```

15.4.6.29 isis priority

The **isis priority** command sets the IS-IS priority for the interface.

The default priority is ***64***. The network device with the highest priority will be elected as the designated intermediate router to send link-state advertisements for that network.

The **no isis priority** and **default isis priority** commands restore the default priority (***64***) on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

isis priority ***priority_level***

no isis priority

default isis priority

Parameters

priority_level Value ranges from ***0*** to ***127***. Default value is ***64***.

Examples

- These commands configure a IS-IS priority of ***60*** on **interface ethernet 5**.

```
switch(config)# router isis Osiris
switch(config-router-isis)# interface ethernet 5
switch(config-if-Et5)# isis priority 60
switch(config-if-Et5)#
```

- These commands restores the default IS-IS priority of ***64*** from ***interface ethernet 5***.

```
switch(config)# router isis Osiris
switch(config-router-isis)# interface ethernet 5
switch(config-if-Et5)# no isis priority
switch(config-if-Et5)#
```

- These commands configure the switch with a priority of ***64*** for ***interface vlan 7***.

```
switch(config)# interface vlan 7
switch(config-if-Vl7)# isis priority 64
switch(config-if-Vl7)#
```

- These command restores the default IS-IS priority of ***64*** for ***64***.

```
switch(config)# interface vlan 7
switch(config-if-Vl7)# no isis priority
switch(config-if-Vl7)#
```

15.4.6.30 is-type

The **is-type** command configures the routing level for an IS-IS instance.

An IS-IS router can be configured as Level-1-2 which can form adjacencies and exchange routing information with both Level-1 and Level-2 routers. A Level-1-2 router can be configured to transfer routing information from Level-1 to Level-2 areas and vice versa (via route leaking). By default, all routes from Level-1 area are always leaked into Level-2 network.

Command Mode

Router-IS-IS Configuration

Command Syntax

is-type LAYER_VALUE

Parameters

- **LAYER_VALUE** Layer value options include:
  - **level-1**
  - **level-1-2**
  - **level-2**

Examples

- These commands configure Level 1-2 routing.

```
switch(config)# router isis Osiris
switch(config-router-isis)# is-type level-1-2
switch(config-router-isis)#
```

- These commands configure Level 2 routing.

```
switch(config)# router isis Osiris
switch(config-router-isis)# is-type level-2
switch(config-router-isis)#
```

15.4.6.31 log-adjacency-changes (IS-IS)

The **log-adjacency-changes** command sets the switch to send Syslog messages when it detects link state changes or when it detects that a neighbor state has changed.

The default option is active when ***running-config*** does not contain any form of the command. Entering the command in any form replaces the previous command state in ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

log-adjacency-changes

no log-adjacency-changes

default log-adjacency-changes

Examples

- These commands configure the switch to send a Syslog message when a neighbor state changes.

```
switch(config)# router isis Osiris
switch(config-router-isis)# log-adjacency-changes
switch(config-router-isis)#
```

- These commands configure not to log the peer changes.

```
switch(config)# router isis Osiris
switch(config-router-isis)# no log-adjacency-changes
switch(config-router-isis)#
```

15.4.6.32 lsp dynamic flooding

Use the **lsp flooding dynamic** command to configure dynamic flooding. Dynamic flooding must be enabled on all roters in the area. The **no** form of the command removes LSP dynamic flooding. LSP flooding dynamic is disabled by default.

Command Mode

Router configuration mode

Command Syntax

lsp flood dynamic [level-1 | level-2]

no lsp flood dynamic [level-1 | level-2]

default lsp flood dynamic [level-1 | level-2]

Parameters

- **level-1** Level 1 adjeacencies.
- **level-2** Level 2 adjencencies.

Example

```
switch(config)# router isis Amun
switch(config-router-isis)# net 49.0000.0000.3333.00
switch(config-router-isis)# is-hostname ip3
switch(config-router-isis)# lsp flooding dynamic
```

15.4.6.33 match isis level

The **match isis level** command configures a route map to match on ISIS level. It filters the Level-1 or Level-2 routes by using route maps match statement.

The **no match isis level** and **default match isis level** commands disables the match ISIS level configuration from ***running-config***.

Command Mode

Route-map Configuration

Command Syntax

match isis level [level-1 | level-2]

no match isis level [level-1 | level-2]

default match isis level [level-1 | level-2]

Parameters

- **level-1** IS-IS level 1.
- **level-2** IS-IS level 2.

Example

These commands place the switch in ***route-map*** mode, and configures a route map to match isis level to Level-1.

```
switch(config)# route-map Test
switch(config-route-map-test)# match isis level level-1
```

15.4.6.34 mpls label range

The **mpls label range** command derives the indices of the actual MPLS label on the SRGB advertised by the router. The default value of SRGB in EOS is Base: ***900000***, Size: ***65536***. In other words, the labels that any global segment could represent is between ***900000-965535***.

Command Mode

Global Configuration

Command Syntax

mpls label range ***value***

Parameters

value Specifies the Segment Routing global range.

- **dynamic** Specifies labels reserved for dynamic assignment. Default value is (***100000***) (***262144***).
  - **IS-IS-sr** Specifies labels reserved for IS-IS SR global segment identifiers (SIDs). Default value is (***900000***) (***65536***).
  - **static** Specifies labels reserved for static MPLS routes. Default value is (***16***) (***99984***).

Example

The following command configures an IS-IS SR global range with a value of (***900000***)-- starting label range, (***65536***)--Numbers of labels to reserve.

```
switch(config)# mpls label range isis-sr 900000 65536
```

15.4.6.35 multi-topology

The **multi-topology** command configures IS-IS Multi-Topology (MT) support (disabled by default), enabling an IS-IS router to compute a separate topology for IPv4 and IPv6 links in the network. With MT configured, not all the links in a network need to support both IPv4 and IPv6. Some can support IPv4 or IPv6 individually. The IPv4 SPF will install IPv4 routes using the IPv4 topology, and similarly the IPv6 SPF will install IPv6 routes using the IPv6 topology. Without MT support, all links in an IS-IS network need to support the same set of address families. When MT is enabled, and each link has a separate IPv4 metric and IPv6 metric.

The **no multi-topology** and **default multi-topology** commands restores the default interface to both IPv4 and IPv6 address families.

Command Mode

Router IS-IS Address-Family Configuration

Command Syntax

multi-topology

no multi-topology

default multi-topology

Examples

- These commands configure MT for the IS-IS router.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv6 unicast
switch(config-router-isis-af)# multi-topology
switch(config-router-isis-af)#
```

- These commands unconfigure MT for the IS-IS router.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv6 unicast
switch(config-router-isis-af)# no multi-topology
switch(config-router-isis-af)#
```

15.4.6.36 net

The **net** command configures the Network Entity Title of the IS-IS instance. By default, no NET is defined.

The **no net** and **default net** commands removes the NET from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

net ***mask_hex***

no net

default net

Parameters

- ***maxk_hex*** Mask value. Format is hh.hhhh.hhhh.hhhh.hhhh.hhhh.hhhh.hhhh.hhhh.00.

Examples

- These commands specify the NET as ***49.0001.1010.1040.1030.00***, in which the system ID is ***1010.1040.1030***, area ID is ***49.0001***.

```
switch(config)# router isis Osiris
switch(config-router-isis)# net 49.0001.1010.1040.1030.00
switch(config-router-isis)#
```

- These commands remove NET ***49.0001.1010.1040.1030.00*** from **running-config**.

```
switch(config)# router isis Osiris
switch(config-router-isis)# no net 49.0001.1010.1040.1030.00
switch(config-router-isis)#
```

15.4.6.37 node-segment

The **node-segment** command associates the node segments with prefix mask length ***/32*** (IPV4) or ***/128*** (IPV6) addresses. The **node-segment** command must be issued on an IS-IS-enabled loop back interface.

Command Mode

Loop-back Interface Configuration

Command Syntax

node-segment [ipv4 | ipv6] ***index value***

Parameters

- **ipv4** Specifies the IPv4 node configuration.
- **ipv6** Specifies the IPv6 node configuration.
- ***index*** Node segment identifier.
- ***value*** Index to be mapped with IP prefix. Value ranges from ***0***-***65535***.

Examples

- The following commands are used to associate a node-segment with an IPv4 address.

```
switch(config)# int loopback 1
switch(config-if-Lo1)# ip address 21.1.1.1/32
switch(config-if-Lo1)# node-segment ipv4 index 5
```

- The following commands are used to associate a node-segment with an IPv6 address.

```
switch(config)# int loopback 1
switch(config-if-Lo1)# ipv6 add 2000::24/128
```

```
switch(config-if-Lo1)# node-segment ipv6 index 5
```

- The following example shows a warning thrown at the CLI when a ***/32*** or ***/128*** address is not configured on the interface.

```
switch(config)# int loopback 1
switch(config-if-Lo1)# ip address 21.1.1.1/24
switch(config-if-Lo1)# node-segment ipv4 index 1
! /32 IPv4 address is not configured on the interface
```

- The following command removes the node-segment from IS-IS SR from an interface.

```
switch(config-if-Lo1)# no node-segment ipv4 index 1
```

15.4.6.38 passive (IS-IS)

The **passive** command configures the specified IS-IS interface as passive. The switch will continue to advertise the IP address in the LSP, but the interface will not send or receive IS-IS control packets.s

The **no passive** command removes the passive configuration, allowing the interface to send and receive IS-IS control packets. The **default passive** command sets the interface to the default interface activity setting by removing the corresponding **passive** or **no passive** statement from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

passive INTERFACE_NAME

no passive INTERFACE_NAME

default passive INTERFACE_NAME

Parameters

INTERFACE_NAME Options include:

- **ethernet *e_range*** Ethernet interface list.
- **loopback *l_range*** loopback interface list.
- **port-channel *p_range*** channel group interface list.
- **vlan *v_range*** VLAN interface list.

Valid ***e_range***, ***l_range***, ***p_range***, and ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Examples

- These commands configure ***interface ethernet 10*** as a passive interface.

```
switch(config)# router isis Osiris
switch(config-router-isis)# passive ethernet 10
switch(config-router-isis)#
```

- These commands restore ***interface ethernet 10*** as an active IS-IS interface.

```
switch(config)# router isis Osiris
switch(config-router-isis)# no passive ethernet 10
switch(config-router-isis)#
```

15.4.6.39 prefix-segment

The **prefix-segment** command associates prefix segments with any IS-IS prefix a router is originating an IP Reachability TLV for.

Command Mode

Segment-Routing MPLS Configuration

Command Syntax

prefix-segment ***ip-address index value***

Parameters

- ***ip-address*** It can be IP address, or IP address with prefix, or an IPv6 address prefix.
- ***index*** Node segment identifier.
- ***value*** Index to be mapped with IP prefix. Value ranges from ***0***-***65535***.

Example

The following commands are used to associate a prefix segment with an IPv4 address with index value of ***50***.

```
switch(config)# router isis instance1
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)# prefix-segment 1.1.1.0/24 index 50
```

15.4.6.40 proxy-node-segment

The **proxy-node-segment** command configures a proxy-node-SID for a IS-IS prefix originating from the router that does not support IS-IS SR.

Command Mode

Segment-Routing MPLS Configuration

Command Syntax

proxy-node-segment ***ip-address index value***

Parameters

- ***ip-address*** It can be IP address, or IP address with prefix, or an IPv6 address prefix.
- ***index*** Node segment identifier.
- ***value*** Index to be mapped with IP prefix. Value ranges from ***0***-***65535***.

Example

A proxy-node-segment associates a ***/32*** or a ***/128*** route with an SID as shown below.

```
switch(config)# router isis instance1
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)# proxy-node-segment 1.1.1.0/32 index 50
```

15.4.6.41 redistribute (IS-IS)

The **redistribute** command redistributes the specified types of routes into IS-IS.

The **no redistribute** and **default redistribute** commands disable route redistribution from the specified domain by removing the corresponding **redistribute** statement from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

`redistribute` **ROUTE_TYPE**

`no redistribute` **ROUTE_TYPE**

`default redistribute` **ROUTE_TYPE**

Parameters

ROUTE_TYPE The route type for which routes are redistributed. These are the option to include.

- bgpredistribute BGP routes
- connectedredistribute connected routes
- ospfredistribute OSPF routes
- ospfv3redistribute OSPFv3 routes
- staticredistribute static routes

Examples

- These commands redistribute connected routes into the IS-IS domain.

```
switch(config)# router isis Test
switch(config-router-isis)# redistribute connected
```

- These commands redistribute static routes into the IS-IS domain.

```
switch(config)# router isis Test
switch(config-router-isis)# redistribute static
```

- These commands redistribute the BGP routes into ISIS domain in the ***address-family*** mode.

```
Switch(config)# router isis 1
Switch(config-router-isis)# address-family ipv4
Switch(config-router-isis-af)# redistribute bgp route-map bgp-to-isis-v4
```

- These commands redistribute the BGP routes into ISIS domain in the ***router-isis*** mode.

```
Switch(config)# router isis 1
Switch(config-router-isis)# redistribute bgp route-map bgp-to-isis
```

15.4.6.42 redistribute bgp route-map

The **`redistribute bgp route-map`** command redistributes the BGP routes from the specified route map into IS-IS. Only one route map can be specified; reissuing the command overrides any previous configuration.

The **`no redistribute bgp`** and **`default redistribute bgp`**commands disable BGP route redistribution from the specified domain by removing the **redistribute bgp** statement from ***running-config***.

The command is available in both ***router isis*** configuration mode and the ***address-family*** submode. The command is rejected if configured in both modes at the same time. Issuing the **no** or **default** command in ***router isis*** configuration mode has no effect on redistribution configured in the ***address-family*** submode.

Note: If the command is configured in an ***address-family*** submode, it only redistributes routes from that address family. If it is configured in ***router-isis*** mode, it applies to all enabled address families.

Command Mode

Router-IS-IS Configuration

Router-IS-IS Address-Family Configuration

Command Syntax

`redistribute bgp route-map` ***map_name***

no redistribute bgp

default redistribute ROUTE_TYPE

Parameter

map_name Route map to be used for redistribution of BGP routes.

Examples

- These commands redistribute IPv4 BGP routes from the route map called ***bgp-to-isis-v4*** into the ISIS domain.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv4
switch(config-router-isis-af)# redistribute bgp route-map bgp-to-isis-v4
switch(config-router-isis-af)#
```

- These commands redistribute all BGP routes from the route map ***bgp-to-isis*** into ISIS.

```
switch(config)# router isis 1
switch(config-router-isis)# redistribute bgp route-map bgp-to-isis
```

15.4.6.43 router isis

The **router isis** command places the switch in router ISIS configuration mode.

Router ISIS configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The **no router isis** command deletes the IS-IS instance.

The **exit** command returns the switch to ***global*** configuration mode.

Command Mode

Global Configuration

Command Syntax

router isis ***instance_name*** [**VRF_INSTANCE**]

no router isis ***instance_name***

default router isis ***instance_name***

Parameters

- ***instance_name*** routing instance.
- **VRF_INSTANCE**
  - ***no parameter***
  - **vrf** ***vrf_name***

Examples

- These commands places the switch in the ***router isis*** mode and creates an IS-IS routing instance named ***Osiris***.

```
switch(config)# router isis Osiris
switch(config-router-isis)#
```

- This command attempts to open an instance with a different routing instance name from that of the existing instance. The switch displays an error and stays in ***global*** configuration mode.

```
switch(config)# router isis Osiris
```

```
% More than 1 ISIS instance is not supported
switch(config)#
```

- This command deletes the IS-IS instance.

```
switch(config)# no router isis Osiris
switch(config)#
```

15.4.6.44 segment-routing mpls

The **segment-routing mpls** command places the switch in the ***segment-routing mpls*** configuration mode.

The **no segment-routing mpls** and **default segment-routing mpls** commands disable IS-IS SR and delete all IS-IS SR configurations.

Command Mode

Router IS-IS Configuration

Command Syntax

segment-routing mpls

no segment-routing mpls

default segment-routing mpls

Example

The following commands place the switch in ***segment-routing mpls*** configuration mode.

```
switch(config)# router isis instance1
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)#
```

15.4.6.45 show ip route

When services like LDP pseudowires, BGP LU, L2 EVPN, or L3 MPLS VPN use IS-IS SR tunnels as an underlay, these services are automatically protected by TI-LFA tunnels that protect the IS-IS SR tunnels. The **show ip route** command displays the hierarchy of the overlay-underlay-TI-LFA tunnels.

```
switch# show ip route
 B        2001:db8:3::/48 [200/0]
           via 2002::b00:301/128, IS-IS SR tunnel index 3, label
 122697
  via TI-LFA tunnel index 5, label imp-null(3)
  via fe80::200:76ff:fe03:0, Ethernet26/1, label imp-null(3)
  backup via fe80::200:76ff:fe01:0, Ethernet30/1, label 900002 900003
```

15.4.6.46 set isis level

The **set isis level** command configures a route map to set ISIS level.

The **no set isis level** and **default set isis level** commands disables the set ISIS level configuration from ***running-config***.

Command Mode

Route-map Configuration

Command Syntax

`set isis level` [level-1 | level-2 | level-1-2]

`no set isis level` [level-1 | level-2 | level-1-2]

`default set isis level` [level-1 | level-2 | level-1-2]

Parameters

- **level-1** IS-IS level 1.
- **level-2** IS-IS level 2.
- **level-1-2** IS-IS level 1 and level 2.

Example

These commands place the switch in the ***route-map*** mode, and configures a route map to set isis level to **level-1**.

```
switch(config)# route-map Test
switch(config-route-map-test)# set isis level level-1
```

15.4.6.47 set-overload-bit

The **`set-overload-bit`** command sets the overload bit in link state packets (LSPs) to signal that the switch is not available for forwarding transit traffic (for instance, during startup or when the switch is being taken down for maintenance). To configure the switch to set the overload bit for a specified period after a reboot, use the **on-startup** option.

The **`no set-overload-bit`** and **`default set-overload-bit`** commands remove the corresponding **`set-overload-bit`** command from ***running-config***.

Note: When using the **on-startup** option, the overload bit will remain set in LSPs until the IS-IS agent has been up for the configured interval.

Command Mode

Router-IS-IS Configuration

Command Syntax

`set-overload-bit` [**on-startup** *interval*]

`no set-overload-bit`

`default set-overload-bit`

Parameters

- **on-startup** Configures the switch to set the overload bit in LSPs for a period of *interval* seconds after startup.
- *interval* The period in seconds for which the overload bit remains set after startup.

Examples

- These commands configure the switch to sets the overload bit for ***120*** seconds after startup.

```
switch(config)# router isis Osiris
switch(config-router-isis)# set-overload-bit on-startup 120
switch(config-router-isis)#
```

- These commands remove the configured overload bit of ***120*** seconds from the ***running-config***.

```
switch(config)# router isis Osiris
switch(config-router-isis)# no set-overload-bit on-startup
```

```
switch(config-router-isis)#
```

15.4.6.48 show isis database

The **show isis database** command displays the link state database of IS-IS. The default command displays active routes and learned routes.

Command Mode

EXEC

Command Syntax

show isis database [**INSTANCES**][**INFO_LEVEL**]

show isis database [**INFO_LEVEL**] [**VRF_INSTANCE**]

Parameters

- **INSTANCES** Options include:
  - ***no parameter***
  - ***instance_name***
- **INFO_LEVEL** Options include:
  - ***no parameter***
  - **detail**
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter***
  - **vrf *vrf_name***

Display Values

- **ISIS Instance**
- **LSPID**
- **Seq Num**
- **Cksum**
- **Life**
- **IS**

Examples

- This command displays general information about the link state database of IS-IS.

```
switch# show isis database

ISIS Instance: Osiris
  ISIS Level 2 Link State Database
    LSPID                 Seq Num   Cksum  Life  IS Flags
    1212.1212.1212.00-00  4         714    1064  L2 <>
    1212.1212.1212.0a-00  1         57417  1064  L2 <>
    2222.2222.2222.00-00  6         15323  1116  L2 <>
    2727.2727.2727.00-00  10        15596  1050  L2 <>
    3030.3030.3030.00-00  12        62023  1104  L2 <>
    3030.3030.3030.c7-00  4         53510  1104  L2 <>
switch>
```

- This command displays detailed information about the link state database of IS-IS.

```
switch# show isis database detail

ISIS Instance: Osiris
  ISIS Level 2 Link State Database
```

```
LSPID                 Seq Num   Cksum  Life  IS Flags
1212.1212.1212.00-00  4         714    1060  L2 <>
  Area address: 49.0001
  Interface address: 10.1.1.2
  Interface address: 2002::2
  IS Neighbor:   1212.1212.1212.0a Metric: 10
  Reachability: 10.1.1.0/24 Metric: 10 Type: 1
  Reachability: 2002::/64 Metric: 10 Type: 1
1212.1212.1212.0a-00  1         57417  1060  L2 <>
  IS Neighbor:   2727.2727.2727.00 Metric: 0
  IS Neighbor:   2222.2222.2222.00 Metric: 0
  IS Neighbor:   1212.1212.1212.00 Metric: 0
2222.2222.2222.00-00  6         15323  1112  L2 <>
  Area address: 49.0001
  Interface address: 10.1.1.1
  Interface address: 10.1.1.3
  Interface address: 2002::3
  IS Neighbor:   1212.1212.1212.0a Metric: 10
  Reachability: 10.1.1.0/24 Metric: 10 Type: 1
  Reachability: 10.1.1.0/24 Metric: 10 Type: 1
  Reachability: 2002::/64 Metric: 10 Type: 1
2727.2727.2727.00-00  10        15596  1046  L2 <>
  Area address: 49.0001
  Interface address: 10.1.1.1
  Interface address: 30.1.1.1
  Interface address: 2002::1
  Interface address: 2001::1
  IS Neighbor:   1212.1212.1212.0a Metric: 10
  IS Neighbor:   3030.3030.3030.c7 Metric: 10
  Reachability: 10.1.1.0/24 Metric: 10 Type: 1
  Reachability: 30.1.1.0/24 Metric: 10 Type: 1
  Reachability: 2002::/64 Metric: 10 Type: 1
  Reachability: 2001::/64 Metric: 10 Type: 1
3030.3030.3030.00-00  12        62023  1100  L2 <>
  Area address: 49.0001
  Interface address: 30.1.1.2
  Interface address: 2001::2
  IS Neighbor:   3030.3030.3030.c7 Metric: 10
  Reachability: 12.1.1.0/24 Metric: 1 Type: 1
  Reachability: 110.1.1.0/24 Metric: 0 Type: 1
  Reachability: 30.1.1.0/24 Metric: 10 Type: 1
  Reachability: 2001::/64 Metric: 10 Type: 1
3030.3030.3030.c7-00  4         53510  1100  L2 <>
  IS Neighbor:   2727.2727.2727.00 Metric: 0
  IS Neighbor:   3030.3030.3030.00 Metric: 0 switch>
```

15.4.6.49 show isis database detail

The **`show isis database detail`** command displays a view of LSPDB of different devices in the IS-IS domain.

Command Mode

EXEC

Command Syntax

`show isis database detail`

Example

The command output displays the TLVs and sub-TLVs that are being self-originated or the ones that have been received from other routers.

```
switch# show isis database detail

ISIS Instance: inst1 VRF: default
 ISIS Level 2 Link State Database
 LSPID Seq Num Cksum Life IS Flags
 1111.1111.1001.00-00 10 63306 751 L2 <>
 NLPID: 0xCC(IPv4) 0x8E(IPv6)
 Area address: 49.0001
 Interface address: 1.0.7.1
 Interface address: 1.0.0.1
 Interface address: 2000:0:0:47::1
 Interface address: 2000:0:0:40::1
 IS Neighbor : lf319.53 Metric: 10
    LAN-Adj-sid: 100000 flags: [ L V ] weight: 0 system ID: 1111.1111.1002
 IS Neighbor (MT-IPv6): lf319.53 Metric: 10
    LAN-Adj-sid: 100001 flags: [ L V F ] weight: 0 system ID: 1111.1111.1002
 Reachability : 1.0.11.0/24 Metric: 1 Type: 1 Up
    SR Prefix-SID: 10 Flags: [ R ] Algorithm: 0
 Reachability : 1.0.3.0/24 Metric: 1 Type: 1 Up
 Reachability : 1.0.7.1/32 Metric: 10 Type: 1 Up
    SR Prefix-SID: 2 Flags: [ N ] Algorithm: 0
 Reachability : 1.0.0.0/24 Metric: 10 Type: 1 Up
 Reachability (MT-IPv6): 2000:0:0:4b::/64 Metric: 1 Type: 1 Up
    SR Prefix-SID: 11 Flags: [ R ] Algorithm: 0
 Reachability (MT-IPv6): 2000:0:0:43::/64 Metric: 1 Type: 1 Up
 Reachability (MT-IPv6): 2000:0:0:47::1/128 Metric: 10 Type: 1 Up
    SR Prefix-SID: 3 Flags: [ N ] Algorithm: 0
 Reachability (MT-IPv6): 2000:0:0:40::/64 Metric: 10 Type: 1 Up
 Router Capabilities: 252.252.1.252 Flags: [ ]
    SR Capability: Flags: [ I V ]
       SRGB Base: 900000 Range: 65536
 Segment Binding: Flags: [ F ] Weight: 0 Range: 1 Pfx 2000:0:0:4f::1/128
    SR Prefix-SID: 19 Flags: [ ] Algorithm: 0
 Segment Binding: Flags: [ ] Weight: 0 Range: 1 Pfx 1.0.15.1/32
    SR Prefix-SID: 18 Flags: [ ] Algorithm: 0
```

15.4.6.50 show isis dynamic flooding

Use the **show isis dynamic flooding** command to monitor Dynamic Flooding.

Command Mode

EXEC

Command Syntax

show isis dynamic flooding [interfaces | level-1 | level-2 | nodes | paths | topology | interface]

Parameters

- **interfaces** Flooding interfaces
- **level-1** Level 1 adjencencies only.
- **level-2** Level 2 adjencencies only.
- **nodes** Nodes in the flooding topology.
- **paths** Paths in the flooding topology.
- **topology** Flooding topology.

Examples

- The command **show isis dynamic flooding nodes** shows the list of nodes in the area and the indices for the nodes.

```
switch# show isis dynamic flooding nodes
IS-IS Instance: Amun VRF: default
   Level 1 Nodes:
     Index           Node ID
     0               ip6.00
     1               ip4.00
     2               ip2.00
     3               ip1.00
     4               ip3.00
     5               ip5.00
```

- The command **show isis dynamic flooding paths** shows the list of paths in the flooding topology using node indices.

```
switch# show isis dynamic flooding paths
IS-IS Instance: Amun VRF: default
  Level 1:
    Path: 0 1 2 3 4 5 0
```

- To view the flooding topology, use the **show isis dynamic flooding topology** command:

```
switch# show isis dynamic flooding topology
IS-IS Instance: Amun VRF: default
  Level 1:
    Path: ip6.00 ip4.00 ip2.00 ip1.00 ip3.00 ip5.00 ip6.00
```

- To view which interfaces dynamic flooding will use, use the **show isis dynamic flooding interfaces** command:

```
switch# show isis dynamic flooding interfaces
IS-IS Instance: Amun VRF: default
  Level 1:
    Ethernet5
    Ethernet4
```

15.4.6.51 show isis graceful-restart vrf

The **show isis graceful-restart vrf** command displays the GR configuration and graceful-restart related state of the IS-IS instance as well as its neighbors.

Command Mode

EXEC

Command Syntax

show isis graceful-restart vrf ***vrf-name***

Example

In this example the show isis graceful-restart command displays the output for the default vrf instance.

```
switch# show isis graceful-restart vrf default
IS-IS Instance: 1 VRF: default
 System ID: 0000.0000.0001
 Graceful Restart: Enabled, Graceful Restart Helper: Enabled
 State: Last Start exited after T2 (level-1) expiry
 T1 : 3s
 T2 (level-1) : 30s/20s remaining
 T2 (level-2) : 30s/not running
```

```
 T3 : not running

System ID       Type   Interface     Restart Capable  Status
is-hostname-1   L1L2   Ethernet1     Yes              Running
is-hostname-2   L1     Ethernet2     Yes              Restarting
```

15.4.6.52 show isis hostname

The **show isis hostname** command displays mapping between the System ID and IS-IS hostname.

Command Mode

EXEC

Command Syntax

show isis hostname

Example

This command mapping between the System ID and IS-IS hostnames ***host1*** and ***host2***.

```
switch# show isis hostname
ISIS Instance: 1 VRF: default
Level System ID Hostname
L1 1111.1111.1001 host1
L1 1111.1111.1002 host2
```

15.4.6.53 show isis interface

The **show isis interface** command displays interface information for the IS-IS instance.

Command Mode

EXEC

Command Syntax

show isis interface [**INSTANCES**][**INTERFACE_NAME**][**INFO_LEVEL**]

show isis interface [**INTERFACE_NAME**] [**INFO_LEVEL**][**VRF_INSTANCE**]

Parameters

- **INSTANCES** Options include:
  - ***no parameter***
  - ***instance_name***
- **INTERFACE_NAME** Values include:
  - ***no parameter*** all interfaces.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port channel interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- **INFO_LEVEL** Options include:
  - ***no parameter***
  - **detail**
- **VRF_INSTANCE** specifies the VRF instance.

- ***no parameter***
- **vrf *vrf_name***

Display Values

- **ISIS Instance**
- **System ID**
- **Index**
- **MTU**
- **Metric**
- **LAN-ID**
- **DIS**
- **Type**
- **Interface**
- **SNPA**
- **State**
- **Hold time**

Examples

- This command displays general IS-IS information for instance ***Osiris***.

```
switch# show isis interface

ISIS Instance: Osiris
  Interface Vlan20:
    Index: 59 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 2
      LAN-ID: 1212.1212.1212, Priority: 64
      DIS: 1212.1212.1212, DIS Priority: 64
  Interface Ethernet30:
    Index: 36 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 1
      LAN-ID: 3030.3030.3030, Priority: 64
      DIS: 3030.3030.3030, DIS Priority: 64
```

- This command displays detailed IS-IS information for instance ***Osiris***.

```
switch# show isis interface detail

ISIS Instance: Osiris
  Interface Vlan20:
    Index: 59 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 2
      LAN-ID: 1212.1212.1212, Priority: 64
      DIS: 1212.1212.1212, DIS Priority: 64
    Adjacency 2222.2222.2222:
      State: UP, Level: 2 Type: Level 2 IS
      Hold Time: 30, Supported Protocols: ipv4, ipv6
      SNPA: 2:1:0:c:0:0, Priority: 64
      IPv4 Interface Address: 10.1.1.3
      IPv6 Interface Address: fe80::1:ff:fe0c:0
      Areas:
        49.0001
    Adjacency 1212.1212.1212:
```

```
      State: UP, Level: 2 Type: Level 2 IS
      Hold Time: 9, Supported Protocols: ipv4, ipv6
      SNPA: 2:1:0:d:0:0, Priority: 64
      IPv4 Interface Address: 10.1.1.2
      IPv6 Interface Address: fe80::1:ff:fe0d:0
      Areas:
        49.0001
  Interface Ethernet30:
    Index: 36 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 1
      LAN-ID: 3030.3030.3030, Priority: 64
      DIS: 3030.3030.3030, DIS Priority: 64
    Adjacency 3030.3030.3030:
      State: UP, Level: 2 Type: Level 2 IS
      Hold Time: 9, Supported Protocols: ipv4, ipv6
      SNPA: 2:1:0:b:0:0, Priority: 64
      IPv4 Interface Address: 30.1.1.2
      IPv6 Interface Address: fe80::1:ff:fe0b:0
      Areas:
        49.0001
```

- This example displays the state of TI-LFA protection for IPv4/IPV6 prefixes learned on that IS-IS interface.

```
switch# show isis interface Vlan2387

IS-IS Instance: inst1 VRF: default

  Interface Vlan2387:
    Index: 36 SNPA: P2P
    MTU: 1497 Type: point-to-point
    BFD IPv4 is Disabled
    BFD IPv6 is Disabled
    Hello Padding is Enabled
    Level 2:
      Metric: 10, Number of adjacencies: 1
      Link-ID: 24
      Authentication mode: None
      TI-LFA node protection with SRLG loose protection is enabled for the
 following IPv4 segments: node segments, adjacency segments
      TI-LFA protection is disabled for IPv6
```

15.4.6.54 show isis local-convergence-delay

The **`show isis local-convergence-delay`** command shows the current or last attempt at delaying the convergence of protected routes on a link down/BFD neighbor down event. If the timer aborts for some reason (such as a topology change causing a new SPF), the attempt fails.

```
switch# show isis local-convergence-delay

IS-IS Instance: inst1 VRF: default
  System ID: 1111.1111.1001
  IPv4 local convergence delay configured, 5000 msecs
  IPv6 local convergence delay configured, 5000 msecs
  Level 1 attempts 0, failures 0
  Level 2 attempts 3, failures 1

  Level 2 in progress due to LINK DOWN on Vlan2138
    TI-LFA node protection is enabled for IPv4
```

```
    IPv4 Routes delayed: 0
      Delay timer started at: 2019-07-25 23:16:33
      Delay timer expires in 2 secs
    TI-LFA protection is disabled for IPv6

  Level 2 last attempt due to LINK DOWN on Vlan2138, Succeeded
    TI-LFA node protection is enabled for IPv4
    IPv4 Routes delayed: 3
      Delay timer started at: 2019-07-25 23:14:51
      Delay timer stopped at: 2019-07-25 23:14:56
    TI-LFA protection is disabled for IPv6
```

The **detail** keyword also lists all the routes that have been delayed.

```
switch# show isis local-convergence-delay detail
  ...
  Level 2 last attempt due to LINK DOWN on Vlan2138, Succeeded
    TI-LFA node protection is enabled for IPv4
    IPv4 Routes delayed: 3
      Delay timer started at: 2019-07-25 23:14:51
      Delay timer stopped at: 2019-07-25 23:14:56
      Delayed routes:
        10.0.7.1/32
        10.0.9.1/32
        10.0.10.1/32
    TI-LFA protection is disabled for IPv6
```

15.4.6.55 show isis neighbors

The **`show isis neighbors`** command displays IS-IS neighbor information.

Command Mode

EXEC

Command Syntax

`show isis neighbors` [**INSTANCES**] [**INFO_LEVEL**]

`show isis neighbor` [**INFO_LEVEL**] [**VRF_INSTANCE**]

Parameters

- **INSTANCES** Options include:
  - ***no parameter***
  - ***instance_name***
- **INFO_LEVEL** Options include:
  - ***no parameter***
  - **detail**
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter***
  - **vrf *vrf_name***

Display Values

- **`Inst. ID`**

- **System ID**
- **Type**
- **Interface**
- **SNPA**
- **State**
- **Hold time**
- **Area Address**

Example

This command displays general information about the IS-IS neighbors.

```
switch(config)# show isis neighbors

Inst Id   System Id           Type Interface       SNPA              State Hold time
10        2222.2222.2222      L2   Vlan20          2:1:0:c:0:0       UP    30
10        1212.1212.1212      L2   Vlan20          2:1:0:d:0:0       UP    9
10        3030.3030.3030      L2   Ethernet30      2:1:0:b:0:0       UP    9
switch(config)#
```

15.4.6.56 show isis network topology

The **show isis network topology** command displays a list of all IS-IS devices that are reachable in the network.

Command Mode

EXEC

Command Syntax

show isis network topology

show isis INSTANCES network topology

show isis network topology VRF_INSTANCE

Parameters

- **INSTANCES** Options include:
  - ***no parameter***
  - ***instance_name***
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter***
  - **vrf *vrf_name***

Display Values

- **System Id**
- **Metric**
- **Next-Hop**
- **Interface**
- **SNPA**

Example

This command displays the list of all devices reachable in the network.

```
switch# show isis network topology

IS-IS Instance: Osiris VRF: default
  IS-IS paths to level-2 routers
    System Id        Metric   IA Metric Next-Hop         Interface     SNPA
```

```
    2222.2222.2222   10       0         2222.2222.2222   Ethernet1     P2P switch>
```

15.4.6.57 show isis segment-routing adjacency-segments

The **show isis segment-routing adjacency-segments** command displays the global adjacency SID value and other related information.

Command Mode

EXEC

Command Syntax

show isis segment-routing adjacency-segments

Examples

- In this example the **show isis segment-routing adjacency-segments** command displays the output for the interface configured like this:

```
interface Ethernet1
 ip address 1.1.1.1/24
 ipv6 address 1000::1/64
 isis enable isis1
 isis network point-to-point
 adjacency-segment ipv4 p2p index 1 global
 adjacency-segment ipv6 p2p index 2 global
```

- The show output for the above interface configuration:

```
switch# show isis segment-routing adjacency-segments

System ID: 1000.0000.0002                       Instance: isis1
SR supported Data-plane: MPLS                   SR Router ID: 1.1.1.4
Adj-SID allocation mode: SR-adjacencies
Adj-SID allocation pool: Base: 100000     Size: 16384
Adjacency Segment Count: 2
Flag Descriptions: F: Ipv6 address family, B: Backup, V: Value
                   L: Local, S: Set

Segment Status codes: L1 - Level-1 adjacency, L2 - Level-2 adjacency, P2P -
Point-to-Point adjacency, LAN - Broadcast adjacency

Locally Originated Adjacency Segments
Adj IP Address    Local Intf   SID     SID Source     Flags                Type
---------------  ----------   ------  -------------  -------------------  -------
1.1.1.2           Et1          1       Configured    F:0 B:0 V:0 L:0 S:0  P2P L1
fe80::1:ff:fe65:0  Et1          2       Configured    F:1 B:0 V:0 L:0 S:0  P2P L1


Received Global   Adjacency Segments
SID               Originator              Neighbor           Flags
---------         --------------------    ----------------   --------------------
0                 rtrmpls1                1000.0000.0002      F:0 B:0 V:0 L:0 S:0
```

- The following is the C-API output for the **show isis segment-routing adjacency-segments** command.

```
switch# show isis segment-routing adjacency-segments | json
{
   "vrfs": {
       "default": {
           "isisInstances": {
               "isis1": {
                   "routerId": "1.1.1.4",
                   "adjSidPoolSize": 16384,
                   "receivedGlobalAdjacencySegments": [
                       {
                           "systemId": "1000.0000.0001",
```

```
                            "hostname": "rtrmpls1",
                           "sid": 0,
                           "flags": {
                               "s": false,
                               "b": false,
                               "v": false,
                               "f": false,
                               "l": false
                           },
                           "nbrSystemId": "1000.0000.0002"
                       }
                   ],
                   "systemId": "1000.0000.0002",
                   "adjSidAllocationMode": "SrOnly",
                   "dataPlane": "MPLS",
                   "adjacencySegments": [
                       {
                           "lan": false,
                           "sidOrigin": "configured",
                           "flags": {
                               "s": false,
                               "b": false,
                               "v": true,
                               "f": false,
                               "l": false
                           },
                           "sid": 1,
                           "localIntf": "Ethernet1",
                           "ipAddress": "1.1.1.2",
                           "level": 1
                       },
                        {
                           "lan": false,
                           "sidOrigin": "configured",
                           "flags": {
                               "s": false,
                               "b": false,
                               "v": false,
                               "f": true,
                               "l": false
                           },
                           "sid": 2,
                           "localIntf": "Ethernet1",
                           "ipAddress": "fe80::1:ff:fe65:0",
                           "level": 1
                       }
                   ],
                   "adjSidPoolBase": 100000,
                   "misconfiguredAdjacencySegments": []
               }
           }
       }
   }
```


- ```
  switch# show isis segment-routing adjacency-segments
  ...
  Locally Originated Adjacency Segments
   Adj IP Address      Local Intf     SID          Flags                 Protection
  -----------------    ----------     --------     --------------------- ------------
   10.1.0.1            Vl2138         100001       F:0 B:1 V:1 L:1 S:0    node
   10.1.0.2            Vl2968         100002       F:0 B:1 V:1 L:1 S:0    node with SRLG loose
   10.1.0.3            Vl2387         965537       F:0 B:1 V:1 L:1 S:0    node with SRLG strict


  Received Global Adjacency Segments
  SID       Originator            Neighbor            Flags                     Protection
  ```
  ```

```
--------- -------------------- -------------------- ------------------------- ----------
5         1111.1111.1005       1111.1111.1004       F:0 B:1 V:0 L:0 S:0       node
```

15.4.6.58 show isis segment-routing global-blocks

The **show isis segment-routing global-blocks** command lists the SRGBs in use by all SR supporting devices in IS-IS domain including the SRGB in use by IS-IS SR on this device.

Command Mode

EXEC

Command Syntax

show isis segment-routing global-blocks

Example

```
switch# show isis segment-routing global-blocks
System ID: 1111.1111.1002              Instance: inst1
SR supported Data-plane: MPLS          SR Router ID: 252.252.2.252
SR Global Block( SRGB ): Base: 900000  Size: 65536
Number of ISIS segment routing capable peers: 3
SystemId             Base         Size
-------------------- ------------ -----
1111.1111.1002       900000       65536
1111.1111.1001       900000       65536
```

15.4.6.59 show isis segment-routing prefix-segments

The **show isis segment-routing prefix-segments** command provides the details of all prefix segments being originated as well the segments received from IS-IS SR speakers in the domain.

Command Mode

EXEC

Command Syntax

show isis segment-routing prefix-segments

Example

```
switch# show isis segment-routing prefix-segments
System ID: 1111.1111.1002         Instance: inst1
SR supported Data-plane: MPLS     SR Router ID: 252.252.2.252
Node: 2 Proxy-Node: 2 Prefix: 2 Total Segments: 6
Flag Descriptions: R: Re-advertised, N: Node Segment, P: no-PHP
                   E: Explicit-NULL, V: Value, L: Local
Segment status codes: * - Self originated Prefix, L1 - level 1, L2 - level 2
    Prefix     SID     Type        Flags                    SystemID         Type
-------------------- ---------    ----------------------  ---------------  -----
 1.0.7.1/32      2     Node        R:0 N:1 P:0 E:0 V:0 L:0   1111.1111.1001   L1
* 1.0.8.1/32      4     Node        R:0 N:1 P:0 E:0 V:0 L:0   1111.1111.1002   L2
 1.0.11.0/24     10     Prefix      R:1 N:0 P:0 E:0 V:0 L:0   1111.1111.1001   L2
* 1.0.12.0/24    12     Prefix      R:1 N:0 P:0 E:0 V:0 L:0   1111.1111.1002   L2
 1.0.15.1/32     18     Proxy-Node  R:0 N:0 P:0 E:0 V:0 L:0   1111.1111.1001   L2
 1.0.16.1/32     20     Proxy-Node  R:0 N:0 P:0 E:0 V:0 L:0   1111.1111.1003   L2
```

```
switch# show isis segment-routing prefix-segments
...
  Prefix            SID   Type          System ID       Level  Protection
  -------------     ----- ------ ...    --------------- ------ -----------
* 10.1.1.1/32        0    Node   ...    1111.1111.1001   L2    unprotected
  10.1.1.2/32        1    Node   ...    1111.1111.1002   L2    node with SRLG loose
  10.1.1.3/32        4    Node   ...    1111.1111.1005   L2    node with SRLG strict
  10.1.1.4/32        10   Prefix ...    1111.1111.1004   L1    node
```

About the Output

After the usual output header that represents the system ID, instance name, etc and parameters of a router, there is a line depicting prefix segment counters. Each field in this line relates to the number of segments that are present in this routers IS-IS instance. For example, the above example shows that this device has 2 Node Segments (Self originated as well as the ones received from other IS-IS SR devices).

The main section of this show commands output is the section that lists all the prefix segments and related information like prefix, SID, type of segment (Prefix, Node, Proxy-Node), the flag values being carried in the sub-TLVs of these prefix segments and the system ID of the originating router. The Type field will be useful on a IS type level-1-2 router. It shows whether the installed prefix segment is from a level-1 prefix or a level-2 prefix.

15.4.6.60 show isis segment-routing

The **`show isis segment-routing`** command displays the summary information on IS-IS SR status.

Command Mode

EXEC

Command Syntax

`show isis segment-routing`

Example

The command output displays the summary information on IS-IS SR status.

```
switch(config)# show isis segment-routing
System ID: 1111.1111.1002               Instance: inst1
SR supported Data-plane: MPLS           SR Router ID: 252.252.2.252
SR Global Block( SRGB ): Base: 900000   Size: 65536
Adj-SID allocation mode: SR-adjacencies
Adj-SID allocation pool: Base: 100000     Size: 16384
All Prefix Segments have    : P:0 E:0 V:0 L:0
All Adjacency Segments have : F:0 B:0 V:1 L:1 S:0
ISIS Reachability Algorithm : SPF (0)
Number of ISIS segment routing capable peers: 3
Self-Originated Segment Statistics:
Node-Segments       : 2
Prefix-Segments     : 2
Proxy-Node-Segments : 0
Adjacency Segments :
```

About the Output

The first line of the output shows the IS-IS system ID of this device and the name of the instance with which IS-IS is configured.

The supported data plane is shown against the SR supported Data-plane field, while the router ID being advertised in the Router Capability is mentioned in the SR Router ID field.

The SRGB in use and the MPLS label pool being used for adjacency segment allocation are mentioned in this output. The current adjacency allocation mode which refers to whether we are allocating adjacency segments to all IS-IS adjacencies or only those adjacencies which support SR or None of the adjacencies is shown in the Adj-SID allocation mode field.

Flag contents of All Prefix Segments originated on this router, Flag contents of All Adjacency Segments originated on this router and supported IS-IS Reachability Algorithm have been provided through this command output and they carry the meaning as per the IS-IS SR IETF draft.

This show command provides a statistics related to IS-IS SR in terms of various counters ranging from number of IS-IS SR enabled peers, number of Node-SIDs, prefix-SIDs, proxy-node-segments and adjacency segments being originated on this router in IS-IS.

The **show isis segment-routing** command also provides information if segment routing has been administratively disabled as shown.

```
switch(config-router-isis-sr-mpls)# show isis segment-routing
! IS-IS (Instance: inst1) Segment Routing has been administratively shutdown.
```

15.4.6.61 show isis segment-routing tunnel

The **show isis segment-routing tunnel** command displays all the IS-IS SR tunnels. The field **TI-LFA tunnel index** displays the index of the TI-LFA tunnel protecting the SR tunnel. The same TI-LFA tunnel that protects the LFIB route also protects the corresponding IS-IS SR tunnel.

```
switch#show isis segment-routing tunnel 10.0.10.1/32
Index   Endpoint        Nexthop    Interface   Labels     TI-LFA
                                                          tunnel index
------ --------------- ----------- ----------- ---------- -------------
4       10.0.10.1/32    10.0.0.2   Vlan2387    [900004]   0
```

15.4.6.62 show isis summary

The **show isis summary** command displays information about the configured IS-IS instances.

Command Mode

EXEC

Command Syntax

show isis summary

show isis [**INSTANCES**] summary

show isis summary **VRF_INSTANCE**

Parameters

- **INSTANCES** Options include:
  - ***no parameter***
  - ***instance_name***
- **VRF_INSTANCE** Specifies the VRF instance.
  - ***no parameter***
  - **vrf** ***vrf_name***

Display Values

- **System ID**
- **IPv4 Preference**
- **IPv6 Preference**
- **IS-Types**
- **LSP Generation interval**
- **SPF Interval**
- **Current SPF Hold Interval**
- **IS-Types Run Time**

- **Area Addresses**
- **Designated Intermediate Systems (DIS) Interfaces**
- **Link State DataBase (LSDB) size**

Display Status

- **Multi Topology**
- **Authentication Mode**
- **Graceful Restart**
- **Graceful Restart Helper**

Example

This command displays general information about the configured IS-IS instances.

```
switch(config-router-isis-af)# show isis summary

IS-IS Instance: 1 VRF: default
 System ID: 0000.0000.0001, administratively enabled
 Multi Topology disabled, not attached
 IPv4 Preference: Level 1: 115, Level 2: 115
 IPv6 Preference: Level 1: 115, Level 2: 115
 IS-Type: Level 1 and 2, Number active interfaces: 0
 Routes both IPv4 and IPv6
 LSP size maximum: Level 1: 9000, Level 2: 9000
                          Max wait(s) Initial wait(ms) Hold interval(ms)
 LSP Generation Interval:     5             50              50
 SPF Interval:                2           1000            1000
 Current SPF hold interval(ms): Level 1: 1000, Level 2: 1000
 Last Level 1 SPF run 1 seconds ago
 Last Level 2 SPF run 1 seconds ago
 Authentication mode: Level 1: None, Level 2: None
 Graceful Restart: Disabled, Graceful Restart Helper: Enabled
 Area Addresses:
   49.0001
 level 1: number dis interfaces: 0, LSDB size: 1
 level 2: number dis interfaces: 0, LSDB size: 1
```

15.4.6.63 show isis ti-lfa path

The **show isis ti-lfa path** command displays the repair path with the list of all the system IDs from the P-node to the Q-node for every destination/constraint tuple. You will see that even though node protection is configured, a link protecting LFA is computed too. This is to fallback to link protecting LFAs if the node protecting LFA becomes unavailable.

```
switch#show isis ti-lfa path 1111.1111.1005
TI-LFA paths for IPv4 address family
Topo-id: Level-2
Destination     Constraint                      Path
1111.1111.1005  exclude node 1111.1111.1002     1111.1111.1003
                                                1111.1111.1004
                exclude Vlan2387                1111.1111.1002
                SRLG strict
```

```
switch#show isis ti-lfa path 10.10.10.1/32
```

```
TI-LFA paths for IPv4 address family
Topo-id: Level-1
Destination     Constraint                    Path
--------------- ----------------------------- --------------
10.10.10.1/32   exclude Vlan2387              1111.1111.1002
                                              1111.1111.1003
                exclude node 1111.1111.1004   1111.1111.1002
    SRLG strict       1111.1111.1003
```

15.4.6.64 show isis ti-lfa tunnel

The TI-LFA repair tunnels are just internal constructs that are shared by multiple LFIB routes that compute similar repair paths. The **`show isis ti-lfa tunnel`** command displays TI-LFA repair tunnels with the primary and backup via information.

```
switch#show isis ti-lfa tunnel 1
Tunnel Index 1
   via 10.0.1.2, 'Vlan2968'
      label stack 3
   backup via 10.0.0.2, 'Vlan2387'
      label stack 900004 900002
```

15.4.6.65 show tunnel fib

The **`show tunnel fib`** command that displays tunnels programmed in the tunnel FIB also includes the TI-LFA tunnels along with protected IS-IS SR tunnels.

```
switch#show tunnel fib ti-lfa 1

Type 'TI-LFA', index 1, forwarding None
   via 10.0.1.2, 'Vlan2968'
      label stack 3
   backup via 10.0.0.2, 'Vlan2387'
      label stack 900004 900002
```

```
switch#show tunnel fib isis segment-routing

Type 'IS-IS SR', index 1, endpoint 2002::b00:201/128, forwarding
 Primary
   via TI-LFA tunnel index 3 label 3
      via fe80::200:76ff:fe01:0, 'Ethernet30/1' label 900002
      backup via fe80::200:76ff:fe03:0, 'Ethernet26/1' label 132769

Type 'IS-IS SR', index 2, endpoint 2002::b00:101/128, forwarding
 Primary
   via TI-LFA tunnel index 4 label 3
      via fe80::200:76ff:fe01:0, 'Ethernet30/1' label 3
```

```
      backup via fe80::200:76ff:fe03:0, 'Ethernet26/1' label 132769
900001
```

15.4.6.66 show mpls label ranges

The **show mpls label ranges** command displays the MPLS label range available on a router is categorized into different pools which cater to different applications running on the router.

Command Mode

EXEC

Command Syntax

show mpls label ranges

Example

```
switch# show mpls label ranges
Start    End      Size     Usage
-----------------------------------------
0        15       16       reserved
16       99999    99984    static mpls
100000   116383   16384    isis (dynamic)
116384   362143   245760   free (dynamic)
362144   899999   537856   unassigned
900000   965535   65536    isis-sr
```

15.4.6.67 show mpls lfib route

The **show mpls lfib route** command displays the LFIB information for a specified route or for all routes. The source column depicts the MPLS control plane protocol that is responsible for the label binding that resulted in this LFIB route.

Command Mode

EXEC

Command Syntax

show mpls lfib route [*label_num*]

Syntax

- ***label_num*** Displays only the LFIB information for the specified route. If no label number is specified, the command displays information for all LFIB routes.

Example

- This command displays LFIB information for all routes.

```
switch# show mpls lfib route
MPLS forwarding table (Label [metric] Vias) - 7 routes
MPLS next-hop resolution allow default route: False
Via Type Codes:
        M - Mpls Via, P - Pseudowire Via,
        I - IP Lookup Via, V - Vlan Via,
        VA - EVPN Vlan Aware Via, ES - EVPN Ethernet Segment Via,
        VF - EVPN Vlan Flood Via, AF - EVPN Vlan Aware Flood Via,
        NG - Nexthop Group Via
Source Codes:
        S - Static MPLS Route, B2 - BGP L2 EVPN,
        B3 - BGP L3 VPN, R - RSVP,
```

```
        P - Pseudowire, L - LDP,
        IP - IS-IS SR Prefix Segment, IA - IS-IS SR Adjacency Segment,
        IL - IS-IS SR Segment to LDP, LI - LDP to IS-IS SR Segment,
        BL - BGP LU, ST - SR TE Policy,
        DE - Debug LFIB
 IA 100000 [1]
            via M, 1.0.1.2, pop
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930
 IA 100001 [1]
            via M, fe80::200:eff:fe02:0, pop
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930
 IP 900008 [1]
            via M, 1.0.1.2, swap 900008
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930
 IP 900009 [1]
            via M, fe80::200:eff:fe02:0, swap 900009
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930 switch#
```

- This command displays LFIB information only for the route labeled ***900008***.

```
switch# show mpls lfib route 900008
MPLS forwarding table (Label [metric] Vias) - 7 routes
MPLS next-hop resolution allow default route: False
Via Type Codes:
        M - Mpls Via, P - Pseudowire Via,
        I - IP Lookup Via, V - Vlan Via,
        VA - EVPN Vlan Aware Via, ES - EVPN Ethernet Segment Via,
        VF - EVPN Vlan Flood Via, AF - EVPN Vlan Aware Flood Via,
        NG - Nexthop Group Via
Source Codes:
        S - Static MPLS Route, B2 - BGP L2 EVPN,
        B3 - BGP L3 VPN, R - RSVP,
        P - Pseudowire, L - LDP,
        IP - IS-IS SR Prefix Segment, IA - IS-IS SR Adjacency Segment,
        IL - IS-IS SR Segment to LDP, LI - LDP to IS-IS SR Segment,
        BL - BGP LU, ST - SR TE Policy,
        DE - Debug LFIB
 IP 900008 [1]
            via M, 1.0.1.2, swap 900008
            payload autoDecide, ttlMode uniform, apply egress-acl
            interface Vlan2930 switch#
```

15.4.6.68 show mpls segment-routing bindings

The **`show mpls segment-routing bindings`** command displays the local label bindings and label bindings on the peer routers for each prefix that has a segment advertised. Peer ID here represents the IS-IS system ID of the peer.

Command Mode

EXEC

Command Syntax

`show mpls segment-routing bindings`

Example

```
switch# show mpls segment-routing bindings
1.0.7.1/32
```

```
 Local binding: Label: 900002
 Remote binding: Peer ID: 1111.1111.1001, Label: imp-null
 Remote binding: Peer ID: 1111.1111.1003, Label: 900002
1.0.8.1/32
 Local binding: Label: imp-null
 Remote binding: Peer ID: 1111.1111.1001, Label: 900004
 Remote binding: Peer ID: 1111.1111.1003, Label: 900004
1.0.9.1/32
 Local binding: Label: 900006
 Remote binding: Peer ID: 1111.1111.1001, Label: 900006
 Remote binding: Peer ID: 1111.1111.1003, Label: imp-null
```

15.4.6.69 shutdown (IS-IS)

The **shutdown** command disables IS-IS on the switch without modifying the IS-IS configuration.

The **no shutdown** and **default shutdown** commands enable the IS-IS instance by removing the **shutdown** command from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- These commands disable IS-IS on the switch.

```
switch(config)# router isis Osiris
switch(config-router-isis)# shutdown
switch(config-router-isis)#
```

- This command enables IS-IS on the switch.

```
switch(config)# router isis Osiris
switch(config-router-isis)# no shutdown
switch(config-router-isis)#
```

15.4.6.70 shutdown (IS-IS SR)

The **shutdown** and **default shutdown** commands administratively disable IS-IS SR on the switch without modifying the IS-IS SR configuration.

The **no shutdown** command enables IS-IS SR.

Command Mode

Segment-Routing MPLS Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- These commands administratively disable IS-IS SR on the switch but preserve the IS-IS SR configuration.

```
switch(config)# router isis Osiris
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)# shutdown
switch(config-router-isis-sr-mpls)#
```

- This command enables IS-IS SR on the switch.

```
switch(config)# router isis Osiris
switch(config-router-isis)# segment-routing mpls
switch(config-router-isis-sr-mpls)# no shutdown
switch(config-router-isis-sr-mpls)#
```

15.4.6.71 spf-interval

The **spf-interval** command sets the Shortest Path First (SPF) timer that defines the interval between IS-IS path calculations. The default value is two seconds.

This command also configures the maximum wait interval between any two SPF runs, initial wait interval before executing the first SPF computation, and the hold time between the first and second SPF runs.

The **no spf-interval** and **default spf-interval** commands restore the default maximum IS-IS path calculation interval to two seconds by removing the **spf-interval** command from ***running-config***.

For information about viewing SPF interval values, see **Displaying IS-IS Instance Information**.

Command Mode

Router-IS-IS Configuration

Command Syntax

spf-interval ***max-wait*** [***initial-wait*** | ***hold-time***]

no spf-interval

default spf-interval

Parameters

- ***max-wait*** Value ranges from ***1*** through ***300*** seconds. Default maximum wait interval is ***2*** seconds.
- ***initial-wait*** Value ranges from ***1*** through ***300000*** ms. Default initial wait interval is ***1000*** ms.
- ***hold-time*** Value ranges from ***1*** through ***300000*** ms. Default hold interval is ***1000*** ms.

Guidelines

EOS does not support configuring topology-specific SPF timers in multi-topology deployments and IS-IS level-specific SPF timers.

Examples

- This command configures the SPF maximum wait interval to ***50*** seconds.

```
switch(config)# router isis Osiris
switch(config-router-isis)# spf-interval 50
```

- This command configures maximum wait interval, initial wait interval, and hold time to ***20*** seconds, ***10000*** ms, and ***5000*** ms respectively.

```
switch(config)# router isis inst1
switch(config-router-isis)# spf-interval 20 10000 5000
```

- This command reverts the SPF interval configuration to its default value.

```
switch(config)# router isis Osiris
switch(config-router-isis)# no spf-interval
```

15.4.6.72 timers local-convergence-delay

The Point of Local Repair (PLR) switches to the TI-LFA backup path on link failure or BFD neighbor failure but switches back to the post-convergence path once the PLR computes SPF and updates its LFIB. This sequence of events can lead to micro-loops in the topology if the PLR converges faster than other routers along the post-convergence path. So a configuration option is provided to apply a delay, after which the LFIB route being protected by the TI-LFA loop-free repair path will be replaced by the post-convergence LFIB route.

Command Mode

IS-IS address-family sub-mode

Command Syntax

`timers local-convergence-delay` [***delay_in_seconds***] protected-prefixes

Parameters

- ***delay_in_seconds*** The convergence delay, in seconds. A default of 10 seconds is used when the command is used without an explicitly specified delay.
- **protected-prefixes** The prefix which the LFIB route being protected by the TI-LFA loop-free repair path will be replaced by the post-convergence LFIB route.

15.5 Border Gateway Protocol (BGP)

Border Gateway Protocol (BGP) exchanges routing information among neighboring routers in different Autonomous Systems (AS). Arista switches use BGP version 4+, incorporating the multiprotocol extensions defined by ***RFC 4760*** so that BGP can carry both IPv4 and IPv6 routes simultaneously over a single BGP peering.

This section contains the following topics:

- BGP Conceptual Overview
- Configuring BGP
- BGP IPv6 Link Local Peers Discovery
- BGP Examples
- BGP Commands

Arista switches support these BGP functions:

- A single BGP instance.
- Simultaneous internal (iBGP) and external (eBGP) peering.
- Multiprotocol BGP, including IPv4-mapped IPv6 address next hops for IPv6 labeled-unicast routes.
- BGP Confederations.
- BGP Selective Route Download.
- BGP Route Reflection.

15.5.1 BGP Conceptual Overview

BGP is a protocol that exchanges routing information among neighboring routers in different autonomous systems through TCP sessions.

BGP neighbors (peers) communicate through a TCP session on port ***179***. They are established by manual configuration commands (static peers) or by creating a peer group listen range and accepting incoming peering requests in that range (dynamic peers). Internal BGP (iBGP) peers operate within a single Autonomous System (AS). External BGP (eBGP) peers operate between autonomous systems. Border routers are on AS boundaries and exchange information with other autonomous systems; the primary function of border routers is distributing routes. Internal routers do not distribute route updates that they receive.

BGP defines a state machine for establishing connections. BGP routers maintain a state variable for each peer-to-peer session to track connection status. The state machine consists of these states:

- **Idle**: the router initializes BGP resources, refuses inbound BGP connection attempts, initiates a TCP connection to the peer, then transitions to the **Connect** state.
- **Connect**: the router waits for the TCP connection to complete, then sends an OPEN message to the peer and transitions to the **OpenSent** state if successful. If unsuccessful, it sets the **ConnectRetry** timer and transitions to the **Active** state upon expiry.
- **Active**: the router sets the **ConnectRetry** timer to zero and returns to the **Connect** state.
- **OpenSent:** the router waits for an OPEN message from the peer. After receiving a valid message, it transitions to the **OpenConfirm** state.
- **OpenConfirm**: the router waits for a keepalive message from its peer. If the message is received prior to a timeout expiry, the router transitions to the **Established** state. If the timeout expires or an error condition exists, the router transitions to the **Idle** state.
- **Established**: peers exchange UPDATE messages about routes they advertise. If an UPDATE message contains an error, the router sends a NOTIFICATION message and transitions to the **Idle** state.

During established BGP sessions, routers exchange UPDATE messages about the destinations to which they offer connectivity. The route description includes the destination prefix, prefix length, autonomous systems in the path, the next hop, and information that affects the acceptance policy of the receiving router. UPDATE messages also list destinations to which the router no longer offers connectivity.

BGP detects and eliminates routing loops while making routing policy decisions by using the network topology as defined by AS paths and path attributes.

15.5.1.1 Multiprotocol BGP

Multiprotocol BGP facilitates the advertisement of network routes and switch capabilities to neighbors from multiple address families over a single BGP peering. The switch supports IPv4 unicast and IPv6 unicast address families.

Neighbors negotiate to select an address family when establishing a connection. The peer session is based on this address family, which identifies the following:

- the set of network layer protocols to which the address carried in the Next Hop field must belong.
- the encoding format of the next-hop address.
- the semantics of Network Layer Reachability Information (NLRI).

15.5.1.2 BGP Confederations

BGP confederations divide an Autonomous System (AS) into subsystems (sub-ASs), each identified by a unique sub-AS number, while still appearing externally as a single AS.

15.5.1.3 QoS Control of Neighbor Discovery and ARP Packets

To help prevent BGP sessions from being affected by dropped neighbor discovery and ARP packets, some Arista switches assign those packets to a higher priority output queue when they are being software forwarded. This helps minimize hardware drops from competition with data plane packets traffic congestion.

15.5.1.4 Best-path Selection

Routing information received via the BGP protocol often contains more than one route to the same destination: the BGP best-path selection algorithm determines which of these routes will be installed in the routing table. Criteria are evaluated in order; at each step, if there is a tie for best path the next criterion is applied. If there is still a tie at the end of the process, BGP installs the route received from the peer with the lowest address. When Equal Cost Multi Path (ECMP) routing is enabled, multiple paths to a single destination may be installed in the IP routing table.

Route preferences can be shaped through configuration choices as described in Configuring Best-path Selection.

15.5.1.5 BGP Convergence

BGP supports convergence where it waits for all peers to join and receive all the routes from other peers.

Before declaring convergence, BGP also waits for IGP protocols to converge so that all IBGP sessions are established, and routes that were learned over IBGP sessions, are resolved via the IGP routes. BGP declares convergence when it has received route updates from all its peers and End-Of-RIB (EOR) markers from all the expected peers and IGP protocols have converged. Using BGP convergence, you can avoid hardware updates or route advertisement churn during a switch reload or a BGP instance start.

15.5.1.6 BGP Communities

A BGP community is a group of subnet address prefixes that share a common identifying attribute. Communities simplify routing policies by consolidating IP network spaces into logical entities that BGP speakers can address to accept, prefer, and distribute routing information. BGP communities are defined by setting the community value within route maps. Community lists then reference one or more communities as follows:

- **Standard** community lists refer to communities by name or number.
- **Expanded** community lists reference communities using regular expressions.

15.5.1.7 BGP Graceful Shutdown Community

Autonomous System Boundary Routers (ASBRs) do not update all paths received from external BGP sessions and routers. They hide inefficient alternate paths and update only best paths in the routing table. BGP route policies are applied to all internal BGP sessions of ASBRs that support the graceful shutdown procedure.

As a part of maintenance mode, these route policies perform the following functionalities on routing advertisements:

- Match the graceful shutdown community with route map rules.
- Set the local preference attribute value of the paths that are tagged with the graceful shutdown community as ***0***.

Refer to Maintenance Mode for detailed information on maintenance mode.

15.5.1.8 BGP Labeled-Unicast (LU) path Nexthop resolution over Tunnel RIB Entries

BGP Labeled-Unicast Protocol (BGP LU) path next-hop is enhanced to allow BGP in ***ribd*** mode to support resolution of BGP LU path next-hop over entries in the Tunnel RIB and fall-back to resolving over connected route when there is no entry in Tunnel RIB that provides a direct match for the BGP LU path next-hop. Previously, BGP in “ribd” mode allowed resolution of BGP Labeled-Unicast Protocol (BGP LU) path next-hop over only connected routes, resolution of the next-hop over IGP or static routes was not allowed since the next-hop router may not be in the MPLS forwarding path in which case the traffic will get dropped by the next-hop router (per IGP).

The following two use cases explain how BGP LU path next-hop resolution over tunnels would help in achieving desired or efficient traffic forwarding.

- Egress Peer Engineering (EPE)
- Inter-AS Option C

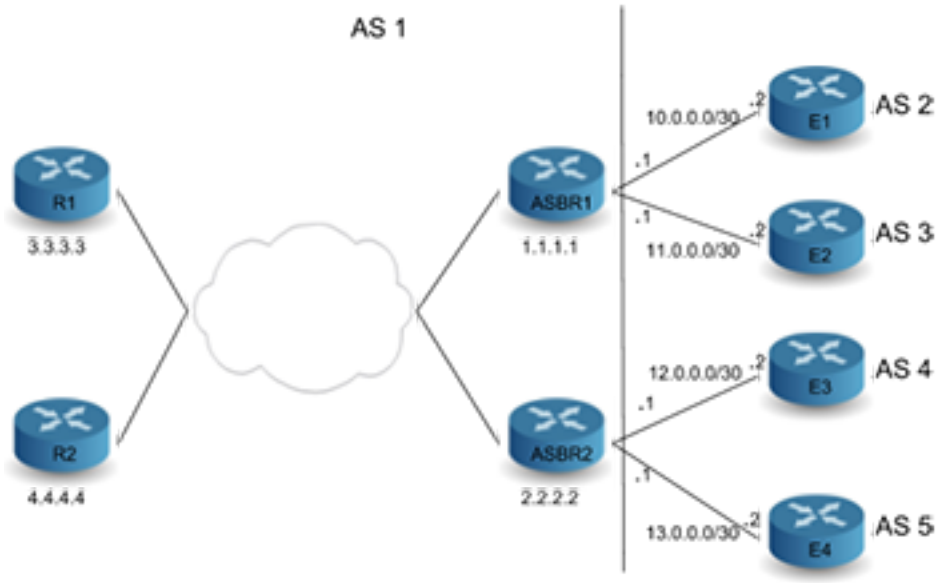
Egress Peer Engineering (EPE)

Egress Peer Engineering is a source-routing paradigm that provides ability to select an egress node/interface through which traffic goes out of an Autonomous System (AS). As shown in Figure 1 below R1, R2, ASBR1 & ASBR2 are in AS 1 and E1, E2, E3 & E4 are in different Ases. R1, R2, ASBR1 & ASBR2 could be connected each other directly or reachable to each other over an IGP (OSPF/ISIS) or MPLS tunnel. Let's assume reachability of loop-back addresses 1.1.1.1, 2.2.2.2, 3.3.3.3 & 4.4.4.4 through LDP or Segment Routing (SR). There exists an iBGP Full Mesh between R1, R2, ASBR1 & ASBR2. eBGP session is present between ASBR1 & E1, ASBR1 & E2, ASBR2 & E3 and ASBR2 & E4. Consider following BGP updates are received on ASBR1:

Prefix ***50.0.0.0/8*** next-hop ***10.0.0.2*** as-path ***2 100*** from ***E1***.

Prefix ***50.0.0.0/8*** next-hop ***11.0.0.2*** as-path ***3 200 300*** from ***E2***.

BGP path from ***E1*** will be selected as best path due to shorter AS path length. ***ASBR1*** advertises this prefix to both ***R1*** & ***R2***. Any traffic destined to prefix ***50.0.0.0/8*** from ***R1*** will always be tunneled to ***ASBR1*** and then it will always be sent on an interface connected to ***E1***. Traditional Destination based routing enforced by BGP policy and best path selection on the ASBRs may route traffic to a single AS as exit when a case can be made that for some prefixes an exit via some other AS may be preferable. BGP LU can be used here to perform traffic engineering or selecting Egress peer through which traffic should be forwarded.



A Centralized EPE Controller can be used to establish iBGP session with ***R1*** and *R2*. Let's assume Controller advertises BGP LU routes for ***E2***, i.e., ***11.0.0.2/32***, with next-hop set to loop-back IP address of ***ASBR1***, that is, ***1.1.1.1*** and a label ***111*** to ***R1*** & ***R2***.

```
switch# show ip bgp 11.0.0.2/32
BGP routing table information for VRF default
Router identifier 3.3.3.3, local AS number 1
BGP routing table entry for 11.0.0.2/32
Paths: 1 available
 Local
   1.1.1.1 labels [111] from 100.100.100.1 (100.100.100.1)
     Origin IGP, metric 0, localpref 100, IGP metric 40, weight 0, received
21:07:07 ago, valid, external, not installed
```

```
     Rx SAFI: Labels
     Tunnel RIB eligible
```

BGP LU path next-hop will get resolved over an ISIS SR tunnel present on ***R1*** and ***R2*** to reach ***1.1.1.1***, loop-back IP address of ***ASBR1***.

```
switch# show tunnel rib brief
 Endpoint   Tunnel Type  Index(es)   Metric  Metric2 Preference   Preference2
---------- ------------ ---------   ------- ------- -----------  -----------
1.1.1.1/32  IS-IS SR IPv4     5         40       0       115           0 switch#show bgp labeled-
unicast tunnel
Index  Endpoint     Nexthop/Tunnel Index Interface Labels Contributing Metric
-----  --------     ------------------- --------- ------ ------------ ------
 1     11.0.0.2/32  IS-IS SR IPv4 (5)        -     [ 111 ]    Yes        0

Metric 2 Pref Pref 2
-------- ---- ------
 100     200    0 switch#show isis segment-routing tunnel
  Index       Endpoint          Nexthop           Interface        Labels
--------      ----------        -------           ---------        ----------
    5         1.1.1.1/32        6.6.6.6           Ethernet 5       [ 900001 ]
```

Controller or CLI can be used to install a static label route on ***ASBR1*** such that ingress label ***111*** have a forwarding action of “POP and forward” to next-hop (***11.0.0.2***) in MPLS forwarding table.

```
switch# show mpls lfib route
MPLS forwarding table (Label [metric] Vias) - 20 routes
MPLS next-hop resolution allow default route: False
Via Type Codes:
         M - Mpls Via, P - Pseudowire Via,
         I - IP Lookup Via, V - Vlan Via,
         VA - EVPN Vlan Aware Via, ES - EVPN Ethernet Segment Via,
         VF - EVPN Vlan Flood Via, AF - EVPN Vlan Aware Flood Via,
         NG - Nexthop Group Via
Source Codes:
         S - Static MPLS Route, B2 - BGP L2 EVPN,
         B3 - BGP L3 VPN, R - RSVP,
         P - Pseudowire, L - LDP,
         IP - IS-IS SR Prefix Segment, IA - IS-IS SR Adjacency Segment,
         IL - IS-IS SR Segment to LDP, LI - LDP to IS-IS SR Segment,
         BL - BGP LU, ST - SR TE Policy,
         DE - Debug LFIB

S   111      [100]
               via M, 11.0.0.2, pop
                payload ipv4, apply egress-acl
                interface Ethernet 4
```

For prefixes to which traffic should be sent over interface connected ***E2*** controller will advertise a BGP route with next-hop being BGP LU prefix and higher local-preference compared to paths advertised by ***ASBR1*** and ***ASBR2***, so that path received from controller will be preferred over paths coming from ***ASBR1*** and ***ASBR2***.

```
switch# show ip bgp 50.0.0.0/8
BGP routing table information for VRF default
Router identifier 3.3.3.3, local AS number 1
BGP routing table entry for 50.0.0.0/8
Paths: 3 available
 Local
   11.0.0.2 from 100.100.100.1 (100.100.100.1)
     Origin IGP, metric 0, localpref 200, IGP metric 0, weight 0, received
 00:00:15
ago, valid, internal, best
     Rx SAFI: Unicast
 2 100
   1.1.1.1 from 1.1.1.1 (1.1.1.1)
```

```
     Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, received
 00:04:49
ago, valid, internal
     Rx SAFI: Unicast
 2 200 300
   2.2.2.2 from 2.2.2.2 (2.2.2.2)
     Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, received
 00:30:38
ago, valid, internal
     Rx SAFI: Unicast
```

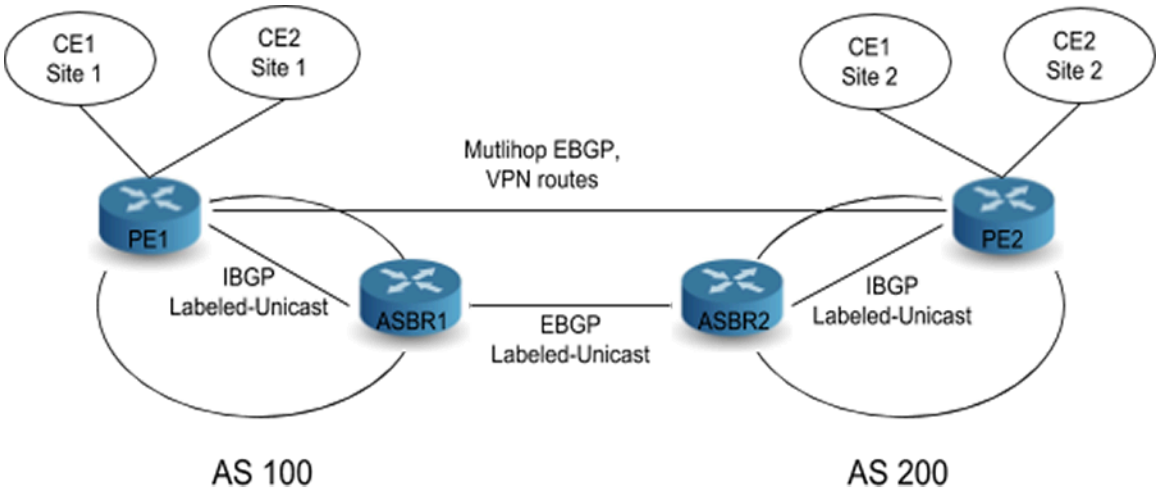
This results in pushing two labels on ***R1***, top label is the label corresponding to ISIS SR tunnel to reach ***ASBR1*** and bottom label is the label that corresponds to egress interface. Similarly LU route for ***12.0.0.0.2*** or ***13.0.0.2*** can be advertised from controller to select egress peer between ***E3*** and ***E4***. This approach provides Egress peer selection on an ingress router ***R1***/***R2***.

```
switch# show ip route 50.0.0.0/8
VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

B I    50.0.0.0/8 [200/0] via 11.0.0.2/32, BGP LU tunnel index 1
                            via 6.6.6.6, Ethernet 5, label 900001 111
```

Inter-AS Option C

Inter-AS Option C is an efficient and scalable MPLS IP VPN solution to provide connectivity between two sites of a customer connected to Provider Edge (PE) routers in different ASes. Following diagram shows a typical topology.



PE1 and ***ASBR1*** and ***PE2*** and ***ASBR2*** distribute loop-back addresses using an IBGP Labeled Unicast (LU) session. ***ASBR2*** advertises system addresses in ***AS200*** to ***ASBR1*** with next-hop as itself over EBGP LU session between them and installing Label swap entry of label sent to ***ASBR1*** (L2) to label received from ***PE2*** (L1) in MPLS forwarding table. ***ASBR1*** further propagates system addresses in ***AS200*** learned from ***ASBR2*** into ***AS100*** or to ***PE1*** using IBGP LU session with next-hop as itself and installing Label swap entry with label

advertised to ***PE1*** (L3) to Label received from ***ASBR2*** (L2) in MPLS forwarding table. Similarly ***ASBR1*** sends system addresses in ***AS100*** to ***ASBR2*** over EBGP LU session, ***ASBR2*** forwards them into ***AS200*** or to ***PE2*** using IBGP LU session with itself as next-hop and this would trigger installing appropriate label swap actions into MPLS forwarding table. These advertisements results in the creation of a label switched path from ***PE1*** to ***PE2***.

PE1 and ***PE2*** exchange VPN routes between each other using a Multi hop EBGP session with next-hop being their own loop-back/system addresses. This method eliminates the requirement of storing or sending/receiving VPN routes at ASBR routers. When PE and ASBR routers are non-adjacent, but in the same AS, then LDP or ISIS-SR can be used as a transport label signaling protocol and this would need resolving BGP LU path next-hop over LDP or ISIS-SR tunnel. An IP packet destined to an address in ***CE1*** site ***2*** is received on ***PE1*** from ***CE1*** site ***1 PE1*** would need to push ***3*** labels onto it. Bottom label corresponds to packet destination address in a particular VRF of ***CE1*** site ***2*** advertised by ***PE2*** to ***PE1*** over Multi hop EBGP session, Middle label belongs to ***PE2*** system address sent by ***ASBR1*** and top label corresponding to ***ASBR1*** system address assigned by transport label signaling protocol.

15.5.1.9 BGP Selective Route Download

BGP Selective Route Download allows the learning and advertising of BGP routes without installing them in hardware. The BGP routes are filtered before installation in hardware through the route map definition and routes that are filtered out are flagged as inactive in the Routing Information Base (RIB).

The route map used for filtering is applied only to BGP learned paths and not on locally originated routes, for example, BGP aggregate or redistributed routes. Also, because the BGP routes filtered by Selective Route Download are not active in the RIB, they are not used for recursive resolution, they are not redistributed into other protocols, and they do not contribute to BGP aggregates.

When BGP Selective Route Download is configured, the best path for peer advertisement is chosen based on the following aspects. If received BGP paths exist, then the best of them is advertised to BGP peers, else, the aggregate is preferred if configured and active. If neither BGP paths nor a BGP aggregate is available, then the RIB winner is advertised.

Note: The number of routes is limited based on the compute and memory resources available at runtime.

15.5.1.10 BGP Route Reflector

A BGP route reflector is a switch within an autonomous system that forwards route information learned from iBGP peers to other iBGP peers as an alternative to a full-mesh topology. When the switch is configured as a route reflector it can also be configured to preserve the BGP attributes of the reflected routes (next-hop, local preference, and metric) in its route advertisements regardless of outbound BGP policies.

15.5.1.11 BGP Nexthop Resolution RIBs: EVPN and IPV4/6 Labeled-Unicast Support

Adds the BGP Nexthop Resolution RIBs feature for EVPN and labeled-unicast address families.

BGP Nexthop Resolution RIBs: EVPN and IPV4/6 Labeled-Unicast Support adds support for user-configured BGP Nexthop Resolution RIB profiles for various BGP-based services such as IP unicast, L3 VPN, EVPN, etcetra. This feature allows an administrator to customize the next hop resolution semantics of BGP routes with an ordered list, or profile, of resolution RIB domains (for example, either tunnel or IP domain). This allows EOS to direct specific services over the specified RIB domains, overriding the default behavior. Further, this feature, through the use of user-defined tunnel RIBs, empowers an administrator to further select a subset of tunneling protocols for specific services.

Note: This feature is only available when running the multi-agent routing protocol model.

15.5.1.11.1 Support for Set Large Community List Limitations

Resolution of NLRI from (directly connected) eBGP Speakers

For IPv4 or IPv6 unicast NLRI received from eBGP, directly connected BGP sessions are resolved by only using connected routes, or **system-connected**, in the parlance of this feature. This feature does not change this behavior, nor will configuration of a non-default resolution profile affect this behavior.

Address Family Profile Restrictions

Certain BGP address families only support a subset of possible next-hop resolution profiles. This section documents such limitations.

Address family	Restriction
IPv4/IPv6 unicast (non 6PE)	None.
IPv6 unicast 6PE	Only supports tunnel domains*.
IPv4/IPv6 unicast (eBGP directly connected)	Only supports system-connected; Not configurable.
IPv4/IPv6 VPN	Only supports tunnel domains* and system-connected.
IPv4/IPv6 LU	Only supports tunnel domains* and system-connected.
EVPN (MPLS)	Only supports tunnel domains* and system-connected.
EVPN (VXLAN)	Only supports IP domains+.

* Tunnel domains refer to tunnel RIBs, e.g. system-colored-tunnel-rib, system-tunnel-rib, or user-defined tunnel RIBs.

+ IP domains are either of system-unicast-rib or system-connected.

15.5.1.12 BGP Logical OR of Multiple Community Lists in the Same Match Command

In the multi-agent routing protocol model, the BGP agent now supports matching community lists with a logical OR via the route map **`match community or-results`** command (same applies for extended and large communities with **`match extcommunity`** and **`match large-community`**).

Without the **`or-results`** portion of the command, the default is to compute the logical AND of all provided community lists. Before, one would need to merge existing community lists into one to do a logical OR:

Issue:

```
ip community-list COMMLIST1 permit 1:1
ip community-list COMMLIST2 permit 2:2

! No way to match "COMMLIST1" or "COMMLIST2" in a singe
! route-map sequence
match community COMMLIST1 COMMLIST2
```

Workaround:

```
ip community-list standard mergedCommunityList permit 1:1
ip community-list standard mergedCommunityList permit 2:2

match community mergedCommunityList
```

15.5.1.12.1 Limitations

This feature is available only when configuring BGP in the multi-agent routing protocol model.

15.5.1.13 BGP Flowspec

The ***EOS Release 4.21.3F*** introduces support for BGP Flowspec, as defined in ***RFC5575*** and ***RFC7674***. The typical use case is to filter or redirect DDoS traffic on edge routers.

BGP Flowspec rules are disseminated using a new BGP address family. The rules include both matching criteria used to match traffic, and actions to perform on the matching traffic. The rules are programmed into TCAM resources and applied on the ingress ports for which flowspec is enabled.

15.5.1.13.1 Release Updates

EOS Release 4.x **enhancements:**

- Added support for BGP Flowspec applied to SVI.
- BGP Flowspec releases TCAM banks as they are no longer needed to store matches. Previously, once TCAM banks allocated to BGP Flowspec, they never released.

EOS Release 4.22.0 **Enhancements:**

- Added support for redirect over MPLS or GRE Tunnels.
- Added support for traffic-rate action.

EOS Release 4.22.1 **Enhancements:**

Added support for hitless rule updates. This enhancement ensures that persistent filtering rules are always active while other filtering rules are updated (example: rules are published or withdrawn by a BGP peer).

EOS Release 4.23.1 **Enhancements:**

- Added support for best-effort rule programming. When a switch receives more filtering rules from its BGP neighbors than can fit within TCAM hardware, it programs the highest priority Flowspec rules up to the maximum TCAM available on a per-ASIC basis. The maximum TCAM available could either be the per-ASIC maximum free TCAM banks or the limit set by **`feature flow-spec bank maximum tcam`** in the hardware sub-configuration. (**LIMITATION** - the best-effort rule programming does not apply when Flowspec rules, after expansion into HW TCAM entries, occupy more than 24k 160b IPv4 or 320b IPv6 HW TCAM entries. In this case, programming fails, and no flowspec rules are programmed in hardware. This limitation is resolved in ***EOS Release 4.24.2***).
- Added support for traffic-marking action. To enable traffic-marking action, the **`feature flow-spec port (ipv4|ipv6)`** command of the active TCAM profile must include action **set-dscp**.
- Added support for packet length (Type 10) component match on IPv4 packets. To enable matching on IPv4 packet length, the **`feature flow-spec port ipv4`** command of the active TCAM profile must include keyword **field ip-length**.

EOS Release 4.23.2 **Enhancements:**

- Added support for configuring BGP Flowspec in a non-default VRF. Only a single VRF is supported.
- Added support for packet length (Type 10) component match on IPv6 packets. To enable matching on IPv6 packet length, the **`feature flow-spec port ipv6`** command of the active TCAM profilemust include the keword **field ipv6-length**.

***EOS Release 4.24.0* Enhancements:**

Added support for configuring BGP Flowspec on subinterfaces. To enable subinterface support, the TCAM profile of the flow-spec feature must include **`port qualifier size 3 bits`** (see Flowspec TCAM Profile and Flowspec Policer TCAM Profile below).

***EOS Release 4.24.1* Enhancements:**

- Added support for BGP Flowspec in the DCS-7500R3 and DCS-7280R3 series.
- Added support for configuring BGP Flowspec in multiple VRFs.

***EOS Release 4.24.2* Enhancements:**

Removed ***EOS Release 4.23.1*** limitation to best effort programming.

***EOS Release 4.25.2* Enhancements:**

- Added support for BGP Flowspec applied to SVI.
- BGP Flowspec will release TCAM banks they are no longer needed to store matches. Previously, once TCAM banks were allocated to BGP Flowspec, they would never be released.

15.5.1.13.2 Limitations

Flowspec functionality:

- The following actions are supported:
  - Drop
  - Redirect to a VRF
  - Redirect to a nexthop (https://tools.ietf.org/html/draft-simpson-idr-flowspec-redirect-02 and https://tools.ietf.org/html/draft-ietf-idr-flowspec-redirect-ip-02 are supported).
  - Redirect to an IPv6 nexthop (https://tools.ietf.org/html/draft-ietf-idr-flowspec-redirect-ip-02) is supported only in EOS-4.22.0 or later.
  - Policer (***EOS Release 4.22.0*** or later)
  - Traffic-marking (***EOS Release 4.23.1*** or later)

  To redirect to a nexthop, IP RIB must have a route to resolve the specified nexthop. When redirecting to a VRF, a default route for the VRF must be configured and traffic is sent to the nexthop for the default route in this VRF.
- Prior to version ***EOS Release 4.22.0***, to redirect to a nexthop or VRF, the resolving route cannot be via MPLS VPN or GRE tunnel, so the resolving route must have regular IP nexthop(s) for the redirect action. This limitation was removed in ***EOS Release 4.22.0***, except for the IPv6 GRE tunnels support for redirect action.
- All matching components described in ***RFC 5575*** are supported, except for the following known caveats:
  - For TCP flags, the ECE, CWR, and NS flags are not supported.
  - For fragment flags, only the ***Is a fragment (IsF)*** bit is supported only for IPv4 packets. Combining source and destination ports and the Fragment flags in the same rule is not supported.
- When flowspec policer is supported, the flowspec counter feature is disabled due to a hardware limitation.
- Beginning with ***EOS Release 4.23.2***, the flowspec address family can be configured in a non-default VRF. However, only a single VRF (default or non-default) may be used on ***EOS Release 4.24.0*** or earlier versions.
- The additional BGP NLRI type (AFI=1, SAFI=134) which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment is not supported.
- The validation procedure described in ***RFC 5575*** is not supported. Any received flowspec rules are considered valid.

BGP limitations:

- ECMP of flowspec rules are not supported. If the same rule is received from two peers and ECMP is configured, only the actions received from the ECMP head are applied.
- BGP Graceful Restart is not supported.
- Policies applied on the flowspec NLRI are not supported. This means that prefix-list matching rules in a route-map will not match against flowspec rules.
- BGP Additional Paths Send functionality is not supported.

Platform limitations:

- Flowspec rules can only be applied to traffic received on ***routed Ethernet and Port-Channel*** interfaces in the initial release. L3 subinterfaces are supported starting from ***EOS Release 4.24.0***.SVIs are supported starting from ***EOS Release 4.25.2***. L2 interfaces are not supported.
- Counters can either be reported for Flowspec or ACLs, but not both.
- With ***EOS Release 4.24.1*** and earlier versions, if the number of flow-spec rules exceed the available hardware TCAM resources, all rules are removed and a message is logged.
- When reinstalling the entire set of Flowspec rules, all existing rules are removed from the hardware upon installation.

Scaling limits:

- Similar to other TCAM features, the number of rules (BGP NLRI) that are supported in flowspec depend on the match criteria of each rule. Assuming that Flowspec is the only TCAM feature enabled on the switch, it attempts to use all of the TCAM space available (24K entries per chip) in the forwarding chip. Simple flowspec IPv4 rules will map to one entry, allowing a max of 24K rules. Simple IPv6 rules each take two entries, resulting in a max of 12K rules.
- Some types of rules expand into multiple entries in the TCAM. Port ranges are a common example. Combining source and destination port ranges in a single rule multiplies the number of entries needed to cover all combinations, which can quickly consume all of the TCAM space.
- The Flowspec and Flowspec Policer TCAM profiles support configuring the feature on up to seven VRFs starting with ***EOS Release 4.24.1***. This scale can be adjusted with the number of bits in the feature's port qualifier size at the expense of removing other TCAM key fields.
- Make-before-break policer allocation affects scaling limits.

15.5.1.14 Support for Set Large Community List

EOS adds support to use large community lists in the set large community route map set clause.

The Support for Set Large Community List feature allows a large community list to be shared between a number of route maps. Changes to the large community list then affect all route-maps which use this list. This makes applying the same policy change to different inbound and outbound communication easier.

Properties of large communities and how to create large community lists are not be covered as those are described here.

15.5.1.14.1 Configuring Support for Set Large Community List

The following commands have been added to route map configuration

`set large-community large-community-list` **LIST1** [**LIST2**][additive | delete]

`no set large-community large-community-list` **LIST1** [**LIST2**][additive | delete]

`default set large-community large-community-list` **LIST1** [**LIST2**][additive | delete]

The following command replaces the large community value of the contents of the permit sequences of the specified large community list. It is possible to specify more than one large community list to the set clause. In this example, the community values in permit sequences in the lists are concatenated and applies in the set clause.

`set large-community large-community-list` **LIST1** [**LIST2**]

no set large-community large-community-list **LIST1** [**LIST2**]

default set large-community large-community-list **LIST1** [**LIST2**]

The following command works similarly to the prior command, however, it does not replace communities already set on a route; it concatenates the community values with the values specified in the list. Duplicate communities are only shown once.

default set large-community large-community-list **LIST1** [**LIST2**][additive]

set large-community large-community-list **LIST1** [**LIST2**][additive]

no set large-community large-community-list **LIST1** [**LIST2**][additive]

In the following command, the **delete** keyword is used. The **delete** keyword specifies that any large community values in the input matching any of the large community values (or large community value regular expressions) in the specified large community lists are removed.

default set large-community large-community-list **LIST1** [**LIST2**][delete]

set large-community large-community-list **LIST1** [**LIST2**][delete]

no set large-community large-community-list **LIST1** [**LIST2**][delete]

Apply the following command to the concerned neighbour which large communities are to be sent, otherwise they are not sent.

neighbour ***X.X.X.X*** **send-community large**

15.5.1.14.2 Support for Set Large Community List Show Commands

Use the following command to show information about all of the configured route maps.

show route-map

This is an example output of the **show route-map** command.

```
switch# show route-map
route-map rm1 permit 10
  Description:
  Match clauses:
  SubRouteMap:
  Set clauses:
    set large-community large-community-list lgl1 lgl2
```

15.5.1.14.3 Support for Set Large Community List Limitations

Resolution of NLRI from (directly connected) eBGP Speakers

For IPv4 or IPv6 unicast NLRI received from eBGP, directly connected BGP sessions are resolved by only using connected routes, or **system-connected**, in the parlance of this feature. This feature does not change this behavior, nor will configuration of a non-default resolution profile affect this behavior.

Address Family Profile Restrictions

Certain BGP address families only support a subset of possible next-hop resolution profiles. This section documents such limitations.

Address family	Restriction
IPv4/IPv6 unicast (non 6PE)	None.
IPv6 unicast 6PE	Only supports tunnel domains*.
IPv4/IPv6 unicast (eBGP directly connected)	Only supports system-connected; Not configurable.
IPv4/IPv6 VPN	Only supports tunnel domains* and system-connected.
IPv4/IPv6 LU	Only supports tunnel domains* and system-connected.
EVPN (MPLS)	Only supports tunnel domains* and system-connected.
EVPN (VXLAN)	Only supports IP domains+.

* Tunnel domains refer to tunnel RIBs, e.g. system-colored-tunnel-rib, system-tunnel-rib, or user-defined tunnel RIBs.

+ IP domains are either of system-unicast-rib or system-connected.

15.5.1.15 BGP Additional Paths Send Optimization

BGP Add-Path TX, or send, allows for a BGP speaker to advertise multiple paths (instead of a single best-path) for a prefix towards a peering BGP speaker. BGP Add-Path increases path diversity in a network. It restores fast traffic and has efficient link usage through multipathing. This can also be used as a monitoring solution for eligible paths to a monitoring or receiving Add-Path speaker.

Without Add-Path, a sending speaker only sends the best-path for a prefix and a receiving speaker collects all best-path announcements from its peers. The receiving speaker uses only the peer's address to identify the path.

With Add-Path, the sending speaker can potentially send multiple paths using distinct path-id's to a peer and the receiver can use to distinguish the multiple paths coming from the same sender.

15.5.1.16 Ordered Next Hops in FEC

In symmetric network topology, for the same Equal Cost Multi-Path (ECMP) route programmed at different devices in a switch layer, the various devices can program ECMP next-hops in the Forwarding Equivalence Class (FEC) for that route in varying orders. This could result in inconsistent hashing of traffic for those destination routes at the same layer of switches in the network and could be undesired behavior for certain classes of applications. Ordered FEC is an approach to order the next hops in the FEC of a route based on a network-wide device identifier for each next-hop resulting inconsistent ordering of next hops in the FEC for a route across all switches in a layer.

A BGP router-id can be used as a unique network-wide device identifier and BGP paths received from various peers for a BGP ECMP route can have their paths and subsequently, next-hops sorted based on the corresponding peer's router-id. Ordered Next Hops in the FEC feature would use the BGP router-id to achieve a consistent ordering of next hops in the FEC for a route. This feature is available with multi-agent routing protocol models.

15.5.1.16.1 Configuring Ordered Next Hops in FEC

Use the following configuration commands to implement Ordered FEC solution for BGP routes.

- The BGP instance must be configured to order ECMP paths received for a BGP route deterministically using **`bgp bestpath tie-break router-id`** under ***router bgp*** configuration mode.

```
switch(config)# router bgp 100
```

```
switch(config-router-bgp)# address-family ipv4
switch(config-router-bgp)# bgp bestpath tie-break router-id
switch(config-router-bgp)#
```

Note: Other tie-break options available under ***router bgp*** configuration mode is not supported for Ordered Next Hops in FEC solution.

- The device must be configured to enforce ordering of next hops as determined by the protocol agents in the FEC programmed for the route using the **`rib fib fec ecmp ordered`** command under ***router general*** configuration mode.

```
switch(config)# router general
switch(config-router-general)# rib fib fec ecmp ordered
switch(config-router-general)#
```

15.5.1.16.2 Ordered FEC Show Commands

The **`show ip route fec`** command displays if the next-hops in the FEC of a route have been ordered. The output below indicate the show command output before enabling the Ordered FEC solution on the device, and after enabling it. The **`show ip bgp`** command output is also included to correlate next hop with corresponding router-id of the peer that the path was received from.

Example

```
switch# show ip bgp 1.0.16.0
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 1
BGP routing table entry for 1.0.16.0/24
 Paths: 8 available
  30
    1.0.10.2 from 1.0.10.2 (10.0.1.1)
      Origin EGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:01:53 ago, valid, external, ECMP head, ECMP, best, ECMP contributor
      Rx SAFI: Unicast
  10
    1.0.8.2 from 1.0.8.2 (10.0.4.1)
      Origin EGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:01:55 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
  20
    1.0.9.2 from 1.0.9.2 (10.0.3.1)
      Origin EGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:01:54 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
  40
    1.0.11.2 from 1.0.11.2 (10.0.8.1)
      Origin EGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:01:52 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
  50
    1.0.12.2 from 1.0.12.2 (10.0.2.1)
      Origin EGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:01:52 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
  60
    1.0.13.2 from 1.0.13.2 (10.0.5.1)
      Origin EGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:01:51 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
  70
    1.0.14.2 from 1.0.14.2 (10.0.6.1)
```

```
      Origin EGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:01:50 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
  80
    1.0.15.2 from 1.0.15.2 (10.0.7.1)
      Origin EGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:01:49 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
switch#
switch#show ip ro 1.0.16.0 fec
FEC ID 4294967334, used by 100 IPv4 prefixes and 0 IPv6 prefixes
Next hops:
    via 1.0.8.2, Ethernet8
    via 1.0.9.2, Ethernet9
    via 1.0.10.2, Vlan2317
    via 1.0.11.2, Vlan2836
    via 1.0.12.2, Vlan2043
    via 1.0.13.2, Ethernet4
    via 1.0.14.2, Vlan2000
    via 1.0.15.2, Vlan2191
switch#
switch(config)#router general
switch(config-router-general)#rib fib fec ecmp ordered
switch(config-router-general)#end
switch#show ip route 1.0.16.0 fec
FEC ID 4294967334, used by 100 IPv4 prefixes and 0 IPv6 prefixes
Next hops (ordered):
    via 1.0.10.2, Vlan2317
    via 1.0.12.2, Vlan2043
    via 1.0.9.2, Ethernet9
    via 1.0.8.2, Ethernet8
    via 1.0.13.2, Ethernet4
    via 1.0.14.2, Vlan2000
    via 1.0.15.2, Vlan2191
    via 1.0.11.2, Vlan2836
```

15.5.1.16.3 Limitations

Ordered FEC is supported only for IPv4 and IPv6 BGP ECMP routes received with directly connected next hops.

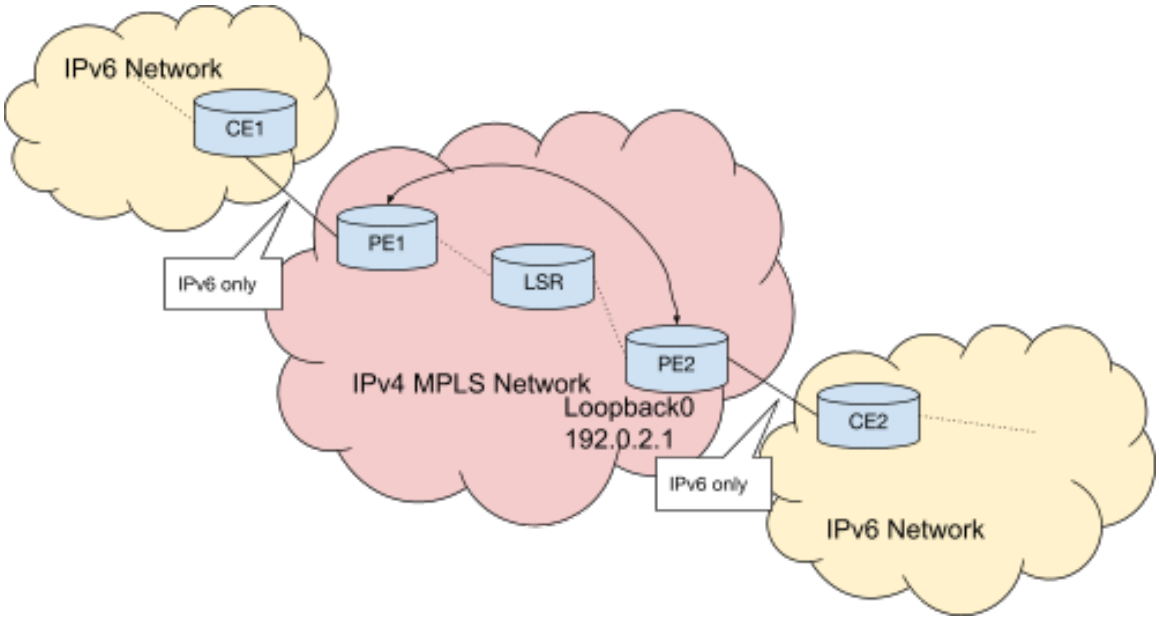
15.5.1.17 BGP IPv4-mapped IPv6 Address Next Hops for IPv6 Labeled-Unicast Routes

A BGP router in an IPv4 network may need to receive or send labeled-unicast routes to and from IPv6 networks. A receiving BGP router can be configured so that when it receives a next hop with an IPv4-mapped IPv6 address, the IPv4 address is used for resolving the next hop. Similarly, a sending BGP router can use the IPv4-mapped IPv6 address of its interface as the next hop. For example, this allows IPv6 labeled-unicast EPE bindings to be carried across an IPv4 MPLS network with a next hop corresponding to the border node's loopback IPv4-mapped IPv6 address. The iBGP peer receiving the IPv6 labeled-unicast EPE bindings resolves the IPv4-mapped IPv6 next hop over a IPv4 MPLS transport tunnel.

Example

In this example, a labeled-unicast path from a BGP router in one IPv6 network needs to cross an IPv4 MPLS network to a BGP router in another IPv6 network, as shown in the figure.

Figure 15-14: IPv4-mapped IPv6 address example



- Customer edge router CE2 advertises an IPv6 labeled-unicast route to provider edge router PE2.
- PE2 advertises to PE1 the IPv6 labeled-unicast route using the IPv4-mapped address of its loopback interface.
- PE1 receives the IPv6 labeled-unicast route, and uses the IPv4 address of the IPv4-mapped IPv6 address in order to resolve the next hop. It resolves to PE2 with an IPv4 multiprotocol label switching (MPLS) tunnel.
- PE1 advertises the IPv6 labeled-unicast route to CE1.

This allows a BGP speaker to send and receive IPv6 labeled-unicast paths with IPv4-mapped IPv6 next hops through the use of appropriate send-side policy and receive-side policy.

15.5.2 Configuring BGP

These sections describe basic BGP configuration steps:

- Configuring BGP Instances
- Configuring BGP Neighbors
- Configuring GTSM for BGP
- Configuring Routes
- Configuring Address Families
- Configuring Best-path Selection
- Configuring BGP Convergence
- Configuring BGP Graceful Shutdown Community
- Configuring BGP Additional Paths Send
- Configuring BGP Selective Route Download
- Configuring Nexthop Resolution
- Configuring BGP Confederations
- Configuring BGP Flowspec
- Configuring BGP Logical OR of Multiple Community Lists
- Setting the BGP Missing Policy Action
- Configuring BGP IPv4-mapped IPv6 Address Next Hops for IPv6 Labeled-Unicast Routes

- BGP Operational Commands

15.5.2.1 Configuring BGP Instances

15.5.2.1.1 Creating an Instance and Entering BGP Configuration Mode

The switch supports one BGP instance, which is associated with a specified Autonomous System (AS). To other BGP peers, the AS number uniquely identifies the network to which the switch belongs. Arista switches support four-byte AS numbers as described in ***RFC 4893***. Four-byte AS number capability is communicated to BGP peers in OPEN messages. When communicating with a BGP peer which does not support four-byte AS numbers, the switch will replace AS numbers greater than ***65535*** with the well-known two-byte AS number ***23456*** (also called AS_TRANS), and encode the actual four-byte AS numbers using the ***AS4_PATH*** attribute.

The switch must be in router-BGP configuration mode to run BGP configuration commands. The **router bgp** command places the switch in the ***router-BGP*** configuration mode for creating a BGP instance if one was not previously created. BGP configuration commands apply globally to the BGP instance.

Example

This command places the switch in router-BGP configuration mode. It also creates a BGP instance in AS ***50*** if an instance was not previously created.

```
switch(config)# router bgp 50
switch(config-router-bgp)#
```

When a BGP instance exists, the **router bgp** command must include its autonomous system. Any attempt to create a second instance results in an error message.

Example

This command attempts to open a BGP instance with a different AS number from that of the existing instance. The switch displays an error and stays in global configuration mode.

```
switch(config)# router bgp 100
% BGP is already running with AS number 50
switch(config)#
```

15.5.2.1.2 Configuring BGP in a VRF

IPv6 VRF support in EOS allows application of a BGP configuration to a single VRF instance, overriding global commands. To apply VRF-specific BGP configuration, use the **vrf** command within router-BGP configuration mode to enter BGP VRF configuration mode. IPv6 BGP VRF configuration is performed in the VRF submode of the router-BGP configuration mode. This submode is also where a Route Distinguisher (RD) is configured for a VRF on switches running Ethernet VPN (EVPN): use the **rd (Router-BGP VRF and VNI Configuration Modes)** command to configure an RD for a VRF.

Examples

- These commands place the switch in BGP VRF configuration mode for VRF ***purple***. Commands issued in this mode override the global BGP configuration for the specified VRF instance.

```
switch(config)# router bgp 1
switch(config-router-bgp)# vrf purple
switch(config-router-bgp)#
```

- These commands activate IPv6 address-family support for the IPv6 neighbor ***2001:0DB8:8c01::1*** in VRF ***purple***.

```
switch(config-router-bgp-vrf-purple)# router-id 1.1.1.1
switch(config-router-bgp-vrf-purple)# neighbor 2001:0DB8:8c01::1 remote-as
 16
switch(config-router-bgp-vrf-purple)# address-family ipv6
switch(config-router-bgp-vrf-purple-af)# neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-vrf-purple-af)#
```

- This command configures a route distinguisher for VRF ***purple***.

```
switch(config-router-bgp-vrf-purple)# rd 530:12
switch(config-router-bgp-vrf-purple)#
```

15.5.2.1.3 Using RCF in BGP configuration

RCF functions support in EOS allows application of a BGP configuration to filter routes and update route attributes. RCF functions can be configured for inbound and outbound updates on BGP neighbors under the IPv4 unicast, IPv6 unicast, IPv4 labeled unicast, and IPv6 labeled unicast address families.

Examples

- These commands configure the switch in RCF functions for IPv4 application.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# address-family ipv4
switch(config-router-bgp-af)# neighbor 192.168.0.1 rcf in INBOUND_POLICY()
switch(config-router-bgp-af)# neighbor 192.168.0.1 rcf out OUTBOUND_POLICY()
```

- These commands configure the switch in RCF functions for IPv6 unicast application.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# address-family ipv6 labeled-unicast
switch(config-router-bgp-af-label)# neighbor 192.168.0.1 rcf in
 LU_INBOUND_POLICY()
switch(config-router-bgp-af-label)# neighbor 192.168.0.1 rcf out
 LU_OUTBOUND_POLICY()
```

- These commands configure RCF function with the redistribute configuration statement for connected and static routes.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# redistribute connected rcf CONNECTED_POLICY()
switch(config-router-bgp)# redistribute static rcf STATIC_POLICY()
```

- These commands configure RCF function on routes redistributed into BGP from IS-IS. Level 1, level 2, or both IS-IS level routes can be specified for RCF application.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# redistribute isis level-1 rcf ISIS_LEVEL_1_
POLICY()
switch(config-router-bgp)# redistribute isis level-2 rcf ISIS_LEVEL_2_
POLICY()
switch(config-router-bgp)# redistribute isis level-1-2 rcf ISIS_LEVEL_1_
2_POLICY()
```

15.5.2.2 Configuring BGP Neighbors

15.5.2.2.1 Establishing BGP Neighbors

BGP neighbors, or peers, are established by configuration commands that initiate a TCP connection. BGP supports two types of neighbors:

- **Internal neighbors** are in the same autonomous system.
- **External neighbors** are in different autonomous systems.

BGP neighbors can be either static or dynamic:

- **Static neighbors** are established by manually configuring the connection.
- **Dynamic neighbors** are established by creating a listen range and accepting incoming connections from neighbors in that address range.

Static neighbors may belong to a static peer group, allowing them to be configured as a group. Configuration applied to an individual member of a static peer group overrides the group configuration for that peer. Dynamic neighbors must belong to a dynamic peer group, and can only be configured as a group.

Static BGP Neighbors

The **neighbor remote-as** command connects the switch with a peer, establishing a static neighbor.

Once established, a static neighbor may be added to an existing peer group. Any configuration applied to the peer group then is inherited by the neighbor, unless a conflicting configuration has been entered for that peer. Settings applied to a member of the peer group override group settings.



Note: To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (for example, in an IPv6-only environment), configure an appropriate router ID using the **router-id (BGP)** command.

Examples

- These commands establish an internal BGP connection with the peer at ***10.1.1.14***.

```
switch(config)# router bgp 50
switch(config-router-bgp)# neighbor 10.1.1.14 remote-as 50
switch(config-router-bgp)#
```

- These commands establish an external BGP connection with the peer at ***192.168.2.5***.

```
switch(config)# router bgp 50
switch(config-router-bgp)# neighbor 192.168.2.5 remote-as 100
switch(config-router-bgp)#
```

Dynamic BGP Neighbors

The **bgp listen range** command specifies a range of IPv4 addresses from which the switch will accept incoming dynamic BGP peering requests, and creates the named dynamic peer group to which those peers belong. Dynamic BGP neighbors are peers which have not been manually established, but are accepted into a dynamic peer group when the switch receives a peering request from them.

Dynamic peers cannot be configured individually, but inherit any configuration that is applied to the peer group to which they belong. Peering relationships with dynamic peers are terminated if the peer group is deleted.

Example

These commands create a peer group called “brazil” which accepts dynamic peering requests from the ***192.168.2.0/24*** subnet.

```
switch(config)# router bgp 50
switch(config-router-bgp)# bgp listen range 192.168.2.0/24 peer-group brazil
 remote-as 50
switch(config-router-bgp)#
```

Displaying Neighbor Connections

The **show ip bgp summary** and **show ip bgp neighbors** commands display neighbor connection status.

Example

This command indicates the connection state with the peer at ***192.168.2.5*** is **Estab** (established). The peer is an external neighbor because it is in AS ***100*** and the local server is in AS ***50***.

```
switch# show ip bgp summary
BGP summary information for VRF default
BGP router identifier 192.168.104.2, local AS number 50
Neighbor Status Codes: m - Under maintenance
  Neighbor     V  AS  MsgRcvd  MsgSent  InQ OutQ  Up/Down State  PfxRcd PfxAcc
  192.168.2.5  4  100     198      281    0    0 03:11:31 Estab  12     12
 switch#
```

Static BGP Peer Groups

A static BGP peer group is a collection of BGP neighbors which can be configured as a group. Once a static peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Neighbors added to the group will inherit any settings already created for the group. Static peer group members may also be configured individually, and the settings of an individual neighbor in the peer group override group settings for that neighbor.

When the **default** form of a BGP configuration command is entered for a member of a static peer group, the peer inherits that configuration from the peer group.

A static peer group is created with the **neighbor peer group (create)** command, or by using the **bgp listen range** command to accept dynamic peering requests. Once a static peer group has been created, static neighbors can be manually added to the group by using the **neighbor peer group (neighbor assignment)** command. The **no neighbor peer group (neighbor assignment)** command removes a neighbor from a static peer group.

The **no neighbor peer group (create)** command will delete a static peer group. When a peer group is deleted, the members of that group revert to their individual configurations, or to the system default for any attributes that have not been specifically configured for that peer.

Examples

- These commands create a peer group named ***akron***.

  ```
  switch(config)# router bgp 50
  switch(config-router-bgp)# neighbor akron peer group
  switch(config-router-bgp)#
  ```

- This command adds the neighbors at ***1.1.1.1*** and ***2.2.2.2*** to peer group ***akron***.

  ```
  switch(config-router-bgp)# neighbor 1.1.1.1 peer group akron
  ```

```
switch(config-router-bgp)# neighbor 2.2.2.2 peer group akron
switch(config-router-bgp)#
```

- These commands configure the members of peer group ***akron***, but cause the neighbor at ***1.1.1.1*** to use the system default value for out-delay.

```
switch(config-router-bgp)# neighbor akron remote-as 109
switch(config-router-bgp)# neighbor akron out-delay 101
switch(config-router-bgp)# neighbor akron maximum-routes 12000
switch(config-router-bgp)# no neighbor 1.1.1.1 out-delay
switch(config-router-bgp)#
```

Dynamic BGP Peer Groups

A dynamic BGP peer group is a collection of BGP neighbors in a specified address range which makes a peer request to the switch. Members of dynamic peer group are configured in groups and not as individuals. A dynamic peer group name is used as a parameter to apply the configuration across all the members in the group. Neighbors joining the group inherit any settings already created for the group.

The **`bgp listen range`** command is used to create a dynamic peer group. This command identifies the BGP peering request from a range of IP address, and names the dynamic peer group to which those peers belong to. The **`bgp listen range`** command can be configured to accept a peering request from a single AS number or to accept peer request from the range of AS numbers. To accept the request from the range of AS numbers use the **peer filter** option in the command as shown. If the peer filter referred by the **`bgp listen range`** command does not exist, or if the filter exists but has no match commands, it will accept any AS number.



Note: When a listen range command is modified, any existing dynamic neighbor that is already established will get reset.

To delete a dynamic peer group, use the **no** or **default** form of the **`bgp listen range`** command. All peering relationships with group members are terminated when the dynamic peer group is deleted.

Examples

- These commands create a dynamic peer group called ***brazil*** in a single AS, which accepts peering requests from the ***192.0.2.0/24*** subnet the single AS is ***5***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp listen range 192.0.2.0/24 peer-group brazil
 remote-as 5
switch(config-router-bgp)#
```

- These commands create a dynamic peer group called ***brazil*** in a range of ASNs, which accepts peering requests from the ***192.0.2.0/24*** subnet. The range of AS numbers is defined by peer filter option.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp listen range 192.0.2.0/24 peer-group brazil
 peer-filter group-1
switch(config-router-bgp)#
```

- The **`show ip bgp peer group`** command displays the source of a listen range's remote AS number definition as shown.

```
switch(config-router-bgp)# show ip bgp peer-group
BGP peer-group is brazil
 BGP version 4
 Listen-range subnets:
VRF default:
192.0.2.0/24, remote AS 5
192.0.2.0/24, peer filter group1
```

```
switch(config-router-bgp)#
```

15.5.2.2.2 Peer Filter

A peer filter defines a set of rules to decide whether to accept or reject the incoming peer request based on the peer's attributes. The peer filter is defined using a sequence number and a match statement, and supports one new match statement for matching against a range of BGP AS numbers. A peer filter is defined in peer filter configuration mode as shown. The peer filter command supports only matching AS ranges. Unlike route maps, peer filters do not support sets, continues or subroutines.

To delete a peer filter, use the **`no peer filter`** or **`default peer filter`** commands.

Examples

- These commands define a peer filter that accepts any AS number.

```
switch(config)# peer-filter group1
switch(config-peer-filter-group1)# 10 match as-range 1-4294967295 result
 accept
switch(config-peer-filter-group1)#
```

- These commands define a peer filter that accepts any AS number between ***65000*** and ***65100*** (inclusive) except ***65008*** and ***65009***.

```
switch(config)# peer-filter group2
switch(config-peer-filter-group2)# 10 match as-range 65008-65009 result
 reject
switch(config-peer-filter-group2)# 20 match as-range 65000-651000 result
 accept
switch(config-peer-filter-group2)#
```

- These commands define a peer filter that accepts ***3*** specific remote AS numbers.

```
switch(config)# peer-filter group3
switch(config-peer-filter-group3)# 10 match as-range 65003 result accept
switch(config-peer-filter-group3)# 20 match as-range 65007 result accept
switch(config-peer-filter-group3)# 30 match as-range 65009 result accept
switch(config-peer-filter-group3)#
```

- The **`show peer-filter`** command displays the peer filter definition.

```
switch(config)# show ip bgp peer-group3
peer-filter group3
   10 match as-range 65003 result accept
   20 match as-range 65007 result accept
30 match as-range 65009 result accept
switch(config)#
```

15.5.2.2.3 Special Considerations for IPv6

BGP predates the use of IPv6, and BGP configuration assumes IPv4 connections by default. The following additional steps are used to configure IPv6 BGP neighbors.

Note: To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (e.g., in an IPv6-only environment), configure an appropriate router ID using the **`router-id (BGP)`** command.

Activating IPv6 Neighbors

By default, the switch does not negotiate or advertise IPv6 BGP routes. In order to establish a session with an IPv6 neighbor, it must be made active in the IPv6 address family. The **ipv6-unicast** option of the **bgp default** command causes the switch to send IPv6 capability messages and all network advertisements with IPv6 prefixes to all BGP neighbors. The **neighbor activate** command issued in IPv6 address family configuration mode does the same for a single BGP neighbor.

Examples

- These commands make all BGP neighbors active in the IPv6 address family.

```
switch(config)# router bgp 11
switch(config)# address-family ipv6
switch(config-router-bgp-af)# bgp default ipv6-unicast
switch(config-router-bgp-af)# exit
switch(config-router-bgp)#
```

- These commands make the BGP neighbor at ***2001:0DB8:8c01::1*** active in the IPv6 address family.

```
switch(config)# router bgp 11
switch(config)# address-family ipv6
switch(config-router-bgp-af)# neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-af)# exit
switch(config-router-bgp)#
```

Sending IPv4 NLRIs over IPv6 Connections

The switch supports the exchange of IPv4 NLRIs with IPv6 neighbors. To enable this feature for all IPv6 neighbors, use the **ipv4-unicast transport ipv6** option of the **bgp default** command in the ***IPv4 address family*** configuration mode. To enable it for a single IPv6 neighbor, use the **neighbor activate** command for that neighbor in the ***IPv4 address family*** configuration mode.

To send IPv4 NLRIs to IPv6 neighbors, the IPv4 next-hop address must also be communicated. To explicitly configure an IPv4 next hop to send to a specific IPv6 neighbor, use the **neighbor local-v4-addr** command. In some network configurations, the switch can also be configured to automatically determine the best IPv4 next-hop address for an individual IPv6 neighbor or for all neighbors in the VRF using the **neighbor auto-local-addr** command.

Examples

- These commands permit IPv4 NLRI transport over all IPv6 connections by making the IPv4 address family active on IPv6 BGP neighbors, then configure the switch to automatically select a local IPv4 address to be sent in NLRIs to the IPv6 neighbors in a peer group called ***indianapolis***.

```
switch(config)# router bgp 11
switch(config-router-bgp)# address-family ipv4
switch(config-router-bgp-af)# bgp default ipv4-unicast transport ipv6
switch(config-router-bgp-af)# exit
switch(config-router-bgp)# neighbor indianapolis auto-local-addr
switch(config-router-bgp)#
```

- These commands permit IPv4 NLRI transport with the IPv6 neighbor at ***2001:0DB8:8c01::1*** using a local IPv4 address of ***10.7.5.11***.

```
switch(config)# router bgp 11
switch(config-router-bgp)# address-family ipv4
switch(config-router-bgp-af)# neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-af)# exit
```

```
switch(config-router-bgp)# neighbor 2001:0DB8:8c01::1 local-v4-addr
 10.7.5.11
switch(config-router-bgp)#
```

15.5.2.2.4 Maintaining Neighbor Connections

BGP neighbors maintain connections by exchanging KEEPALIVE, UPDATE, and NOTIFICATION messages. Neighbors that do not receive a message from a peer within a specified period (**hold time**) close the BGP session with that peer. Hold time is typically three times the period between scheduled KEEPALIVE messages. The default keepalive period is ***60*** seconds; default hold time is ***180*** seconds.

The **timers bgp** command configures the hold time and keepalive period. A peer retains its BGP connections indefinitely when its hold time is zero.

Example

This command sets the keepalive period to ***15*** seconds and the hold time to ***45*** seconds.

```
switch(config-router-bgp)# timers bgp 15 45
switch(config-router-bgp)#
```

The **show ip bgp neighbors** command displays the hold time.

Example

This command indicates the BGP hold time is ***45*** seconds.

```
switch# show ip bgp neighbors 10.100.100.2
BGP neighbor is 10.100.100.2, remote AS 100
BGP version 4, remote router ID 192.168.100.13, VRF default
  Negotiated BGP version 4
  Last read 00:00:05, last write 00:00:05
  Hold time is 45, keepalive interval is 15 seconds         <= hold time
  Configured hold time is 45, keepalive interval is 15 seconds
  Connect timer is inactive
  Idle-restart timer is inactive
  BGP state is Established, up for 04:44:05
  Number of transitions to established: 11
  Last state was OpenConfirm
  Last event was RecvKeepAlive
  Last sent notification:Cease/administrative reset, Last time 04:44:09
  Last rcvd notification:Cease/peer de-configured, Last time    2d02h, First
 time 7d08h, Repeats 1
  Neighbor Capabilities:
  Multiprotocol IPv4 Unicast: advertised and received and negotiated
  Four Octet ASN: advertised and received
              <-------OUTPUT OMITTED FROM EXAMPLE------->
switch#
```

15.5.2.2.5 Neighbor Route Configuration

Maximum Routes

The **neighbor maximum-routes** command determines the number of BGP routes the switch accepts from a specified neighbor. The switch disables peering with the neighbor when this number is exceeded.

Example

This command configures the switch to accept ***15,000*** routes from the peer at ***192.168.18.24***.

```
switch(config-router-bgp)# neighbor 192.168.18.24 maximum-routes 15000
switch(config-router-bgp)#
```

Route Reflection

Participating BGP routers within an AS communicate eBGP-learned routes to all of their peers; they do not re-advertise iBGP-learned routes within the AS to prevent routing loops. Although a fully meshed network topology ensures that all AS members share routing information, this topology can result in high volumes of iBGP messages when scaled. Alternatively, one or more routers can be configured as route reflectors in larger networks.

A route reflector re-advertises routes learned through iBGP to a group of BGP neighbors within the AS, replacing the function of a fully meshed topology. The **`neighbor route-reflector-client`** command configures the switch to act as a route reflector and configures the specified neighbor as a client. The **`bgp client-to-client reflection`** command enables client-to-client reflection.

Cluster IDs When using route reflectors, an AS is divided into clusters. A cluster contains at least one route reflector and a group of clients to which they re-advertise route information. A cluster may contain multiple route reflectors to provide redundancy protection. Each reflector has a cluster ID. When the cluster has a single route reflector, the cluster ID is its router ID. When a cluster has multiple route reflectors, a 4-byte cluster ID is assigned to all route reflectors in the cluster, allowing them to recognize updates from other cluster reflectors. The **`bgp cluster-id`** command configures the cluster ID in a cluster with multiple route reflectors.

Attribute Preservation Outbound BGP policies can rewrite the BGP attributes (next-hop, local preference and metric) of routes advertised by a route reflector. To configure the route reflector to preserve these attributes regardless of policy (unless those policies are included in a route map), use the **bgp route-reflector preserve-attributes** command. To include route attributes at all times (even contrary to policies included in route maps), use the always option of the command.

Client-to-client Reflection Usually the clients of a route reflector are not interconnected, and any routes learned by a client are mirrored to other clients and re-advertised within the AS by the route reflector. If the clients of a route reflector are fully meshed, routes received from a client do not need to be mirrored to other clients. In this case, client-to-client reflection should be disabled using the **`no bgp client-to-client reflection`** command.

Examples

- These commands configure the switch as a route reflector and the neighbor at ***172.72.14.5*** as one of its clients, set the cluster ID to ***172.22.30.101***, and configure the reflector to preserve all BGP attributes of re-advertised routes.

```
switch(config-router-bgp)# neighbor 172.72.14.5 route-reflector-client
switch(config-router-bgp)# bgp cluster-id 172.22.30.101
switch(config-router-bgp)# bgp route-reflector preserve-attributes
switch(config-router-bgp)#
```

- This command displays the global BGP status for the default VRF, including route reflector configuration.



Note: The value of the “Attributes of reflected routes” can be **preserved** (reflected routes maintain attributes unless they are removed by an outbound BGP policy map), **always preserved** (reflected routes maintain BGP attributes regardless of all policies), or **not preserved** (reflected routes do not maintain their BGP attrributes).

```
switch# show bgp instance
BGP instance information for VRF default
BGP Local AS: 64512, Router ID: 1.1.4.1
Total peers:              14
  Configured peers:       14
```

```
  UnConfigured peers:     0
  Disabled peers:         4
  Established peers:      9
Graceful restart helper mode enabled
Attributes of reflected routes are preserved
End of rib timer timeout: 00:05:00
BGP Convergence timer is inactive
BGP Convergence information:
    BGP has converged:   yes,   Time taken to converge: 00:05:44
    Outstanding EORs:    0,     Outstanding Keepalives: 0
Convergence timeout: 00:10:00 switch#
```

Route Preference

The primary function of external peers is to distribute routes they learn from their peers. Internal peers receive route updates without distributing them. External peers receive route updates, then distribute them to internal and external peers.

Local preference is a metric that iBGP sessions use to select an external route. Preferred routes have the highest local preference value. UPDATE packets include this metric in the **LOCAL_PREF** field.

The **neighbor export-localpref** command specifies the **LOCAL_PREF** that the switch sends to an internal peer. The command overrides previously assigned preferences and has no effect on external peers.

Example

This command configures the switch to enter ***200*** in the **LOCAL_PREF** field of UPDATE packets it sends to the peer at ***10.1.1.45***.

```
switch(config-router-bgp)# neighbor 10.1.1.45 export-localpref 200
switch(config-router-bgp)#
```

The **neighbor import-localpref** command assigns a local preference to routes received through UPDATE packets from an external peer. This command has no effect when the neighbor is an internal peer.

Examples

- This command configures the switch to assign the local preference of ***50*** for routes advertised from the peer at ***172.16.5.2***.

  ```
  switch(config-router-bgp)# neighbor 172.16.5.2 import-localpref 50
  switch(config-router-bgp)#
  ```

  The **show ip bgp** command displays the **LOCAL_PREF** value for all listed routes.
- This command indicates the route to network 10.10.20.0/24 has a local preference of 400.

```
switch# show ip bgp
BGP routing table information for VRF default
Router identifier 192.168.100.23, local AS number 64512
Route status codes: s - suppressed, * - valid, > - active, # - not
 installed, E - ECMP head, e - ECMP S - Stale, c - Contributing to ECMP, b -
 backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop
 - Link Local Nexthop

        Network          Next Hop         Metric  LocPref Weight  Path
 * >Ec 10.10.20.0/24     192.168.31.3     0       400     0       64521 i
 switch#
```

Graceful Restart

Graceful BGP restart allows a BGP speaker with separate control plane and data plane processing to continue forwarding traffic during a BGP restart. Its neighbors (receiving speakers) may retain routing information from the restarting speaker while a BGP session with it is being re-established, reducing route flapping.

Arista switches can act as helpers (receiving speakers) for graceful BGP restart with neighbors that advertise graceful restart capability.

Graceful restart helper mode is enabled by default, but can be turned off globally with the **no graceful-restart-helper** command. Per-peer configuration takes precedence over the ***global*** configuration.

Examples

- This command disables graceful restart helper mode for all BGP peers.

```
switch(config-router-bgp)# no graceful-restart-helper
switch(config-router-bgp)#
```

- This command disables graceful restart helper mode for the neighbor at ***192.168.32.5*** regardless of global configuration.

```
switch(config-router-bgp)# no neighbor 192.168.32.5 graceful-restart-helper
switch(config-router-bgp)#
```

Peers with graceful restart capability advertise a restart time value as an estimate of the time it will take them to restart a BGP session. When a BGP session with a restarting speaker goes down, the switch (receiving speaker) marks routes from that peer as stale and starts the restart timer. If the session with the peer is not re-established before the restart time runs out, the switch deletes the stale routes from that peer. If the session is re-established within that time, the stale path timer is started. If the stale paths are not updated by the restarting speaker before the stale path time runs out, they are deleted. The maximum time these stale paths are retained after the BGP session is re-established is ***300*** seconds by default, but can be configured using the **graceful-restart stalepath-time** command.

Example

This command configures BGP to discard stale paths from a restarting peer ***500*** seconds after the BGP session with that peer is re-established.

```
switch(config-router-bgp)# graceful-restart stalepath-time 500
switch(config-router-bgp)#
```

15.5.2.2.6 Filtering Routes

Filtering with Route Maps

Route maps are used in BGP to directly filter IPv4 unicast routes. The **neighbor route-map (BGP)** command applies a route map to inbound or outbound BGP routes. To display the route maps associated with a specific BGP neighbor, use the **show ip bgp neighbors** command.

The redistribution of BGP unicast routes into multicast address families allows the network to take a different path for the multicast traffic. It allows redistribution of IPv4 unicast routes into the IPv4 multicast address family and IPv6 unicast routes into the IPv6 multicast address family.

The following command configures the redistribution of IPv4 unicast routes into IPv4 multicast address family in both default and non-default VRF.

```
switch(config-router-bgp)# address-family ipv4 multicast
switch(config-router-bgp-af)# route input address-family ipv4 unicast rcf
 onePfx()
```

The following commands shows the two BGP unicast routes that are received by ***bgprtr1***.

```
bgprtr1(config-router-multicast)# show bgp ipv4 unicast
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L -
 labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI Origin Validation codes: V - valid, I - invalid, U - unknown
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
 Link Local Nexthop

          Network                Next Hop              Metric  AIGP
 LocPref Weight  Path
 * >      10.10.10.1/32          1.1.1.2               0       -          100
   0       200 i
 * >      10.10.20.1/32          1.1.1.2               0       -          100
   0       200 i
```

The following command shows BGP IPv4 multicast output, when a RCF function filters *10.10.20.1/32*.

```
bgprtr1# show bgp ipv4 multicast
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L -
 labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
 Link Local Nexthop

          Network                Next Hop              Metric  AIGP
 LocPref Weight  Path
 * >      10.10.20.1/32          1.1.1.2               -       -          -
   0       ?
```

Filtering with BGP Communities

Community values are assigned to a set of subnet prefixes through route map **`set`** commands. Route map **`match`** commands subsequently use community values to filter routes. The switch uses the following **`ip community-list`** commands to filter community routes into a BGP domain:

- **`ip community-list`** creates a community list by explicitly referencing one or more communities by name or number.
- **`ip community-list regexp`** creates a community list by referencing one or more communities by regular expression.
- **`ip extcommunity-list`** creates an extended community list to identify routes for VRFs or for Link BandWidth (LBW) by explicitly referencing extended communities by prefix and number.
- **`ip extcommunity-list regexp`** creates an extended community list to identify routes for VRFs or for Link BandWidth (LBW) by regular expression.

The BGP community attribute is a ***32*** bit value formatted as follows:

- an integer between ***0*** and ***4294967040***.
- AA:NN, where AA is ***65535*** and NN specifies the community number (***0***-***65535***) within the AS.

These four community attribute values, and the associated BGP speaker actions, are predefined:

- **no-export**: speaker does not advertise the routes beyond the BGP domain.
- **no-advertise**: speaker does not advertise the routes to any BGP peers.
- **local-as**: speaker does not advertise route to any external peers.
- **internet**: speaker advertises the route to the Internet community. By default, this includes all prefixes.

Example

- These commands assign two network subnets to a prefix list, assign a community number to the prefix list members, then utilize that community in an **`ip community-list`** command to permit the routes into the BGP domain.

  1. Compose the IP prefix list.

     ```
     switch(config)# ip prefix-list PL_1 permit 10.1.2.5/24
     switch(config)# ip prefix-list PL_1 permit 10.2.5.1/28
     switch(config)#
     ```

  2. Create a route map that matches the IP prefix list and sets the community value.

     ```
     switch(config)# route-map MAP_1 permit
     switch(config-route-map-MAP_1)# match ip address prefix-list PL_1
     switch(config-route-map-MAP_1)# set community 500
     switch(config-route-map-MAP_1)# exit
     switch(config)#
     ```

  3. Create a community list that references the community.

     ```
     switch(config)# ip community-list CL_1 permit 500
     switch(config)#
     ```

BGP extended communities identify routes for VRFs or for Link BandWidth (LBW). Extended community clauses utilize Route Target (RTt) and Site of Origin Options (SOO):

- **route targets** identify sites that may receive appropriately tagged routes.
- **site of origin** identifies the site where the router learned the route.

Filtering with AS Path Access Lists

An AS path access list is a named list of permit and deny statements which use regular expressions to filter BGP routes based on their AS path attribute. AS path access lists are created using the **`ip as-path access-list`** command, and are applied using a route map **match** clause with the name of the access list as a parameter.

Example

These commands create an AS path access list identifying routes which pass through AS ***3***, create a route map which references the access list, assign the routes it filters to community ***300***, and apply the route map to the neighbor at ***192.68.14.5*** to assign a community value of ***300*** to inbound routes received from that neighbor.

1. Create the AS path access list.

   ```
   switch(config)# ip as-path access-list as_list3 permit _3
   ```

2. Create a route map that matches the AS path access list and sets the community value.

   ```
   switch(config)# route-map MAP_3 permit
   switch(config-route-map-MAP_3)# match as-path as_list3
   switch(config-route-map-MAP_3)# set community 300
   ```

```
switch(config-route-map-MAP_3)# exit
```

3. Apply the route map to the neighbor.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.68.14.5 route-map MAP_3 in
switch(config-router-bgp)#
```

15.5.2.3 Configuring GTSM for BGP

The Generalized TTL Security Mechanism (GTSM) uses a packet's Time to Live (TTL) (IPv4) or Hop Limit (IPv6) to protect BGP peering sessions from Denial-of-Service (DoS) attacks based on forged protocol packets.

An IP packet received from a BGP peer is discarded when its current TTL value is less than (***255***-***n***) where ***n*** is the configured maximum number of hops to the peer. Use the **neighbor ttl maximum-hops** command to configure the maximum hop count.

Note: IP packets to GTSM enabled BGP peers are sent with the configured TTL value of ***255***.

15.5.2.4 Configuring Routes

15.5.2.4.1 Advertising Routes

A BGP neighbor advertises routes it can reach through UPDATE packets. The **network (BGP)** command specifies a prefix that the switch advertises as a route originating from its AS.

The configuration clears the host portion of addresses entered in **network** commands. For example, ***192.0.2.4/24*** is stored as ***192.0.2.0/24***.

Example

This command configures the switch to advertise the ***10.5.8.0/24*** network.

```
switch(config-router-bgp)# network 10.5.8.0/24
switch(config-router-bgp)#
```

By default, BGP will advertise only those routes that are active in the switch's RIB. This can contribute to dropped traffic. If a preferred route is available through another protocol (like OSPF), the BGP route will become inactive and not be advertised; if the preferred route is lost, there is no available route to the affected peers. Advertising inactive BGP routes minimizes traffic loss by providing alternative routes.

The **bgp advertise-inactive** command causes BGP to advertise inactive routes to BGP neighbors. Inactive route advertisement is configured globally, but the global setting can be overridden on a per-VRF basis.

Examples

- This command configures the switch to advertise routes learned through BGP even if they are not active on the switch.

```
switch(config-router-bgp)# bgp advertise-inactive
switch(config-router-bgp)#
```

- This command overrides inactive route advertisement for VRF ***purple***.

```
switch(config-router-bgp)# vrf purple
switch(config-router-bgp-vrf-purple)# no bgp advertise-inactive
switch(config-router-bgp-vrf-purple)#
```

15.5.2.4.1.1Advertising ISIS Routes into BGP Network

The **redistribute isis route-map isis-to-bgp** command advertises the routes learned through IS-IS routes into the BGP network. It also allows the user to selectively advertise some routes and modify route attributes before advertising using route maps.

The command is available in both address-family mode and router BGP mode, but the command is rejected if configured in both address-family mode and router mode at the same time.

While redistributing IS-IS routes into BGP, the **Level-1** or **Level-2** keyword can be used to selectively redistribute Level-1 routes or Level-2 routes into BGP. The keyword is optional, and defaults to Level-2 when not configured.

Use the **show ipv6 bgp detail** command to verify that routes are advertised with correct attributes.

Note: If the command is configured in router-af mode, it only redistributes routes with matching address family. If it is configured in router mode, it applies to all enabled address-families.

Examples

- These commands redistribute IS-IS routes into BGP in the ***address-family*** mode.

```
switch(config)# router bgp 1
switch(config-router-bgp)# address-family ipv4
switch(config-router-bgp-af)# redistribute isis level-1 route-map isis-to-
bgp-v4
switch(config-router-bgp-af)#
```

- These commands redistribute IS-IS routes into BGP in the ***router BGP*** mode.

```
switch(config)# router bgp 1
switch(config-router-bgp)# redistribute isis level-1 route-map isis-to-bgp
switch(config-router-bgp)#
```

15.5.2.4.1.2Advertising OSPF Routes into BGP Network

Routes learned through the OSPF protocol can be redistributed into the BGP domain and advertised by BGP. To redistribute OSPF routes into BGP, use the **redistribute (BGP)** command. By default, **redistribute ospf** will redistribute only internal OSPF routes into BGP; the command must be issued separately with additional parameters for each type of OSPF route that is to be redistributed.

Examples

- These commands redistribute internal OSPF routes into BGP.

```
switch(config)# router bgp 1
switch(config-router-bgp) #redistribute ospf
switch(config-router-bgp)#
```

- These commands redistribute internal, external, and NSSA external OSPF routes into BGP.

```
switch(config)# router bgp 1
switch(config-router-bgp)# redistribute ospf internal
switch(config-router-bgp)# redistribute ospf external
switch(config-router-bgp)# redistribute ospf nssa-external
switch(config-router-bgp)#
```

15.5.2.4.2 BGP Route Aggregation

Aggregation combines the characteristics of multiple routes into a single route for advertisement by the BGP speaker. Aggregation can reduce the amount of information that a BGP speaker is required to store and

transmit when advertising routes to other BGP speakers. Aggregation options affect the attributes associated with the aggregated route, the advertisement of the contributor routes that comprise the aggregate, and which contributor routes are included.

Aggregate routes are created with the **aggregate-address** command, which takes an IP subnet as an argument; any routes configured on the switch that lie within that subnet then become contributors to the aggregate. Note that on Arista switches the BGP aggregate route will become active if there are any available contributor routes on the switch, regardless of the originating protocol. This includes routes configured statically.

Note: This behaviour is observed only when the Single agent routing model (ribd) is run on the switch.

BGP speakers display aggregate routes that they create as null routes (with one exception: if all the contributors to the aggregate have the same BGP path attributes, then the BGP aggregate copies those attributes and is no longer a null route). Aggregate routes are advertised into the BGP autonomous system and redistributed automatically, and their redistribution cannot be disabled. BGP neighbors display inbound aggregate routes as normal BGP routes. Null routes are displayed with the **show ip route** command; normal BGP routes (and null aggregate routes) are displayed with the **show ip bgp** and **show ip route** commands.

Aggregation Options

The **aggregate-address** command provides the following aggregate route options:

- **AS_PATH attribute inclusion**: the **as-set** option controls the aggregate route's AS_PATH and ATOMIC_AGGREGATE attribute contents. AS_PATH identifies the autonomous systems through which UPDATE message routing information passes. ATOMIC_AGGREGATE indicates that the route is an aggregate or summary of more specific routes.

  When the command includes **as-set**, the aggregate route's AS_SET attribute contains the AS numbers of contributor routes. This can help BGP neighbors to prevent loops by rejecting aggregate routes that include their AS number in the AS_SET.

  When the command does not include **as-set**, the aggregate route's ATOMIC_AGGREGATE attribute is set and the AS_PATH attribute does not include AS numbers of contributing routes.
- **Attribute assignment**: the **attribute-map** option assigns attributes contained in set commands in a specified route map's lowest sequence with any set command to the aggregated route, overriding the automatic determination of the aggregate route's attributes by the switch.
- **Route suppression**: the **summary-only** option suppresses the advertisement of the contributor routes that comprise the aggregate.
- **Contributor filtering**: the **match-map** option uses a route map to filter out contributor routes that would otherwise be included in the aggregate.

Example

- These commands create an aggregate route (***10.16.48.0/20***) from four contributor routes (***10.16.48.0/23***, ***10.16.50.0/23***, ***10.16.52.0/23***, and ***10.16.54.0/23***). The aggregate route includes the **AS_PATH** information from the contributor routes.

```
switch(config)# router bgp 1
switch(config-router-bgp)# aggregate-address 10.16.48.0/20 as-set
switch(config-router-bgp)# exit
switch(config)#
```

- These commands create an aggregate route and use a route map to add a local-preference attribute to the route.

```
switch(config)# route-map map1 permit 10
switch(config-route-map-map1)# set local-preference 40
switch(config-route-map-map1)# exit
```

```
switch(config)# router bgp 1
switch(config-router-bgp)# aggregate-address 10.16.48.0/20 attribute-map
 map1
switch(config-router-bgp)# exit
switch(config)#
```

- These commands create an aggregate route and use a route map to allow only those contributors which match a specified prefix list to be included in the aggregate route.

```
switch(config)# route-map matchmap permit 10
switch(config-route-map-matchmap)# match ip address prefix-list agglist
switch(config-route-map-matchmap)# exit
switch(config)# router bgp 1
switch(config-router-bgp)# aggregate-address 1.1.0.0/16 match-map matchmap
switch(config-router-bgp)#
```

Identifying BGP Aggregate Contributors Match in Outbound Policy

When configured, this feature introduces the ability to match on:

1. Any BGP aggregate contributor, in the outbound route maps.
2. A specific BGP aggregate's contributor, in the outbound route maps.

The attributes that are currently supported for matching on BGP aggregate contributors are community, local-preference, prefix, next-hop,route-type.

Match Contributors to Any Aggregate

To match contributors to any BGP aggregate and set attributes (say communities) on said contributor, add an outbound policy with the clause:

```
switch(config-route-map-test)# match aggregate-role contributor
```

The **match aggregate-role contributor** clause only works with outbound policies.

Example

In this example, all the BGP contributor routes (to all aggregates) is assigned to the community ***65536:100*** as they are advertised to the neighbor ***192.0.2.1***.

```
ip community-list BLUE permit 65536:100
!
route-map OUTBOUND_POLICY permit 10
 match aggregate-role contributor
 set community community-list BLUE
!
route-map OUTBOUND_POLICY permit 20
 description "Permit the routes rejected by seq10"
!
router bgp 65536
 aggregate-address 203.0.113.0/24
 neighbor 192.0.2.1 route-map OUTBOUND_POLICY out
!
```

Match Contributors to Specific Aggregates

To match contributors which contribute only to a BGP aggregate with specific attributes (say communities) and set attributes (say communities again) on said contributor, add an outbound policy with the clause:

```
switch(config-route-map-test)# match aggregate-role contributor aggregate-att
ributes MATCH_AGG_COLOR
```

Define the **MATCH_AGG_COLOR** as below:

```
route-map MATCH_AGG_COLOR
     match community RED
```

Add an aggregate definition to explicitly set the desired attributes on the aggregate of interest:

```
route-map AGG_SET_COLOR
 set community community-list RED
!
router bgp 65536
 aggregate-address 203.0.113.0/24 attribute-map AGG_SET_COLOR
```

The route map referenced by the match **aggregate-role contributor aggregate-attributes** clause discards all set operations.

Example

- In this example the BGP contributor routes to the aggregate ***203.0.113.0/24*** (that has community ***65536:200***), to be assigned the community ***65536:100*** when they are advertised outbound to the neighbor ***192.0.2.1***.

```
ip community-list BLUE permit 65536:100
ip community-list RED permit 65536:200
!
route-map AGG_SET_COLOR
 set community community-list RED
!
route-map MATCH_AGG_COLOR
 match community RED
!
route-map OUTBOUND_POLICY permit 10
 match aggregate-role contributor aggregate-attributes MATCH_AGG_COLOR
 set community community-list BLUE
!
route-map OUTBOUND_POLICY permit 20
 description “Permit the routes rejected by seq10”
!
router bgp 65536
 aggregate-address 203.0.113.0/24 attribute-map AGG_SET_COLOR
 neighbor 192.0.2.1 route-map OUTBOUND_POLICY out
!
```

Invert-result Support

This match clause supports the invert-result modifier. When applied, invert-result inverts the result of the match clause to which it is applied.

The results for the following command would be:

```
switch(config-route-map-test)# match invert-result aggregate-role contributor
 aggregate-attributes MATCH_AGG_COLOR
```

1. match all routes that are not contributors to any aggregate.
2. match all routes that are contributors to aggregates where the aggregate doesn't match **MATCH_AGG_COLOR** (provided **MATCH_AGG_COLOR** is configured).
3. not match all routes that are contributors to aggregates where the aggregate matches **MATCH_AGG_COLOR** (provided **MATCH_AGG_COLOR** is configured).
4. not match all routes that are contributors to aggregates if **MATCH_AGG_COLOR** is not configured.

Related Command

match (route-map)

15.5.2.4.3 Customizing the BGP AS-Path Attribute

The BGP Replace AS-Path feature allows the user to customize the **AS_PATH** attribute for prefixes that are either received from a BGP neighbor or advertised to a BGP neighbor. To configure the BGP Replace AS-Path feature, use the **set as-path match** and **set as-path prepend** commands.

To replace the **AS_PATH** attribute of routes received from a BGP neighbor, configure a route map and attach the policy to the corresponding BGP neighbor statement in the inbound direction.

To replace the **AS_PATH** attribute of routes that are advertised to a neighbor, configure a route map and attach the policy to the corresponding BGP neighbor statement in the outbound direction.

The Replace AS-Path feature works in conjunction with the AS-Path Prepend feature which is also used to modify the **AS_PATH** attribute. However, if both features are configured within the same route map, then the replace AS-Path feature takes precedence over the AS-Path Prepend.



Note: The BGP Replace AS-Path feature supports both eBGP and iBGP neighbors. The locally configured AS number is always prefixed to the AS-Path of routes advertised to the eBGP neighbors. This RFC behavior is retained in Arista's implementation of the Replace AS-Path feature as well.

BGP Replace AS-Path has the following limitations:

- Replacing the AS-Path should be used cautiously since it may impact BGP loop prevention.
- A few duplicated routes may be advertised and installed on a router after the original AS-Path of those routes are replaced.To fix this issue, it is always suggested to filter out such routes by prefix with BGP Community.

Example

This command replaces the AS-Path with the **none** option.

```
switch# show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q -
 Queued
for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

          Network                Next Hop              Metric  LocPref Weight Path
* >       101.101.1.0/24         80.80.1.1             -       -       -      200 i
* >       102.102.1.0/24         80.80.1.1             -       -       -      200 i
* >       103.103.1.0/24         80.80.1.1             -       -       -      200 302 i
* >       202.202.1.0/24         80.80.1.1             -       -       -      s200 i
switch# configuration terminal
switch(config)# route-map foo permit 10
switch(config-route-map-foo)# set as-path match all replacement none
switch(config-route-map-foo)# exit
switch(config)# router bgp 200
switch(config-router-bgp)# neighbor 80.80.1.2 route-map foo out
switch(config-router-bgp)# end
switch# show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
```

```
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q -
 Queued
for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

          Network                Next Hop              Metric  LocPref Weight Path
* >       101.101.1.0/24         80.80.1.1             -       -       -      200 i
* >       102.102.1.0/24         80.80.1.1             -       -       -      200 i
* >       103.103.1.0/24         80.80.1.1             -       -       -      200 i
* >       202.202.1.0/24         80.80.1.1             -       -       -      200 i switch#
```

The AS-Path of matching prefixes are replaced with an empty or a null AS-Path. AS ***302*** is removed from prefix ***103.103.1.0/24*** as shown in the above output.This command replaces the AS-Path with the **auto** option.

```
switch(config)# route-map foo permit 10
switch(config-route-map-foo)# set as-path match all replacement auto
switch(config-route-map-foo)# end
switch# show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q -
 Queued
for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

          Network                Next Hop              Metric  LocPref Weight Path
* >       101.101.1.0/24         80.80.1.1             -       -       -      200 200 i
* >       102.102.1.0/24         80.80.1.1             -       -       -      200 200 i
* >       103.103.1.0/24         80.80.1.1             -       -       -      200 200 i
* >       202.202.1.0/24         80.80.1.1             -       -       -      200 200 i switch#
```

The AS-path of matching prefixes are replaced with the locally configured AS 200.

15.5.2.4.4 Modifying the Local AS Value

The switch can replace its local AS number with a configured value when sending OPEN messages to a specified neighbor, allowing the switch to appear as a member of a different AS to that peer. In the case of a static peer, the neighbor must also be configured to recognize the modified AS in order for peering to occur. The additional configuration is unnecessary in the case of dynamic peers.

To configure a different local AS value for the switch, use the **neighbor local-as** command. To configure the peer to expect the altered ASN from the switch, use the **neighbor remote-as** command on the peer.

Example

These commands configure the switch to replace its local ASN in OPEN messages sent to the peer at ***10.13.64.1*** with ***ASN 64500***, and configure the peer to expect that ASN in messages received from the switch.

Switch Configuration

```
switch(config)# router bgp 64497
switch(config-router-bgp)# neighbor 10.13.64.1 local-as 64500 no-prepend
switch(config-router-bgp)#
```

Peer Configuration

```
peer(config)# router bgp 64502
```

```
peer(config-router-bgp)# neighbor 10.4.3.10 remote-as 64500
peer(config-router-bgp)#
```

15.5.2.4.5 AS-path Modifications for Split ASes

By default, BGP rejects routes that contain the local Autonomous System Number (ASN). Sometimes a single autonomous system is divided geographically or otherwise with one or more provider ASs in between. In these cases, a valid route can sometimes be dropped by a customer edge router because the local ASN appears in the AS-path of route advertisements that have traveled through one or more provider networks. To ensure that these routes are not dropped, the provider edge router can be configured to replace the customer AS with its own, or the customer edge router can be configured to ignore its local AS number in received routes.

Replacing Remote ASN in Outbound Route Announcements

To replace a remote ASN with the local ASN in BGP route announcements sent to a specified router, use the **neighbor as-path remote-as replace out** command.

Example

These commands configure the switch to substitute its local ASN for the ASN of the BGP neighbor at ***192.168.2.15*** in BGP routes advertised to that neighbor.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.168.2.15 as-path remote-as replace out
switch(config-router-bgp)#
```

Ignoring Local ASN in Incoming Route Announcements

To accept BGP routes that include the local ASN in their AS-path attribute, use the **neighbor allowas-in** command.

Example

These commands configure the switch to accept routes from the BGP neighbor at ***192.168.1.30*** which contain the switch's ASN in their AS paths as many as ***3*** times.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.168.1.30 allowas-in
switch(config-router-bgp)#
```

15.5.2.5 Configuring Address Families

The switch determines the network prefixes that peering sessions advertise and the BGP neighbor addresses that receive advertisements through address family activity configuration.

An address family is a data structure that defines route advertising status to BGP neighbor addresses. Each BGP neighbor address is assigned an activity level for each address family on the switch. The switch sends capability and network prefix advertisements to neighbor addresses that are active within specified address families:

- **IPv4 address family:** switch advertises IPv4 capability and network commands with IPv4 prefixes to neighbor addresses configured as **IPv4 address family active**.
- **IPv6 address family:** switch advertises IPv6 capability and network commands with IPv6 prefixes to neighbor addresses configured as **IPv6 address family active**.

15.5.2.5.1 Neighbor Address Family Configuration

Address family activity levels for neighbor addresses are configured through **`bgp default`** and **`neighbor activate`** commands.

- The **`bgp default`** command specifies the default activity level of BGP neighbor addresses for a specified address family.
- The **`neighbor activate`** command specifies deviations from default address family activity level for a specified BGP neighbor address.

Default Neighbor Activation

The **`bgp default`** command configures the default address family activity level of all configured BGP neighbor addresses. The switch advertises the following to **address family active** addresses:

- IPv4 address family active: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family active: IPv6 capability and all network advertisements with IPv6 prefixes.

These commands configure default address family activity levels for configured BGP neighbor addresses:

- **bgp default ipv4-unicast**: all BGP neighbor addresses are IPv4 address family active (this is the switch default).
- **no bgp default ipv4-unicast**: no BGP neighbor addresses are IPv4 address family active.
- **bgp default ipv6-unicast**: all BGP neighbor addresses are IPv6 address family active.
- **no bgp default ipv6-unicast**: no BGP neighbor addresses are IPv6 address family active (this is the switch default).
- **bgp default ipv4-unicast transport ipv6**: all BGP neighbor addresses are IPv4 address family active and IPv6 neighbors can receive IPv4 NLRIs.



Note: If it is necessary to exchange IPv4 NLRIs over an IPv6 connection, the IPv4 address family must be activated on the IPv6 neighbor. To do this for all IPv6 neighbors, use the command **`bgp default ipv4-unicast transport ipv6`**. For an individual neighbor, use the **`neighbor activate`** command for the IPv6 neighbor in the IPv4 address-family configuration mode as described below.

Activating Individual Neighbor Addresses

The **`address-family`** command places the switch in address family mode to configure the address family activity level of individual BGP neighbor addresses. The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

The ***running-config*** displays **`address family`** commands in sub-blocks of the BGP configuration. The **`neighbor activate`** command is available in each address family configuration mode and defines the configuration mode ***address family*** activity level of a specified configured BGP neighbor address. Addresses are assigned one of the following states by the activate command:

- **neighbor activate** configures the address as active in the configuration mode ***address family***.
- **no neighbor activate** configures the address as not active in the configuration mode ***address family***.

The switch sends the following announcements to addresses that are active in an address family:

- **IPv4 address family**: IPv4 capability and all network routes with IPv4 prefixes.
- **IPv6 address family**: IPv6 capability and all network routes with IPv6 prefixes.

The **`neighbor route-map (BGP)`** command applies a route map to inbound or outbound BGP routes. In address-family mode, the route map is applied to routes corresponding to the configuration-mode address family. When a route map is applied to outbound routes, the switch advertises only routes matching at least one section of the route map. One outbound and one inbound route map can be applied to a neighbor

for each address family. Applying a route map to a route replaces the previous corresponding route map assignment.

Network Route Advertising in Address Families

The **network (BGP)** command specifies a network for advertisement through UPDATE packets to BGP peers. The command is available in Router-BGP and Router-BGP-Address-Family configuration modes; the mode in which the command is issued does not affect the command's execution.

- Commands with an IPv4 address are advertised to peers that are IPv4 address family-active.
- Commands with an IPv6 address are advertised to peers that are IPv6 address family-active.

Examples

- These commands instantiate BGP, configure three neighbors, and configure two network routes.

  The default activity level for IPv4 and IPv6 address families is set to the default; all neighbor addresses are IPv4 address family active and IPv6 address family not active. IPv4 capability and network routes with IPv4 prefixes are advertised to all neighbor IPv4 addresses.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)# neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)# neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)# network 172.18.23.9/24
switch(config-router-bgp)# network 2001:0DB8:de29::/64
switch(config-router-bgp)#
```

- These commands instantiate BGP on the switch, set IPv4 default activity level (**not active**), set IPv6 default activity level (**active**), and configure three neighbor addresses and two network route prefixes.

  IPv6 capability and network routes with IPv6 prefixes are advertised to all neighbor addresses.

```
switch(config)# router bgp 10
switch(config-router-bgp)# bgp default ipv6-unicast
switch(config-router-bgp)# no bgp default ipv4-unicast
switch(config-router-bgp)# neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)# neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)# neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)# network 172.18.23.9/24
switch(config-router-bgp)# network 2001:0DB8:de29::/64
switch(config-router-bgp)#
```

- These commands configure three neighbors, two network routes, and the default activity level for each address family (**not active**), and specify neighbor addresses for each address family that is active.

```
switch(config)# router bgp 11
switch(config-router-bgp)# neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)# neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)# neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)# network 172.18.23.9/24
switch(config-router-bgp)# network 2001:0DB8:de29::/64
switch(config-router-bgp)# no bgp default ipv4-unicast
switch(config-router-bgp)# no bgp default ipv6-unicast
switch(config-router-bgp)# address-family ipv4
switch(config-router-bgp-af)# neighbor 172.21.14.8 activate
switch(config-router-bgp-af)# neighbor 172.23.18.6 activate
switch(config-router-bgp-af)# exit
switch(config-router-bgp)# address-family ipv6
switch(config-router-bgp-af)# neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-af)# exit
```

```
switch(config-router-bgp)#
```

- These commands permit IPv4 NLRI transport over all IPv6 connections by making the IPv4 address family active on IPv6 BGP neighbors.

```
switch(config)# router bgp 11
switch(config)# address-family ipv4
switch(config-router-bgp-af)# bgp default ipv4-unicast transport ipv6
switch(config-router-bgp-af)# exit
switch(config-router-bgp)#
```

15.5.2.6 Configuring Best-path Selection

The best-path selection algorithm (described under **Best-Path Selection**) determines which of multiple paths to the same destination received by BGP will be added to the IP routing table. To shape route preferences and influence best-path selection, use the following commands in router-BGP configuration mode.

- **`bgp always-compare-med`** configures the switch to always consider the Multi-Exit Discriminator (MED) value when comparing paths (disabled by default).
- **`bgp bestpath as-path ignore`** configures the switch to ignore the length of the Autonomous System (AS) path when comparing routes (disabled by default).
- **`bgp bestpath as-path multipath-relax`** used in Equal-Post Multi Path (ECMP configuration) and enabled by default; the **no** form of the command configures the switch to consider paths unequal if their AS paths have different contents.
- **`bgp bestpath ecmp-fast`** the no form of this command causes the switch to ignore order of arrival in evaluating paths within an ECMP group.
- **`bgp bestpath med confed`** causes comparison of Multi-Exit Discriminator (MED) values in routes originating within the same confederation as the switch and received from confederation peers (disabled by default).
- **`bgp bestpath med missing-as-worst`** configures the switch to treat a missing MED as having the highest (least preferred) value (disabled by default). This command overrides the **missing-as-worst** setting of the **`bgp bestpath med confed`** command.
- **`bgp bestpath tie-break cluster-list-length`** configures the switch to prefer the multipath route with the shortest **CLUSTER_LIST** length in case of a tie in step 10 of the selection process (disabled by default).
- **`bgp bestpath tie-break router-id`** configures the switch to prefer the multipath route with the lowest **ROUTER_ID** in case of a tie in step 10 (disabled by default).

15.5.2.6.1 Displaying Reasons for Best-path Selection

To see the reasons why certain routes were excluded by the best-path selection process, use the **detail** option of the **`show ip bgp`** command. Enter the prefix to which BGP has selected a best path, and the output will display all learned paths. Paths which were not selected as best will display the reason they were not selected after the label **not best**.

The reason will be listed as one of the following:

- **path weight**
- **local preference**
- **AS path length**
- **origin**
- **path MED**
- **eBGP path preferred**
- **IGP cost**
- **AS path details**
- **ECMP-Fast configured**

- **router ID**
- **originator ID**
- **router ID tie-break configured**
- **cluster list length**
- **cluster list length tie-break configured**
- **peer IP address**
- **path ID**
- **redistributed route exists**
- **unknown**
- **another route from the same AS is a better BGP route**
- **peer not ready**
- **unusable**

Example

This command displays the reasons why three routes to ***172.16.0.0/24*** were rejected by the best-path algorithm. The reason for rejection is preceded by the label **Not best:**

```
switch# show ip bgp 172.16.0.0/24 detail
BGP routing table information for VRF default
Router identifier 192.168.100.18, local AS number 64524
Route status: [a.b.c.d] - Route is  queued for advertisement to peer.
BGP routing table entry for 204.1.47.220/30
 Paths: 4 available
  64512 64550 65100
    192.168.14.2 from 192.168.14.2 (192.168.100.21)
      Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago,
 valid,
external, ECMP head, ECMP, best, ECMP contributor
      Rx SAFI: Unicast
  64512 64550 65100
    192.168.24.2 from 192.168.24.2 (192.168.100.22)
      Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago,
 valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: ECMP-Fast configured
  64512 64550 65100
    192.168.34.2 from 192.168.34.2 (192.168.100.23)
      Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago,
 valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: Redistributed route exists
  64512 64550 65100
    192.168.44.2 from 192.168.44.2 (192.168.100.24)
      Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago,
 valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: eBGP path preferred
Not advertised to any peer switch#
```

15.5.2.7 Configuring BGP Convergence

To avoid hardware updates and route advertisement churn during switch reload or BGP instance start, BGP enters into the convergence state where it waits for all peers to join and receive all routes from all the peers.

BGP Convergence is bound by an upper value of convergence time (default value is ***5*** minutes) and BGP declares convergence on expiry of convergence timer. At the end of convergence, BGP updates the routes in FIB and advertises to all the peers.

To configure BGP convergence and the different timeout features, use the following commands in router-BGP configuration mode.

- **`update wait-for-convergence`** enables the BGP convergence feature.
- **`bgp convergence slow-peer time`** configures the BGP convergence idle peer timeout value. The default timeout value is 90 seconds.
- **`bgp convergence time`** configures the BGP convergence timeout value. The default timeout value is 300 seconds.

Different Cases for Convergence with Default Timeout Configuration

- **Convergence Time < 90 seconds after the first peer has joined**: this is the best case when all the configured peers have joined and EORs have been received from all peers in less than ***90*** seconds after the first peer has joined.
- **Convergence Time = 90 seconds after the first peer has joined**: this is the case when one or more BGP peers have joined within ***90*** seconds and EORs have been received from all peers within ***90*** seconds, but there are still some configured peers which have not joined yet. In this case, the convergence is declared after slow-peer timeout is reached.
- **Convergence Time > 90 seconds after the first peer has joined**: this is the case when one or more BGP peers have joined after ***90*** seconds, but EORs have not been received from all peers. As soon as EORs are received from all peers which have joined during the first ***90*** seconds, the convergence is declared.
- **Convergence Time = 300 seconds after the first peer has joined**: this is the case when EOR is not received till ***300*** seconds from some of the peers that have joined during 90 seconds after the first peer has joined.

15.5.2.7.1 Displaying BGP Convergence Status

Use the **`show bgp convergence`** command to view information about the BGP convergence status, and to know if the convergence timer has started or not. The examples below show the command output at different points in the convergence process.

No Peers Have Joined

This is the output when no peers have joined before convergence.

```
switch(config-router-bgp)# show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:02:30
Configured convergence slow peer timeout: 00:00:55
Convergence based update synchronization is enabled
Last Bgp convergence event : None
Bgp convergence state : Not Initiated (Waiting for the first peer to join)
 Convergence timer is not running
 Convergence timeout in use: 00:02:30
 Convergence slow peer timeout in use: 00:00:55
 First peer is not up yet
 All the expected peers are up: no
 All IGP protocols have converged: yes
 Outstanding EORs: 0, Outstanding Keepalives: 0
 Pending Peers: 2
 Total Peers: 2
 Established Peers: 0
 Disabled Peers: 0
 Peers that have not converged yet:
```

```
 IPv4 peers:
 201.1.1.1 (Session : Connect)
 202.1.1.1 (Session : Connect)
 IPv6 peers:
 None
switch(config-router-bgp)#
```

First Peer Has Joined

This is the output when the first peer has joined before convergence.

```
switch# show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:02:30
Configured convergence slow peer timeout: 00:00:55
Convergence based update synchronization is enabled
Last Bgp convergence event 00:00:40 ago
Bgp convergence state : Pending (Waiting for EORs/Keepalives from peer(s) and
 IGP
convergence)
 Convergence timer running, will expire in 00:01:50
 Convergence timeout in use: 00:02:30
 Convergence slow peer timeout in use: 00:00:55
 First peer came up 00:00:13 ago
 All the expected peers are up: no
 All IGP protocols have converged: yes
 Outstanding EORs: 0, Outstanding Keepalives: 0
 Pending Peers: 1
 Total Peers: 2
 Established Peers: 1
 Disabled Peers: 0
 Peers that have not converged yet:
 IPv4 peers:
 201.1.1.1 (Session : Active)
 IPv6 peers:
 None switch#
```

Convergence Timeout Reached

This is the output when the convergence timeout value is reached.

```
switch(config-router-bgp)# show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:02:30
Configured convergence slow peer timeout: 00:00:55
Convergence based update synchronization is enabled
Last Bgp convergence event 00:02:44 ago
Bgp convergence state : Timeout reached
 Time taken to converge 00:02:30
 Pending Peers: 1
 Total Peers: 2
 Established Peers: 1
 Disabled Peers: 0
 Peers that did not converge before local bgp convergence:
 IPv4 peers:
 201.1.1.1 (Session : Active)
 202.1.1.1 (Session : Established)
 IPv6 peers:
 None
switch(config-router-bgp)#
```

Converged State

This is the output during the converged state.

```
switch(config-router-bgp)# show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:05:00
Configured convergence slow peer timeout: 00:01:30
Convergence based update synchronization is enabled
Last Bgp convergence event 00:00:05 ago
Bgp convergence state : Converged
 Time taken to converge 00:00:02
 First peer came up 00:00:05 ago
 Pending Peers: 0
 Total Peers: 3
 Established Peers: 3
 Disabled Peers: 0
 Peers that did not converge before local bgp convergence:
 IPv4 peers:
 None
 IPv6 peers:
 None
switch(config-router-bgp)#
```

15.5.2.8 Configuring BGP Graceful Shutdown Community

15.5.2.8.1 Creating a Route-Map Entry That Sets the Community for Graceful Shutdown

The **set community (route-map)** command specifies community attribute modifications to BGP routes.

Example

```
switch(config)# route-map map1
switch(config-route-map-map1)# set community GSHUT
switch(config)# exit
switch(config)#
```

15.5.2.8.2 Creating a Route-Map Entry with Matching Preferences on Graceful Shutdown Community

The **ip community-list** command creates and configures a BGP access list that is based on BGP communities.

The **match (route-map)** command creates a route map clause entry that specifies one route filtering condition.

Example

```
switch(config)# ip community-list gshut_list permit GSHUT
switch(config)# route-map map1
switch(config-route-map-map1)# match community gshut_list
switch(config-route-map-map1)# exit
switch(config)#
```

15.5.2.8.3 Validating the Route-Map

The **show route-map** command displays the contents of the specified route maps.

Example

```
switch# show route-map map1
```

```
route-map map1 permit 10
Description:
Match clauses:
Set clauses:
set community GSHUT switch#
```

15.5.2.9 Configuring BGP Additional Paths Send

The **bgp additional-paths send mode/application** command is used in the BGP configuration mode to enable BGP additional paths.

The following examples show how to configure Add-Path TX at global, address family (AF) and neighbor for both default VRF and non-default VRF.

Add-Path TX at Global Level (AF and NeighborIndependent) for Default VRF

These commands configure all peers under the default VRF to be Add-Path capable at global level with different options for BGP router ***65003***.

```
switch(config)# router bgp 65003
switch(config-router-bgp)# bgp additional-paths send any

switch(config)# router bgp 65003
switch(config-router-bgp)# bgp additional-paths send limit 2

switch(config)# router bgp 65003
switch(config-router-bgp)# bgp additional-paths send ecmp

switch(config)# router bgp 65003
switch(config-router-bgp)# bgp additional-paths send ecmp limit 2

switch(config)# router bgp 65003
switch(config-router-bgp)# bgp additional-paths send backup
```

Add-Path TX at Address-Family Level (neighbor independent) for Default VRF

These configure all peers under the default VRF to be Add-Path capable when exchanging IPv4 NLRI, under **address-family ipv4** for BGP router ***65003***.

```
switch(config)#router bgp 65003
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#bgp additional-paths send any

switch(config)#router bgp 65003
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#bgp additional-paths send limit 3

switch(config)#router bgp 65003
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#bgp additional-paths send ecmp

switch(config)#router bgp 65003
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#bgp additional-paths send ecmp limit 3

switch(config)#router bgp 65003
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#bgp additional-paths send backup
```

Add-Path TX at Neighbor Level for Default VRF

These configure a specific peer under the default VRF to be Add-Path capable for BGP router ***65003*** and neighbor ***90.0.0.1***.

```
switch(config)# router bgp 65003
switch(config-router-bgp)# neighbor 90.0.0.1 additional-paths send any

switch(config)# router bgp 65003
switch(config-router-bgp)# neighbor 90.0.0.1 additional-paths send limit

switch(config)# router bgp 65003
switch(config-router-bgp)# neighbor 90.0.0.1 additional-paths send ecmp

switch(config)# router bgp 65003
switch(config-router-bgp)# neighbor 90.0.0.1 additional-paths send ecmp limit 4

switch(config)# router bgp 65003
switch(config-router-bgp)# neighbor 90.0.0.1 additional-paths send backup
```

Add-Path TX at Global Level (AF and Neighbor Independent) for Non-default VRF

These commands configure Add-Path TX at global level (AF and neighbor independent) for non-default VRF for BGP router ***65003*** and **Acme** VRF.

```
switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# bgp additional-paths send any

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# bgp additional-paths send limit 5

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# bgp additional-paths send ecmp

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# bgp additional-paths send ecmp limit 5

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# bgp additional-paths send backup
```

Add-Path TX at Address-Family Level (neighbor Independent) for Non-default VRF

These configure all peers under the non-default VRF to be Add-Path capable under **address-family ipv4** for BGP router ***65003*** and **Acme** VRF.

```
switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# address-family ipv4
switch(config-router-bgp-vrf-Acme-af)# bgp additional-paths send any

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# address-family ipv4
switch(config-router-bgp-vrf-Acme-af)# bgp additional-paths send limit 6
```

```
switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# address-family ipv4
switch(config-router-bgp-vrf-Acme-af)# bgp additional-paths send ecmp

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# address-family ipv4
switch(config-router-bgp-vrf-Acme-af)# bgp additional-paths send ecmp limit 6

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# address-family ipv4
switch(config-router-bgp-vrf-Acme-af)# bgp additional-paths send backup
```

Add-Path TX at Neighbor Level (AF Independent) for Non-default VRF

These configure a specific peer under the non-default VRF to be Add-Path capable for BGP router ***65003***, neighbor ***90.0.0.1*** and **Acme** VRF.

```
switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# neighbor 90.0.0.1 additional-paths send any

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# neighbor 90.0.0.1 additional-paths send
 limit 7

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# neighbor 90.0.0.1 additional-paths send
 ecmp

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# neighbor 90.0.0.1 additional-paths send
 ecmp limit 7

switch(config)# router bgp 65003
switch(config-router-bgp)# vrf Acme
switch(config-router-bgp-vrf-Acme)# neighbor 90.0.0.1 additional-paths send
 backup
```

15.5.2.10 Configuring BGP Selective Route Download

The **bgp route install-map** command is used in the BGP configuration mode to enable BGP Selective Route Download. BGP Selective Route Download can also be configured in an address family or VRF instance as shown in the following examples.

The following examples show how to configure a prefix list and route map, then apply BGP Selective Route Download to the map.

Examples

- These commands install BGP routes in the ***10.0.0.0/24*** and ***20.0.0.0/24*** ranges in the RIB (and thus in the hardware), but no other BGP routes.

```
switch(config)# ip prefix-list PFXL_ALLOW
switch(config-ip-pfx)# seq 1 permit 10.0.0.0/24 ge 24 le 32
```

```
switch(config-ip-pfx)# seq 2 permit 20.0.0.0/24 ge 24 le 32
switch(config-ip-pfx)# exit
switch(config-ip-pfx)#
```

- These commands configure the permit and deny rules for BGP routes.

```
switch(config)# route-map BGP_INSTALL_MAP permit 10
switch(config-route-map-BGP_INSTALL_MAP)# match ip address prefix-list
 PFXL_ALLOW
switch(config-route-map-BGP_INSTALL_MAP)# exit
switch(config)# route-map BGP_INSTALL_MAP deny 20
switch(config)#
```

- These commands configure Selective Route Download for the map **BGP_INSTALL_MAP**.

```
switch(config)# router bgp 100
switch(config-router-bgp)# bgp route install-map BGP_INSTALL_MAP
switch(config-router-bgp)#
```

The following examples show how to configure prefix lists individually for the IPv4 and IPv6 address families, then apply BGP Selective Route Download for these address families.

Examples

- These commands configure the IPv4 address family prefix list.

```
switch(config)# ip prefix-list V4_ALLOW
switch(config-ip-pfx)# route-map BGP_V4_MAP permit 10
switch(config-route-map-BGP_V4_MAP)# match ip address prefix-list V4_ALLOW
switch(config-route-map-BGP_V4_MAP)# route-map BGP_V4_MAP deny 20
switch(config-route-map-BGP_V4_MAP)# exit
switch(config-route-map-BGP_V4_MAP)#
```

- These commands configure the IPv6 address family prefix list.

```
switch(config)# ipv6 prefix-list V6_ALLOW
switch(config-ipv6-pfx)# route-map BGP_V6_MAP permit 10
switch(config-route-map-BGP_V6_MAP)# match ipv6 address prefix-list V6_ALLOW
switch(config-route-map-BGP_V6_MAP)# route-map BGP_V6_MAP deny 20
switch(config-route-map-BGP_V6_MAP)# exit
switch(config-route-map-BGP_V6_MAP)#
```

- These commands configure Selective Route Download individually for the two address families.

```
switch(config)# router bgp 200
switch(config-router-bgp)# address-family ipv4
switch(config-router-bgp-af)# bgp route install-map BGP_V4_MAP
switch(config-router-bgp-af)# exit
switch(config-router-bgp)# address-family ipv6
switch(config-router-bgp-af)# bgp route install-map BGP_V6_MAP
switch(config-router-bgp-af)#
```

15.5.2.10.1 Displaying BGP Selective Route Download Information

The **show ip bgp** command displays BGP RIB winning paths that are not installed in the RIB.

Example

The following command displays BGP routing table information for VRF default, showing winning paths that are not installed in the RIB.

```
switch# show ip bgp
```

```
BGP routing table information for VRF default
Router identifier 1.0.0.2, local AS number 100
Route status codes: s - suppressed, * - valid, > - active, # - not installed,
 E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

       Network             Next Hop         Metric  LocPref Weight Path
 * >   6.0.0.0/24          1.0.0.1          0       100     0       ?
 * #   7.0.0.0/24          1.0.0.1          0       100     0       ?
switch#
```

The **show ip bgp** command with a specified prefix displays detailed information and the reason for the BGP RIB winning paths to that prefix not being installed in the RIB.

Example

The following command displays detailed information for the BGP routing table for VRF default.

```
switch# show ip bgp 7.0.0.0/24
BGP routing table information for VRF default
Router identifier 1.0.0.2, local AS number 100
BGP routing table entry for 7.0.0.0/24
 Paths: 1 available
  Local
    1.0.0.1 from 1.0.0.1 (1.0.0.1)
      Origin INCOMPLETE, metric 0, localpref 100, weight 0, valid,
 internal, not
installed (denied by install-map)
switch#
```

The **show ip bgp installed** command displays the list of installed routes in the BGP RIB.

Example

The following command displays the list of installed routes in BGP routing table for VRF defaults.

```
switch# show ip bgp installed
BGP routing table information for VRF default
Router identifier 1.0.0.2, local AS number 100
Route status codes: s - suppressed, * - valid, > - active, # - not installed,
 E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

        Network             Next Hop         Metric  LocPref Weight Path
 * >    6.0.0.0/24          1.0.0.1          0       100     0      ?
switch#
```

The **show ip bgp not-installed** displays the list of non-installed routes in the RIB.

Example

The following command displays the list of non-installed routes in the BGP routing table for VRF default.

```
switch# show ip bgp not-installed
BGP routing table information for VRF default
Router identifier 1.0.0.2, local AS number 100
```

```
Route status codes: s - suppressed, * - valid, > - active, # - not installed,
 E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

        Network             Next Hop         Metric  LocPref Weight Path
 * #    7.0.0.0/24          1.0.0.1          0       100     0      ?
switch#
```

15.5.2.11 Configuring Nexthop Resolution

The configuration model for this feature involves configuring and applying **Nexthop Resolution RIB Profiles** on a per-address family basis. There are two ways a profile can be applied: (1) across an entire address-family, or (2) a granular, route-map based mechanism for specific routes within an address family. The per-address-family configuration is the simplest. It enables specification of a unique profile for all the routes in a given address family, such as IPV4 unicast, or EVPN. In contrast, the route-map approach leverages the matching criteria of route-map statements to apply profiles to individual routes within an address family.

Note: Note the support for each configuration and submode were released in a phased manner.

The general configuration model for the CLI is a new command under the BGP address-family submode:

```
switch(config-router-bgp-af)# next-hop resolution ribs(PROFILE|[route-map
 NAME])
```

The PROFILE option is a list of up to three (3) resolution domains. The NAME option is the name of a route-map. Notice the PROFILE and route-map NAME options are mutually exclusive. That is, a resolution profile can be specified either explicitly at the address family level, or on a per-route basis via a route-map.

To enable setting a profile using a route-map, this feature adds support for a new set statement in the route-map submode:

```
switch(config-route-map-NAME)# set next-hop resolution ribs PROFILE
```

You ban combine this statement with existing match statements to select profiles based on the BGP path attributes of a route, or other properties.

As mentioned, the profile itself is a list of resolution domains:

```
PROFILE:=DOMAIN[DOMAIN[DOMAIN]]
```

Example configuration for EVPN MPLS, EVPN VXLAN and BGP Labeled-unicast:

```
switch(config)# router bgp id
switch(config-router-bgp)# address-family evpn
switch(config-router-bgp-af)# next-hop mpls resolution ribs PRIMARY-RIB
 [FALLBACK-RIB]
switch(config-router-bgp-af)# next-hop vxlan resolution ribs IP-RIB
        ...
    address-family ipv4 labeled-unicast
      next-hop mpls resolution ribs PRIMARY-RIB [FALLBACK-RIB]
    address-family ipv6 labeled-unicast
      next-hop mpls resolution ribs PRIMARY-RIB [FALLBACK-RIB]
```

The **PRIMARY-RIB** and **FALLBACK-RIB** refers to either tunnel domain or IP RIB domain. EVPN VXLAN only supports IP-RIB domain.

Configure the tunnel domain as:

- **tunnel-rib <tunnel-rib-name>** where the ***<tunnel-rib-name>*** refers to either the system-tunnel-rib or the user defined tunnel rib.

The IP domain can be either:

- **system-unicast-rib**
- **system-connected**

The **system-unicast-rib** refers to complete IP RIB and the **system-connected** refers to just the connected routes.

Primary and secondary RIBs cannot come from the same domain (for example, both cannot be from the tunnel domain and both cannot be from the IP RIB domain). The **FALLBACK-RIB** is optional.

Nexthops will first attempt to resolve, using the primary rib. If the resolution fails, it attempts to resolve using the fallback rib (if that exists).

Example

```
router bgp <id>
   address-family ipv4 labeled-unicast
      next-hop resolution ribs tunnel-rib USER_TR system-unicast-rib
```

All the nexthops of the IPV4 labeled-unicast routes will first attempt to resolve, using the **tunnel rib USER_TR**. If the resolution fails, the nexhops attempt to resolve using the **complete unicast IP RIB**.

Each DOMAIN can be either a system or user-defined tunnel RIB or a unicast RIB. The available resolution domains, and their corresponding tokens, are tabulated below:

Table 71: Domains

Domain	Token	Description
IP RIB	`system-unicast-rib`	The complete IP unicast RIB is available for next-hop resolution.
Connected routes (IP)	`system-connected`	Only connected routes are available for next-hop resolution.
System tunnel RIB	`tunnel-rib system-tunnel-rib`	All winning tunnels from all protocols are available for next-hop resolution.
System colored tunnel RIB	`tunnel-rib colored system-colored-tunnel-rib`	All winning, colored tunnels from all protocols are available for next-hop resolution. Only routes with an associated color can be resolved by the system colored tunnel RIB.
User-defined tunnel RIB	`tunnel-rib NAME`	All contributing tunnels to the tunnel RIB called NAME are available for next-hop resolution.
IP RIB of VPN Import VRF	`vrf-unicast-rib`	This token is limited to BGP L3VPNs.

Now the profile configuration is available under select BGP address family submodes. For example:

```
switch(config)# router bgp num
switch(config-router-bgp)# address-family evpn
```

```
switch(config-router-bgp-af)# next-hop mpls resolution ribs PROFILE
      next-hop vxlan resolution ribs PROFILE
   address-family ipv4
      next-hop resolution ribs ( PROFILE | route-map NAME )
   address-family ipv4 labeled-unicast
      next-hop resolution ribs PROFILE
   address-family ipv6
      next-hop resolution ribs ( PROFILE | route-map NAME )
      next-hop 6pe resolution ribs PROFILE
   address-family ipv6 labeled-unicast
      next-hop resolution ribs PROFILE
   address-family vpn-ipv4
      next-hop resolution ribs PROFILE
   address-family vpn-ipv6
      next-hop resolution ribs PROFILE
```

Note that a given address-family may restrict the possible profiles which can be configured, and may not support specifying a route-map. For example, the resolution profile for 6PE routes, configured via *next-hop 6pe resolution ribs PROFILE*, is constrained to only the tunnel domain. That is, the profile cannot specify either *system-unicast-rib or system-connected*. This is, of course, because it is meaningless to resolve a 6PE next-hop using either of those resolution domains.

15.5.2.11.1 Release Matrices

The following tables detail the release in which the possible configurations for this feature are available.

Configuration	Release					
	next-hop resolution ribs PROFILE command					
	4.22.0F	4.22.1F	4.23.1F	4.24.1F	4.25.1F	Unsupported / Not Applicable
IPv4/IPv6 unicast (non 6PE)	X					
IPv6 unicast 6PE	X					
IPv4/IPv6 VPN (vrf-unicast-rib)	X					
IPv4/IPv6 VPN (full profile)		X				
EVPN (MPLS)			X			
EVPN (VXLAN)			X			
IPv4/IPv6 LU			X			
IPv4/IPv6 Multicast						X
IPv4/IPv6 SR TE						X
Flowspec						X
Path Selection						X
Link State						X
RT Membership						X
	PROFILE configuration					
Up to 2 resolution domains	X					
Up to 3 resolution domains				X		
system-colored-tunnel-rib				X		
	next-hop resolution ribs route-map NAME command					
IPv4/IPv6 unicast (non 6PE)					X	
IPv6 unicast 6PE						X
IPv4/IPv6 VPN						X
EVPN (MPLS)						X
EVPN (VXLAN)						X
IPv4/IPv6 LU						X
IPv4/IPv6 Multicast						X
IPv4/IPv6 SR TE						X
Flowspec						X
Path Selection						X
Link State						X
RT Membership						X
	Route-map submode					
match ip[v6] next-hop					X	

match ip[v6] address prefix-list						X
match community						X
match extcommunity						X
match large-community						X
All other match statements						X
All other set statements						X
sub-route-map						X

15.5.2.11.2 Default Resolution Profiles

Given the aforementioned configuration model, the default resolution profiles in EOS for each address-family can be expressed by the following:

Address-family	Default profile
IPv4/IPv6 unicast (non 6PE)	tunnel-rib colored system-colored-tunnel-rib tunnel-ribsystem-tunnel-rib system-unicast-rib
IPv6 unicast 6PE	tunnel-rib colored system-colored-tunnel-rib tunnel-ribsystem-tunnel-rib
IPv4/IPv6 unicast (eBGP directly connected)	system-connected
IPv4/IPv6 VPN	tunnel-rib colored system-colored-tunnel-rib tunnel-ribsystem-tunnel-rib system-connected
IPv4/IPv6 LU	tunnel-rib colored system-colored-tunnel-rib tunnel-ribsystem-tunnel-rib system-connected
EVPN (MPLS)	tunnel-rib colored system-colored-tunnel-rib tunnel-ribsystem-tunnel-rib system-connected
EVPN (VXLAN)	system-unicast-rib
IPv4/IPv6 Multicast	*This is not supported. Multicast next-hops are first resolved in the MRIB. Failure to resolve in the MRIB results in a lookup in the unicast RIB.*
Flowspec	*These next hops are not resolved.*

Note: Support for the **system-colored-tunnel-rib** was released in ***EOS Release 4.24.1F***. For earlier releases, the defaults can be determined by omitting this resolution domain.

15.5.2.11.3 Semantics

When processing the next-hop of a route, the next-hop resolver attempts resolution by using the first domain in the route's resolution profile. If the resolution domain successfully resolves the next-hop, the resolver stops. If resolution fails, however, the resolver moves onto the next domain, if it exists, and tries again. This iterative process continues until the next-hop is either resolved, or the profile is exhausted. In the latter case, the next-hop is left unresolved.

To illustrate this, consider the following example. This resolution profile constrains the resolution of a BGP route to only the IP unicast RIB:

```
switch(config-router-bgp-af)# next-hop resolution ribs system-unicast-rib
```

When a next-hop is unresolvable in the IP unicast RIB, and there are no further resolution domains to try, then the next-hop is ultimately unresolved. In contrast, the following profile first attempts resolution in the system colored tunnel RIB, then the system tunnel RIB, and finally attempts resolution using connected routes:

```
switch(config-router-bgp-af)# next-hop resolution ribs tunnel-rib colored
 system-colored-tunnel-rib tunnel-rib system-tunnel-rib system-connected
```

Therefore, only when a next-hop cannot be resolved by any of those domains will it be ultimately unresolved.

15.5.2.11.3.Route Map Semantics

This section describes semantics and limitations specific to the **next-hop resolution ribs route-map NAME** command.

The use of a route-map to select a custom resolution profile allows for per-route granularity rather than an entire BGP address-family. The next-hop resolution semantics of a next-hop whose profile is set using a route-map are the same as the per-address family configuration. However, unlike in the per-address family configuration model, a route-map makes it possible to leave the resolution profile for a next-hop unspecified. **A next-hop for which the resolution profile is unspecified is left unresolved**. The following example illustrates this as well as the recommended configuration.

The profile below constrains the resolution of a subset of IPV4 unicast routes to only the system tunnel RIB:

```
ip prefix-list SUBSET 192.0.2.1/32 192.0.2.2/32 192.0.2.3/32
route-map TUNNEL_ONLY permit 10
   match ip next-hop prefix-list SUBSET
   set next-hop resolution ribs tunnel-rib system-tunnel-rib

router bgp 64512
   address-family ipv4
      next-hop resolution ribs route-map TUNNEL_ONLY
```

Note, however, that the **TUNNEL_ONLY** route-map applies to *all IPV4 unicast routes*. Further, note that only routes whose next-hop value matches SUBSET will have a resolution profile set. **All other IPV4 unicast routes will have no resolution profile.** Any route without a resolution profile is left **unresolved**. This is often not intentional.

A more common use case is to allow the route's which do not match a given sequence to fallback to the system default resolution behavior. This can be achieved by adding a second sequence to the route-map with no match statements (matches all routes), and a single set statement which sets the default profile (see the **Default Resolution Profiles** section) for the given address family.

Example

```
ip prefix-list SUBSET 192.0.2.1/32 192.0.2.2/32 192.0.2.3/32
route-map TUNNEL_ONLY permit 10
   match ip next-hop prefix-list SUBSET
   set next-hop resolution ribs tunnel-rib system-tunnel-rib
route-map TUNNEL_ONLY permit 20
   set next-hop resolution ribs tunnel-rib colored system-colored-tunnel-rib
 tunnel-rib system-tunnel-rib system-unicast-rib

router bgp 64512
   address-family ipv4
      next-hop resolution ribs route-map TUNNEL_ONLY
```

This feature provides an explicit token to automatically fallback to the default resolution profile of whichever address family the route-map is applied:

```
ip prefix-list SUBSET 192.0.2.1/32 192.0.2.2/32 192.0.2.3/32
```

```
route-map TUNNEL_ONLY permit 10
   match ip next-hop prefix-list SUBSET
   set next-hop resolution ribs tunnel-rib system-tunnel-rib
route-map TUNNEL_ONLY permit 20
   set next-hop resolution ribs system-default

router bgp 64512
   address-family ipv4
      next-hop resolution ribs route-map TUNNEL_ONLY
```

15.5.2.11.4 BGP L3VPNs: Next-hop Resolution ribs vrf-unicast-rib

This subfeature affects both the profile used to resolve BGP VPN routes as well as the VRF in which the route resolution takes place. With this feature disabled, or prior to ***EOS Release 4.22.0F***, imported VPN routes and is subject to the following restriction:

For each VPN route received from a neighbor, the route is imported (based on route-targets) and installed into the target VRF (import-vrf), **only if the nexthop of the route is resolvable via an MPLS tunnel in the default VRF.**

With this feature enabled, the above restriction is lifted, enabling a VPN route to be imported into the target VRF unconditionally. The plain IP unicast route is subsequently resolved using the unicast RIB of the target VRF.

Note: With this feature enabled, no attempt is made to resolve the VPN route over an MPLS tunnel (even if one exists) in the default VRF. Therefore, the VPN routes received from a neighbor remains inactive in the default VRF.

To enable this feature, use the domain token **vrf-unicast-rib** under the IPV4 / IPV6 address family submodes:

```
switch(config-router-bgp-af)# next-hop resolution ribs vrf-unicast-rib
```

15.5.2.11.4.1 Examples

To enable the **vrf-unicast-rib** feature:

```
router bgp 64512
   address-family vpn-ipv4
      next-hop resolution ribs vrf-unicast-rib
```

The following IPV4 VPN route has been received from a neighbor:

```
switch(config)# show bgp vpn-ipv4
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop
         Network             Next Hop         Metric  LocPref Weight Path
         RD: 11.0.1.1:0 IPv4 prefix 50.1.1.0/24
                             42.42.42.42      -       1       0      100 200 i
```

The route is **inactive** in the default VRF.

Also, there is a VRF, ***CUST-1***, where the VPN route (based on the route-targets) is imported. In the ***CUST-1*** VRF, the nexthop of the route is resolved via a static route to ***42.42.42.42***. With this feature enabled, the VPN route is imported and installed in the VRF ***CUST-1***:

```
switch(config)# show ip bgp vrf CUST-1
```

```
BGP routing table information for VRF CUST-1
Router identifier 11.0.0.1, local AS number 300
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop
        Network            Next Hop         Metric  LocPref Weight Path
      * >      50.1.1.0/24         42.42.42.42      -       1       0      100 200 i
```

To confirm the route is installed in the ***VRF CUST-1***, use the **`show ip route`** command:

```
switch(config)# show ip route vrf CUST-1
VRF: CUST-1
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route
 S      42.42.42.42/32 is directly connected, Null0
 B I    50.1.1.0/24 is directly connected, Null0
```

The resolution profile used to resolve the next-hop can be seen using the **`show rib next-hop`** command. Note how the profile includes (only) the system-unicast-rib for ***CUST-1***.

```
switch(config)# show rib next-hop ip vrf CUST-1 bgp detail
VRF: CUST-1, Protocol: bgp
Codes: * - Unresolved Next hop
       L - Part of a recursive route resolution loop
       A - Next hop not resolved in ARP/ND
11.0.1.1 [1 pref/0 metric] [ID: 18] type ipv4
   Resolution RIBs: system-unicast-rib
   via Null0, directly connected [ID 3]
```

15.5.2.11.5 Viewing the BGP Nexthop Resolution Status

The following existing show commands which have been enhanced for this feature:

- **`show route-msp`** **NAME**
- **`show bgp instance`** [vrf **NAME**]
- **`show rib next-hop`** **{ip | ipv6}[PROTCOL] detail**

show route-map NAME

Use the **`show route-map NAME`** command to display the new set statement:

```
switch(config)# show route-map
route-map foo permit 10
  Description:
  Match clauses:
  SubRouteMap:
  Set clauses:
    set next-hop resolution ribs tunnel-rib system-tunnel-rib
route-map foo permit 20
  Description:
  Match clauses:
  SubRouteMap:
  Set clauses:
```

```
    set next-hop resolution ribs tunnel-rib colored system-colored-tunnel-rib
 tunnel-rib system-tunnel-rib system-unicast-rib

switch(config)#show route-map | json
{
   "routeMaps": {
      "foo": {
         "entries": {
            "20": {
               "setRules": {
                  "resolutionRibProfileConfig": {
                     "resolutionMethods": [
                        {
                           "ribType": "tunnel",
                           "colored": true,
                           "name": "system-colored-tunnel-rib"
                        },
                        {
                           "ribType": "tunnel",
                           "name": "system-tunnel-rib"
                        },
                        {
                           "ribType": "ip",
                           "name": "system-unicast-rib"
                        }
                     ]
                  }
               },
               "subRouteMap": {
                  "name": "",
                  "invert": false
               },
               "filterType": "permit",
               "matchRules": {},
               "description": []
            },
            "10": {
               "setRules": {
                  "resolutionRibProfileConfig": {
                     "resolutionMethods": [
                        {
                           "ribType": "tunnel",
                           "name": "system-tunnel-rib"
                        }
                     ]
                  }
               },
               "subRouteMap": {
                  "name": "",
                  "invert": false
               },
               "filterType": "permit",
               "matchRules": {
               },
               "description": []
            }
         }
      }
   }
},
```

show bgp instance [vrf NAME]

Use the **`show bgp instance`** command to inspect the configured profiles and route-maps for each address family. The diplay output has been extended to show the resolution ribs as seen below, done so in order to display the resolution ribs used for EVPN and BGP Labeled-unicast address families. The output displays the resolution rib profile configuration for the respective address families.

Example

```
switch(config-router-bgp)# show bgp instance
BGP instance information for VRF default
...
Address family IPv4 MplsLabel:
  Additional-paths installation is disabled
  Convergence based update synchronization is disabled
  Target RIBs: Tunnel RIB
    Resolution RIBs: tunnel-rib system-tunnel-rib, system-connected
...
Address family IPv6 MplsLabel:
  Additional-paths installation is disabled
  Convergence based update synchronization is disabled
  Target RIBs: Tunnel RIB
    Resolution RIBs: tunnel-rib system-tunnel-rib, system-connected
...
Address family L2VPN EVPN:
  Additional-paths installation is disabled
  Convergence based update synchronization is disabled
    Vxlan Resolution RIBs: system-unicast-rib
    Mpls Resolution RIBs: tunnel-rib system-tunnel-rib, system-connected
```

Use the **show rib next-hop ip bgp** command to display the per-via resolution profile.

show rib-next-hop {ip,ipv6} [proto] detail

Use the **`show rib next-hop`** **{ip | ipv6}[proto] detail** command to display which resolution profile is used to resolve each next-hop.

Example

```
switch#(config-router-bgp)# show rib next-hop ip bgp detail
VRF: default, Protocol: bgp
Codes: * - Unresolved Next hop
       L - Part of a recursive route resolution loop
       A - Next hop not resolved in ARP/ND
192.0.2.1 [110 pref/20 metric] [ID: 1] type ipv4
   Resolution RIBs: tunnel-rib colored system-colored-tunnel-rib, tunnel-rib
 system-tunnel-rib, system-unicast-rib
   via 198.51.100.1, Ethernet3 [ID: 10]
192.0.2.2 * [ID: 86]
   Resolution RIBs: No profile set for this next-hop
192.0.2.3 * [ID: 78]
   Resolution RIBs: tunnel-rib colored system-colored-tunnel-rib, tunnel-rib
 system-tunnel-rib, system-connected
```

Note how ***192.0.2.2*** has no profile set, and is therefore unresolved. This show command illustrates this clearly with the **`No profile set for this next-hop`** message.

15.5.2.11.6 User-defined Tunnel RIBs for NextHop Resolution

Currently, EOS generates a single system-defined tunnel RIB for the next-hop resolution.

When tunnels to the same destination address are learned from multiple protocols, a fixed preference that is associated with each protocol is used to determine the winning tunnel.

However, with the User-defined tunnel RIBs feature the user is allowed to create user-defined tunnel RIBs with:

- Control over which protocols may contribute to the tunnel RIB.
- The ability to override the preference for all tunnels from a protocol to achieve non-default ordering of tunnels.
- The option to use it in a context where the system-defined tunnel RIB does not suffice.

15.5.2.11.6.1 Configuring User-defined Tunnel RIBs

A new ***tunnel-ribs*** configuration mode allows the creation of user-defined tunnel RIBs. For example, the following configuration creates a tunnel RIB with tunnels learned from IS-IS SR and LDP only, with IS-IS SR tunnels being preferred over LDP:

```
switch(config)# tunnel-ribs
switch(config-tunnel-ribs)# tunnel-rib SR_OVER_LDP
switch(config-tunnel-rib-SR_OVER_LDP)# source-protocol isis segment-routing
 preference 10
switch(config-tunnel-rib-SR_OVER_LDP)# source-protocol ldp preference 20
```

When adding a source protocol in a user-defined tunnel RIB, the preference is optional. A lower preference value indicates a more preferred protocol. If the preference is not specified, the following system-defined preference values are used:

Source Protocol	System-defined Preference
Static	15
Nexthop group tunnel	25
RSVP LER	45
LDP	55
IS-IS SR	65
BGP-LU	85

Modifying the system-tunnel-rib

The user can explicitly modify the default preferences for the **system-tunnel-rib** as well as user-defined RIBs:

```
switch#(config)# tunnel-ribs
switch#(config-tunnel-ribs)# tunnel-rib system-tunnel-rib
switch#(config-tunnel-rib-system-tunnel-rib)#?
  source-protocol  Configure the tunnel source
  ---------------------------------------
  comment          Up to 240 characters, comment for this mode
  default          Set a command to its defaults
  exit             Leave Configure mode
  no               Disable the command that follows
  show             Display details of switch operation
  !!               Append to comment

switch#(config-tunnel-rib-system-tunnel-rib)# source-protocol ?
  bgp            BGP tunnel
  isis           IS-IS tunnel
  ldp            LDP tunnel
  nexthop-group  Nexthop group tunnel
  rsvp-ler       RSVP LER tunnel
```

```
  static         Static tunnel

switch#(config-tunnel-rib-system-tunnel-rib)# source-protocol rsvp-ler
 preference 2
switch#(config-tunnel-rib-system-tunnel-rib)# exit
switch#(config-tunnel-ribs)# show active all
tunnel-ribs
   tunnel-rib system-tunnel-rib
      source-protocol static
      source-protocol isis segment-routing
      source-protocol bgp labeled-unicast
      source-protocol nexthop-group
      source-protocol rsvp-ler preference 2
      source-protocol ldp
```

15.5.2.11.6. Displaying Tunnel RIB Information

- Use the **show tunnel rib** to display the user defined RIB information:

```
switch# show tunnel rib SR_OVER_LDP brief
Tunnel RIB: SR_OVER_LDP
Endpoint        Tunnel Type   Index(es) Tunnel Preference  IGP Preference  IGP Metric
--------------- ------------- --------- ------------------ --------------- ----------
1.1.1.1/32      IS-IS SR IPv4 2         10                 115             20
```

- Use the **show tunnel rib brief** command to display the system-defined tunnel RIB information.

```
switch# show tunnel rib brief
Tunnel RIB: system-tunnel-rib
Endpoint        Tunnel Type   Index(es) Tunnel Preference  IGP Preference  IGP Metric
--------------- ------------- --------- ------------------ --------------- ----------
1.1.1.1/32      LDP           1         55                 1               0
```

- Use the **show active all** command to display the information about which source protocols contribute to the system-defined tunnel RIB.

```
switch(config)# tunnel-ribs
switch(config-tunnel-ribs)# tunnel-rib system-tunnel-rib
switch(config-tunnel-ribs)# show active all
tunnel-ribs
   tunnel-rib system-tunnel-rib
      source-protocol static
      source-protocol isis segment-routing
      source-protocol bgp labeled-unicast
      source-protocol nexthop-group
      source-protocol rsvp-ler
      source-protocol ldp
```

15.5.2.12 Configuring BGP Confederations

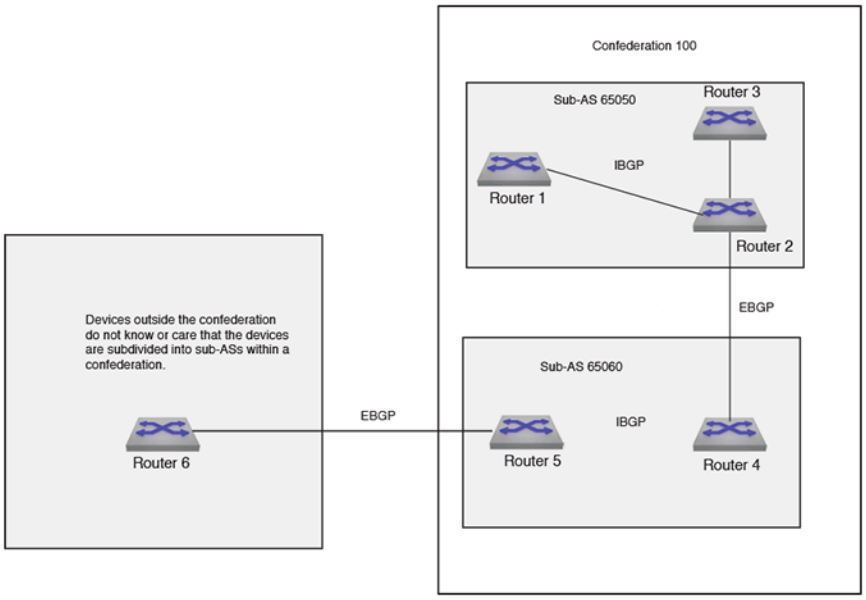
BGP confederations allow you to break an Autonomous System (AS) into multiple sub-ASs, and then to group the sub-ASs as a confederation. The sub-ASs exchange iBGP routing information (next-hop, local-preference and MED), but communicate via eBGP.

To configure a BGP confederation, complete the following tasks on each BGP device in the confederation.

- **Configure the local AS number**: the local AS number is the membership number in a sub-AS. BGP devices with the same local AS number are identified as members of the same sub-AS. BGP devices always use the local AS number when communicating with other BGP4 devices in the confederation.
- **Configure the confederation ID**: the confederation ID is the AS number for those BGP devices that are outside of the confederation. A BGP device outside the confederation is not aware that BGP devices are in multiple sub-ASs. The confederation ID must differ from the sub-AS numbers.

- **Configure the list of sub-AS numbers that are confederation members**: devices in a sub-AS exchange information via iBGP, while devices in different sub-ASs use eBGP.

Figure 15-15: BGP Confederation Example



Example

- The **router bgp** command enables BGP and configures the router in sub-autonomous system ***65050***. The **bgp confederation identifier** command specifies that confederation ***65050*** belongs to autonomous system ***100***.

  The neighbors from other autonomous systems within the confederation are treated as special eBGP peers when using the **bgp confederation peers** command.

```
switch(config)# router bgp 65050
switch(config-router-bgp)# bgp confederation identifier 100
switch(config-router-bgp)# bgp confederation peers 65060
switch(config-router-bgp)#
```

- The Arista EOS will group the maximum ranges together. In this example, peers ***65032*** and ***65036*** are not included in BGP confederation ***100***.

```
switch(config)# router bgp 65050
switch(config-router-bgp)# bgp confederation identifier 100
switch(config-router-bgp)# bgp confederation peers 65060
switch(config-router-bgp)# no bgp confederation peers 65032, 65036
switch(config-router-bgp)#
```

15.5.2.13 Configuring BGP Flowspec

The BGP Flowspec address family is enabled on a per-peer basis with:

Example

```
switch(config)# router bgp id
switch(config-router-bgp)# address-family flow-spec [ipv4 | ipv6]
```

```
switch(config-router-bgp-af)# neighbor address activate
```

Flowspec has to be explicitly enabled on an interface, with:

Example

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# flow-spec ipv4 ipv6
```

Currently, both IPv4 and IPv6 must be enabled together on the interface. A user defined TCAM profile, a feature introduced in ***EOS Relaease 4.20.5F***, must be configured for TCAM support for flowspec.

Warning: Creating user-defined TCAM profile on the Arista switch could cause serious issues that impact traffic. You should test flowspec policer with the profile given in this document. If you need to add new features in the profile, work with Arista's TAC team to define and test the new profile before deploying it on your production switches.

The ACL counters and Flowspec counters cannot be enabled simultaneously. To enable reporting of counters for flow-spec rules, use the following configuration:

Example

```
switch(config)# no hardware counter feature acl in
switch(config)# hardware counter feature flow-spec in
```

15.5.2.13.1 Displaying Flowspec Information

The BGP show commands have been enhanced to display the **`flow-spec`** content for both IPv4 and IPv6 address families:

Example

The **`show bgp flow-spec ipv4 summary`** command displays the count of flowspec rules received from each peer:

```
switch(config)# show bgp flow-spec ipv4 summary
BGP summary information for VRF default
Router identifier 0.0.0.1, local AS number 10
Neighbor Status Codes: m - Under maintenance
  Neighbor     V  AS    MsgRcvd   MsgSent  InQ OutQ  Up/Down State  RulesRcd RulesAcc
  10.0.0.2     4  10         12         4    0    0 00:02:18 Estab  2        2
  10.0.1.2     4  10          6         4    0    0 00:02:18 Estab  0        0
```

The **`show bgp flow-spec ipv4`** displays a brief description of each flowspec rule, including the matching rule and actions. The matching rule uses a format:

dest prefix; **src prefix**; [**component:condition**] **+**

The component is abbreviated, for example, DP for destination port and IP for IP Protocol as shown in the following example. The detail of the show command will display the full component name.

The condition is expressed with logical operators. In the following example, **`IP:=6|=17`** matches any packets whose IP Protocol is 6 (TCP) or 17 (UDP). **`DP:>1010&<1024`** matches any packets whose destination port is greater than ***1010*** and less than ***1024***.

Example

```
switch(config)# show bgp flow-spec ipv4
BGP Flow Specification rules for VRF default
Router identifier 0.0.0.1, local AS number 10
Rule status codes: # - not installed, M - received from multiple peers

   Matching Rule
 Actions
   10.2.3.0/24;*;                                                  Drop
   10.2.4.0/24;10.2.0.0/16;IP:=6|=17;DP:>1010&<1024;               Drop
```

The **`show bgp flow-spec detail`** displays the full details of each flowspec rule including the peer(s) it was received from, BGP properties, and an expanded description of the matching rule:

Example

```
switch(config)# show bgp flow-spec ipv4 detail
BGP Flow Specification rules for VRF default
Router identifier 0.0.0.1, local AS number 10
BGP Flow Specification Matching Rule for 10.2.3.0/24;*;
 Rule identifier: 3882065752
 Matching Rule:
   Destination Prefix: 10.2.3.0/24
   Source Prefix: *
 Paths: 1 available
  Local
    from 10.0.0.2 (10.1.1.2)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal,
 best
      Actions: Drop
BGP Flow Specification Matching Rule for 10.2.4.0/24;10.2.0.0/16;IP
:=6|=17;DP:>1010&<1024;
 Rule identifier: 3882090640
 Matching Rule:
   Destination Prefix: 10.2.4.0/24
   Source Prefix: 10.2.0.0/16
   IP Protocol: =6 | =17
   Destination Port: >1010 & <1024
 Paths: 1 available
  Local
    from 10.0.0.2 (10.1.1.2)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal,
 best
      Actions: Drop
```

The **`show flow-spec ipv4 summary`** command displays an overall status of how many flowspec rules were received and how many were installed:

Example

```
switch(config)# show flow-spec ipv4 summary
```

```
Flow specification rules summary for VRF default
  Total number of rules: 2
  Number of installed rules: 2
```

The **show flow-spec ipv4** displays the installation status of the rule, and a counter of how many hits it has accumulated. This command also compiles the received flowspec rules into rules that can be programmed into the TCAM. For example, logical expressions on values such as the destination port are converted to ranges, as shown below:

Example

```
switch(config)# show flow-spec ipv4
Flow specification rules for VRF default
Applied on: Ethernet47/1
  Flow-spec rule: 10.2.3.0/24;*;
    Rule identifier: 3882065752
    Matches:
      Destination prefix: 10.2.3.0/24
    Actions:
      Police: 80 Mbps (10 MBps)
      Redirect: VRF customer1
                Route via LDP tunnel index 4, MPLS label 100123
                Route via LDP tunnel index 1, MPLS label 116507
    Status:
      Installed: yes
      Counter: 312 packets
  Flow-spec rule: 10.2.4.0/24;10.2.0.0/16;IP:=6|=17;DP:>1010&<1024;
    Rule identifier: 3882090640
    Matches:
      Destination prefix: 10.2.4.0/24
      Source prefix: 10.2.0.0/16
      Next protocol: 17
                     6
      Destination port: 1011-1023
    Actions:
      Police: 80 Mbps (10 MBps)
      Redirect: VRF customer1
                Route via LDP tunnel index 4, MPLS label 100123
                Route via LDP tunnel index 1, MPLS label 116507
    Status:
      Installed: yes
      Counter: 0 packets
```

Infeasible rules are detected and not programmed, and this status is reported using the **show flow-spec ipv4** command. Examples of the infeasible rules are:

- The **lt/gt/eq** operator is missing in the numerical opVal component.
- Co-existence of TCP flag component with ICMP type or code component in the same rule.
- Co-existence of port based component with ICMP type or code component in the same rule.For example, the default route in the specified VRF can be resolved over a GRE or MPLS tunnel. The following **show** command output verifies the resolution over the tunnel:

For redirect actions, additional information is displayed to show how it was resolved.

Example

```
Actions:
```

```
      Redirect: VRF customer1
                Route via LDP tunnel index 4, MPLS label 100123
                Route via LDP tunnel index 1, MPLS label 116507
```

The specified **nexthop** in the flow-spec **redirect** action can be resolved by the respective VRFs IP RIB over MPLS or GRE tunnel, as shown in the following example:

Example

```
Actions:
      Redirect: VRF default, fc00:91:91:91::91
                Route via Static Interface tunnel index 1
```

15.5.2.14 Configuring BGP Logical OR of Multiple Community Lists

Adding the **or-results** token to the **match community** command allows you to do a logical OR between all provided community lists:

```
match community or-results COMMLIST1 COMMLIST2
match extcommunity or-results EXTCOMMLIST1 EXTCOMMLIST2
match large-community or-results LARGECOMMLIST1 LARGECOMMLIST2
```

Full configuration example:

- Enable the Multi-agent mode.

```
switch(config)#service routing protocols model multi-agent
```

- Create community lists (extended and large communities are also compatible with **or-results**).

```
switch(config)#ip community-list COMMLIST1 permit 1:1
switch(config)#ip community-list COMMLIST2 permit 2:2
```

- Configure Route-map with **or-results**.

```
switch(config)#route-map IN-POLICY
switch(config-route-map-IN-POLICY)#match community or-results
                                   COMMLIST1 COMMLIST2
```

15.5.2.14.1 Displaying BGP Logical OR Information

The **or-results** match clauses can be seen with the standard **show route-map** command, as shown in the following command display outputs.

```
switch# show route-map IN-POLICY
route-map IN-POLICY permit 10
  Description:
  Match clauses:
    match community or-results COMMLIST1 COMMLIST2
  SubRouteMap:
  Set clauses:
    set local-preference 500
route-map IN-POLICY permit 20
  Description:
  Match clauses:
  SubRouteMap:
```

```
  Set clauses:
```

```
switch# show run | in 200.200.200.57
   neighbor 200.200.200.57 remote-as 300
   neighbor 200.200.200.57 update-source Loopback200
   neighbor 200.200.200.57 ebgp-multihop
   neighbor 200.200.200.57 route-map IN-POLICY in
   neighbor 200.200.200.57 maximum-routes 0
```

```
switch# show ip bgp community 1:1
BGP routing table information for VRF default
Router identifier 220.220.220.51, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

         Network              Next Hop             Metric  LocPref Weight  Path
 * >     66.170.224.0/20      200.200.200.57       0       500     0       300 ?
 * >     66.170.232.0/21      200.200.200.57       0       500     0       300 ?
 * >     128.29.0.0/16        200.200.200.57       0       500     0       300 ?
```

```
switch# show ip bgp community 2:2
BGP routing table information for VRF default
Router identifier 220.220.220.51, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

         Network              Next Hop             Metric  LocPref Weight  Path
 * >     192.12.24.0/24       200.200.200.57       0       500     0       300 ?
 * >     192.47.242.0/24      200.200.200.57       0       500     0       300 ?
```

15.5.2.15 Setting the BGP Missing Policy Action

To set the default policy behavior for BGP so that all routes can be denied or rejected, use the **bgp missing policy** command. Options control inbound and outbound directions independently. When the inbound direction is affected, currently installed routes from the peer are removed (and withdrawn from other attached peers). When the outbound direction is affected, currently exported routes to the peer are withdrawn. Setting the Missing Policy Action options back to its default/permit value re-applies the current inbound route-map policy processing to the set of routes received from the peer and export routes according to the configured outbound route-map. If soft-reconfiguration is disabled and the inbound direction is affected then the peer must re-send its routes (e.g. a manual “clear ip bgp” command is required).

Configuring the BGP Missing Policy Action

Permit is the default missing policy action when no/default are applied. Entering the ‘default’ form of the command in a non-default VRF will cause the non-default VRF to inherit the setting from the default VRF. Entering the **no** form of the command in a non-default VRF will cause the non-default VRF to be configured with the **permit** setting regardless of the default VRF setting.

The include keyword is optional, and only takes effect in the multi-agent protocol model.

The following configures the BGP missing policy action.

```
switch(config-router-bgp)# bgp missing-policy [include {prefix-list|sub-route-
map}]
direction [in|out] action [permit|deny|deny-in-out]
```

```
switch(config-router-bgp)# [no|default] bgp missing-policy [include {prefix-
list|sub-route-map}]
direction [in|out] action
```

Actions

For the actions, the **permit** and **deny** options inherit the direction of route denial from the direction, while the **deny-in-out** option specifically calls out denying routes in both directions.

- direction **in** action **permit**: allow all routes in the inbound direction when the inbound route-map is misconfigured (default).
- direction **out** action **permit**: allow all routes in the outbound direction when the outbound route-map is misconfigured (default).
- direction **in** action **deny**: deny all routes in the inbound direction when the inbound route-map is misconfigured
- direction **out** action **deny**: deny all routes in the outbound direction when the outbound route-map is misconfigured
- direction **in** action **deny-in-out**: deny all routes in both inbound/outbound directions when the inbound route-map is misconfigured
- direction **out** action **deny-in-out**: deny all routes in both inbound/outbound directions when the outbound route-map is misconfigured

The include keyword specifies that the policy constructs in the route map should also be examined. The options to the include keyword are.

- **sub-route-map**: examine the sub route map references if any are defined in a route map covered by a missing policy statement. If the sub route map statement – or any in a route map chain – makes a reference to a route map which does not exist, then the missing policy action will be applied.
- **prefix-list**: examine the prefix list references if any are defined in a route map covered by a missing policy statement. If the prefix-list statement – or any in the applied route map chain – makes a reference to a prefix list which does not exist, then the missing policy action will be applied.

Displaying BGP Missing Policy Action Configurations

The **`show ip bgp neighbors`** command displays the status of a peer which is currently in the missing policy/default deny state.The **`Missing policy/default deny`** lines would be omitted if the configuration option is disabled or the route-maps are configured correctly.

```
switch(config-router-bgp)# show ip bgp neighbors
BGP neighbor is 1.0.0.2, remote AS 200, external link
BGP version 4, remote router ID 0.0.1.1, VRF default
Negotiated BGP version 4
…
Missing policy/default deny import action is active
Missing policy/default deny export action is active
Inbound route map is rm1
Outbound route map is rm2
…
```

15.5.2.16 Configuring BGP IPv4-mapped IPv6 Address Next Hops for IPv6 Labeled-Unicast Routes

Receive-side Configuration

To configure BGP to translate IPv4-mapped IPv6 addresses to IPv4 addresses when receiving next hops in labeled-unicast routes, use the neighbor next-hop resolution v4-mapped-v6 translation command. With this configuration, when the switch receives an IPv4-mapped IPv6 address for the next hop of an IPv6 labeled-unicast route, it will translate it to an IPv4 address, which allows the next hop to be resolved in an IPv4

network. This command takes effect only if the multi-agent routing protocol model is running. It applies only to the default VRF.

Example

These commands enter BGP IPv6 Labeled-Unicast Address Family Configuration Mode for AS ***64510*** (creating the BGP instance if it does not exist) and enable the translation of IPv4-mapped IPv6 addresses to IPv4 addresses for neighbors in the ***v6_pg*** peer group.

```
switch(config)# router bgp 64510
switch(config-router-bgp)# address-family ipv6 labeled-unicast
switch(config-router-bgp-af-label)# neighbor v6_pg next-hop resolution v4-
mapped-v6 translation
switch(config-router-bgp-af-label)#
```

Send-side Configuration

A BGP router advertising a route can provide the IPv4-mapped IPv6 address of one of its local interfaces, such as a loopback interface, as the next hop. This source interface is specified with the **neighbor next-hop-self** command. The interface must be configured with an IPv4 address for this to be effective.

This configuration does not enable next-hop-self. It simply specifies the interface to be provided if the router advertises itself as the next hop. The next-hop-self action can be enabled with the **neighbor next-hop-self** command, or by configuring Egress Peer Engineering (EPE) using the **neighbor default-originate** command, or by other methods.

Example

These commands enable the switch to advertise itself as a next hop for the peer at ***2001:0db8::1***, and then configure the switch to use the IPv4 address of the ***Loopback 0*** interface for the next hop for the peer at ***2001:0db8::1*** if the route is IPv4-mapped IPv6.

```
switch(config)# router bgp 64510
switch(config-router-bgp)# neighbor 2001:0db8::1 next-hop-self
switch(config-router-bgp)# neighbor 2001:0db8::1 next-hop-self v4-mapped-v6
 source-interface Loopback 0
switch(config-router-bgp)#
```

15.5.2.17 BGP Operational Commands

15.5.2.17.1 Shutdown

The **shutdown (BGP)** command disables BGP operations without disrupting the BGP configuration. The **no router bgp** command disables BGP and removes the BGP configuration.

The **no shutdown (BGP)** command resumes BGP activity.

Examples

- This command disables BGP activity on the switch.

```
switch(config-router-bgp)# shutdown
switch(config-router-bgp)#
```

- This command resumes BGP activity on the switch.

```
switch(config-router-bgp)# no shutdown
switch(config-router-bgp)#
```

15.5.2.17.2 Clearing the Routing Table and Resetting BGP Sessions

When entered without parameters, the **clear ip bgp** command clears all BGP learned routes from the routing table, reads routes from designated peers, and sends routes required by those peers. Routes that are read or sent are processed through any modified route map or AS-path access list.

Followed by an asterisk (*****), it clears the BGP sessions with all BGP peers. To reset the session with a specific peer, enter the peer's IP address at the end of the command.

Example

This command removes all BGP learned routes from the routing table.

```
switch# clear ip bgp
! Peerings for all neighbors were hard reset switch#
```

15.5.3 BGP IPv6 Link Local Peers Discovery

BGP IPv6 Link Local Peers Discovery supports a dynamic configuration model to eliminate the need for the network administrator to assign and configure IPv6 addresses for BGP peering.

Leverage the following details to automatically establish BGP adjacency:

- IPv6 link local addresses are automatically generated by the system based on MAC addresses.
- IPv6 router advertisements are used to communicate these addresses among potential BGP peers.

BGP IPv6 Link Local Peers Discovery uses IPv6 router advertisement to discover the peers IPv6 link local address. Devices are required to have IPv6 routing enabled, and the interface used for peering must have an IPv6 link local address. The time taken to discover the peers IPv6 link local address is proportional to the time taken by the peer to send a router advertisement message. When bringing up BGP sessions based on router advertisements received, a flurry of router advertisements on the interfaces causes the Rib agent to do more work and potentially delays the discovery of BGP neighbors over those interfaces and the establishment of BGP sessions. Since these are link local addresses, the peers must be directly connected at Layer 3.

15.5.4 BGP Examples

This section shows example configurations and topologies for iBGP (BGP Example 1) and eBGP (BGP Example 2).

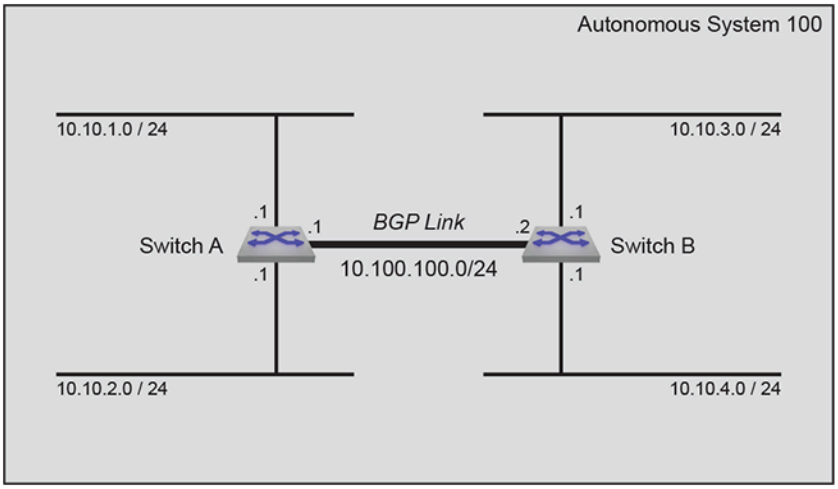
15.5.4.1 BGP Example 1

Example 1 features an internal BGP (iBGP) link that connects peers in AS ***100***.

15.5.4.1.1 BGP Example 1 Diagram

Figure 60: BGP Example 1 displays an iBGP connection, linking neighbors within AS ***100***. Each switch advertises two subnets. In UPDATE packets sent by Switch A, the **LOCAL_PREF** field is ***150***. In UPDATE packets sent by Switch B, the **LOCAL_PREF** field is ***75***.

Figure 15-16: BGP Example 1



15.5.4.1.2 BGP Example 1 Code

This code configures the Example 1 BGP instance on both switches.

1. Configure the neighbor addresses.
   a. Specify the neighbor to ***Switch A***.

```
switchA(config)# router bgp 100
switchA(config-router-bgp)# neighbor 10.100.100.2 remote-as 100
```

   b. Specify the neighbor to ***Switch B***.

```
switchB(config)# router bgp 100
switchB(config-router-bgp)# neighbor 10.100.100.1 remote-as 100
```

2. Configure the routes to be advertised.
   a. Advertise ***Switch A***'s routes.

```
switchA(config-router-bgp)# network 10.10.1.0/24
switchA(config-router-bgp)# network 10.10.2.0/24
```

   b. Advertise ***Switch B***'s routes.

```
switchB(config-router-bgp)# network 10.10.3.0/24
switchB(config-router-bgp)# network 10.10.4.0/24
```

3. Configure the **LOCAL_PREF**.
   a. Configure **LOCAL_PREF** on ***Switch A***.

```
switchA(config-router-bgp)# neighbor 10.100.100.2 export-localpref 150
```

   b. Configure **LOCAL_PREF** on ***Switch B***.

```
switchB(config-router-bgp)# neighbor 10.100.100.2 export-localpref 75
```

4. Modify the hold time and keepalive interval.
   a. Configure timers on ***Switch A***.

```
switchA(config-router-bgp)# timer bgp 30 90
```

   b. Configure timers on ***Switch B***.

```
switchB(config-router-bgp)# timer bgp 30 90
```

15.5.4.2 BGP Example 2

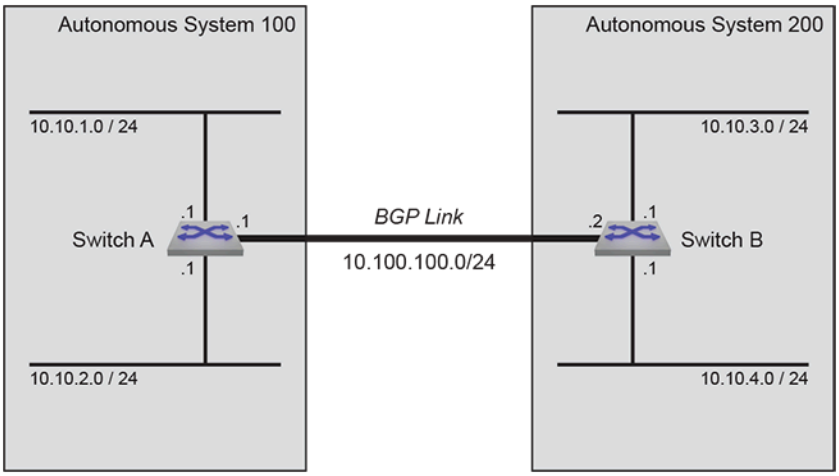
Example 2 creates an external BGP (eBGP) link that connects routers in AS ***100*** and AS ***200***.

15.5.4.2.1 BGP Example 2 Diagram

Figure 61: BGP Example 2 displays an eBGP connection, linking ***Switch A*** in AS ***100*** to ***Switch B*** in AS ***200***. Each switch advertises two subnets.

Switch A assigns a local preference of ***150*** to networks advertised by ***Switch B***. ***Switch B*** assigns a local preference of ***75*** to networks advertised by ***Switch A***.

Figure 15-17: BGP Example 2



15.5.4.2.2 BGP Example 2 Code

This code configures the Example 2 BGP instance on both switches.

1. Configure the neighbor addresses.
   a. Specify the neighbor to ***Switch A***.

```
switchA(config)# router bgp 100
switchA(config-router-bgp)# neighbor 10.100.100.2 remote-as 200
```

   b. Specify the neighbor to ***Switch B***.

```
switchB(config)# router bgp 200
switchB(config-router-bgp)# neighbor 10.100.100.1 remote-as 100
```

2. Configure the routes to be advertised.
   a. Advertise ***Switch A***'s routes.

```
switchA(config-router-bgp)# network 10.10.1.0/24
```

```
switchA(config-router-bgp)# network 10.10.2.0/24
```

b. Advertise ***Switch B***'s routes.

```
switchB(config-router-bgp)#network 10.10.3.0/24
switchB(config-router-bgp)#network 10.10.4.0/24
```

3. Configure the ***LOCAL_PREF***.

a. Configure ***LOCAL_PREF*** on ***Switch A***.

```
switchA(config-router-bgp)# neighbor 10.100.100.2 import-localpref 150
```

b. Configure ***LOCAL_PREF*** on ***Switch B***.

```
switchB(config-router-bgp)# neighbor 10.100.100.2 import-localpref 75
```

4. Modify the hold time and keepalive interval.

a. Configure timers on ***Switch A***.

```
switchA(config-router-bgp)# timer bgp 30 90
```

b. Configure timers on ***Switch B***.

```
switchB(config-router-bgp)# timer bgp 30 90
```

15.5.5 BGP Commands

Global Configuration Commands

- ip as-path access-list
- ip as-path regex-mode
- ip community-list
- ip community-list regexp
- ip extcommunity-list
- ip extcommunity-list regexp
- ip large-community-list regexp
- router bgp

Router General Command

- rib fib fec ecmp ordered

Router-BGP Configuration Mode (Includes Address-Family Mode)

- address-family
- address-family flow-spec
- aggregate-address
- bgp advertise-inactive
- bgp always-compare-med
- bgp bestpath as-path ignore
- bgp bestpath as-path multipath-relax
- bgp bestpath ecmp-fast
- bgp bestpath med confed
- bgp bestpath med missing-as-worst
- bgp bestpath tie-break cluster-list-length

- bgp bestpath tie-break router-id
- bgp client-to-client reflection
- bgp cluster-id
- bgp confederation identifier
- bgp confederation peers
- bgp convergence slow-peer time
- bgp convergence time
- bgp default
- bgp enforce-first-as
- bgp listen range
- bgp log-neighbor-changes
- bgp redistribute-internal (BGP)
- bgp route install-map
- bgp route-reflector preserve-attributes
- distance bgp
- dynamic peer max
- graceful-restart stalepath-time
- graceful-restart-helper
- match as-range
- maximum paths (BGP)
- neighbor
- neighbor activate
- no neighbor
- neighbor allowas-in
- neighbor auto-local-addr
- neighbor default-originate
- neighbor description
- neighbor ebgp-multihop
- neighbor enforce-first-as
- neighbor export-localpref
- neighbor graceful-restart
- neighbor graceful-restart-helper
- neighbor import-localpref
- neighbor local-as
- neighbor local-v4-addr
- neighbor local-v6-addr
- neighbor maximum-routes
- neighbor next-hop-peer
- neighbor next-hop-self
- neighbor next-hop resolution v4-mapped-v6 translation
- neighbor out-delay
- neighbor passive
- neighbor password
- neighbor peer group (create)
- neighbor peer group (neighbor assignment)
- neighbor remote-as
- neighbor remove-private-as
- neighbor rib-in pre-policy retain
- neighbor route-map (BGP)
- neighbor route-reflector-client

- neighbor route-to-peer
- neighbor send-community
- neighbor send-community add / remove
- neighbor send-community link-bandwidth
- neighbor shutdown
- neighbor timers
- neighbor ttl maximum-hops
- neighbor update-source
- neighbor weight
- network (BGP)
- peer-filter
- rd (Router-BGP VRF and VNI Configuration Modes)
- redistribute (BGP)
- router-id (BGP)
- shutdown (BGP)
- timers bgp
- update wait-for-convergence
- vrf

Route Map Configuration Mode

- set as-path match
- set large-community

Clear Commands Privileged EXEC Mode

- clear bgp history
- clear ip bgp
- clear ip bgp counters
- clear ip bgp errors
- clear ip bgp neighbor
- clear ipv6 bgp
- clear ipv6 bgp counters
- clear ipv6 bgp errors
- clear ipv6 bgp neighbor

Display Commands EXEC Mode

- show bgp convergence
- show bgp flow-spec
- show bgp instance
- show bgp labeled-unicast tunnel
- show bgp neighbors history
- show bgp update-group
- show flow-spec
- show ip as-path access-list
- show ip bgp
- show ip bgp community
- show ip bgp installed
- show ip bgp neighbors
- show ip bgp neighbors (route type)
- show ip bgp neighbors (route-type) community

- show ip bgp neighbors regexp
- show ip bgp not-installed
- show ip bgp paths
- show ip bgp peer-group
- show ip bgp regexp
- show ip bgp summary
- show ip community-list
- show ip extcommunity-list
- show ipv6 bgp
- show ipv6 bgp match community
- show ipv6 bgp peers
- show ipv6 bgp peers (route type)
- show ipv6 bgp peers (route type) community
- show ipv6 bgp peers regexp
- show ipv6 bgp regexp
- show ipv6 bgp summary
- show peer-filter
- show run|section bgp
- show tunnel rib brief

15.5.5.1 address-family

The **address-family** command places the switch in address-family configuration mode to configure the address family setting of addresses configured as BGP neighbors. The address-family configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

The ***running-config*** displays the **address-family** commands in sub-blocks of the BGP configuration. The following commands are available in address family configuration mode:

- **neighbor activate** configures the address as active in the configuration mode address family.
- **no neighbor activate** configures the address as not active in the configuration mode address family.
- **neighbor default-originate** advertises a default route to the specified BGP neighbor.
- **neighbor route-map (BGP)** applies a route map to the specified BGP route.
- **network (BGP)** specifies a network for advertisement through UPDATE packets to BGP peers.

The **no address-family** and **default address-family** commands delete the specified address family from ***running-config*** by removing all commands previously configured in the corresponding address-family mode.

The **exit** command returns the switch to router-BGP configuration mode.

Command Mode

Router-BGP Configuration

Command Syntax

bgp [ipv4 | ipv6]

no bgp [ipv4 | ipv6]

default bgp [ipv4 | ipv6]

Parameters

- **ipv4** subsequent commands are applied to the IPv4 unicast address family.
- **ipv6** subsequent commands are applied to the IPv6 unicast address family.

Example

These commands enter address family mode for IPv6-unicast, insert a command, then exit the mode:

```
switch(config)# router bgp 1
switch(config-router-bgp)# address-family ipv6
switch(config-router-bgp-af)# neighbor 172.10.1.1 activate
switch(config-router-bgp-af)# exit
switch(config-router-bgp)#
```

15.5.5.2 address-family flow-spec

Use the **address-family flow-spec** command to filter or redirect DDoS traffic on edge routers. The **no** and **default** versions of the command removes the filter to redirect the DDoS traffic.

Command Mode

BGP router configuration mode (config-router-bgp)

Command Syntax

address-family flow-spec [ipv4 | ipv6]

no address-family flow-spec [ipv4 | ipv6]

default address-family flow-spec [ipv4 | ipv6]

Parameters

- **ipv4** IPv4 flow specifications.
- **ipv6** IPv6 flow specifications.

Example

The BGP Flowspec address family is enabled on a per-peer basis with:

```
switch(config)# router bgp id
switch(config-router-bgp)# address-family flow-spec [ipv4|ipv6]
switch(config-router-bgp-af)# neighbor address activate
```

15.5.5.3 aggregate-address

The **aggregate-address** command creates an aggregate route in the Border Gateway Protocol (BGP) database. Aggregate routes combine the characteristics of multiple routes into a single route that the switch advertises. Aggregation can reduce the amount of information that a BGP speaker is required to store and transmit when advertising routes to other BGP speakers. Aggregate routes are advertised only after they are redistributed.

The advertised address of the aggregate is entered as an IP subnet; any routes configured on the switch that lie within that subnet then become contributors to the aggregate. Note that on Arista switches the BGP aggregate route will become active if there are any available contributor routes on the switch, regardless of the originating protocol. This includes routes configured statically.



Note: Aggregate routes are redistributed automatically, and their redistribution cannot be disabled.

Command options affect the attributes associated with the aggregated route, the advertisement of the contributor routes that comprise the aggregate, and which contributor routes are included.

Command options affect the following aggregate routing attributes:

- **AS_PATH attribute inclusion**: the **as-set** option controls the aggregate route's **AS_PATH** and **ATOMIC_AGGREGATE** attribute contents. **AS_PATH** identifies the autonomous systems through which UPDATE message routing information passes. **ATOMIC_AGGREGATE** indicates that the route is an aggregate or summary of more specific routes.

  When the command includes **as-set**, the aggregate route's **AS_SET** attribute contains the AS numbers of contributor routes. This can help BGP neighbors to prevent loops by rejecting aggregate routes that include their AS number in the **AS_SET**.

  When the command does not include **as-set**, the aggregate route's **ATOMIC_AGGREGATE** attribute is set and the aggregate route **AS_PATH** will include the longest leading **PATH_SEQ** of the **AS_PATH** which is common to all contributor routes. For example, for the aggregate ***1.0.0.0/16*** with two contributors present, the **AS_PATH** for the aggregate is ***100 200*** as shown.

  Aggregate

  1.0.0.0/16 as-path ??

  Contributors

  1.0.1.0/24 **as-path** ***100 200 400 500***

  1.0.2.0/24 as-path 100 200 300
- **Attribute assignment**: the **attribute-map** option assigns attributes contained in set commands in a specified route map's lowest sequence with any set command to the aggregated route, overriding the automatic determination of the aggregate route's attributes by the switch.
- **Route suppression**: the ***summary-only*** option suppresses the advertisement of the contributor routes that comprise the aggregate.
- **Contributor filtering**: the ***match-map*** option uses a route map to filter out contributor routes that would otherwise be included in the aggregate.

The **`no aggregate-address`** and **`default aggregate-address`** commands remove the corresponding **`aggregate-address`** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

`aggregate-address` **AGGREGATE_NET** [**AS_SET**][**SUMMARY**][**ATTRIBUTE_MAP**][**MATCH_MAP**]

`no aggregate-address` **AGGREGATE_NET**

`default aggregate-address` **AGGREGATE_NET**

Parameters

- **AGGREGATE_NET** aggregate route IP address. Options include:
  - ***netv4_addr*** IPv4 subnet address (CIDR or address-mask notation).
  - ***netv6_addr*** IPv6 subnet address (CIDR notation).
- **AS_SET** controls **AS_PATH** attribute values associated with aggregate route. Options include:
  - ***no parameter*** **ATOMIC_AGGREGATE** attribute is set. Route contains no **AS_PATH** data.
  - **as-set** route includes **AS_PATH** information from contributor routes as **AS_SET** attributes.
- **SUMMARY** controls advertisement of contributor routes. Options include:
  - ***no parameter*** contributor and aggregate routes are advertised.
  - **summary-only** contributor routes are not advertised.
- **ATTRIBUTE_MAP** controls attribute assignments to the aggregate route. Options include:
  - ***no parameter*** attribute values are not assigned to route.
  - **attribute-map** ***map_name*** assigns attribute values in set commands of the map's permit clauses. Deny clauses and match commands in permit clauses are ignored.

- **MATCH_MAP** filters contributors to the aggregate route. Options include:
  - ***no parameter*** no contributors are filtered.
  - **match-map *map_name*** filters contributor routes using the named match-map.

Examples

- These commands create an aggregate route (***10.16.48.0/20***) from the contributor routes ***10.16.48.0/23***, ***10.16.50.0/23***, ***10.16.52.0/23***, and ***10.16.54.0/23***. The aggregate route includes the **AS_PATH** information from the contributor routes.

```
switch(config)# router bgp 1
switch(config-router-bgp)# aggregate-address 10.16.48.0/20 as-set
switch(config-router-bgp)# exit
switch(config)#
```

- These commands create an aggregate route and use a route map to add a local-preference attribute to the route.

```
switch(config)# route-map map1 permit 10
switch(config-route-map-map1)# set community 45
switch(config-route-map-map1)# exit
switch(config)# router bgp 1
switch(config-router-bgp)# aggregate-address 10.16.48.0/20 attribute-map
 map1
switch(config-router-bgp)# exit
switch(config)#
```

- These commands create an aggregate route and use a route map to allow only those contributors which match a specified prefix list to be included in the aggregate route.

```
switch(config)# route-map matchmap permit 10
switch(config-route-map-matchmap)# match ip address prefix-list agglist
switch(config-route-map-matchmap)# exit
switch(config)# router bgp 1
switch(config-router-bgp)# aggregate-address 1.1.0.0/16 match-map matchmap
switch(config-router-bgp)#
```

15.5.5.4 bgp advertise-inactive

By default, BGP will advertise only those routes that are active in the switch's RIB. This can contribute to dropped traffic. If a preferred route is available through another protocol (like OSPF), the BGP route will become inactive and not be advertised; if the preferred route is lost, there is no available route to the affected peers. Advertising inactive BGP routes minimizes traffic loss by providing alternative routes.

The **bgp advertise-inactive** command configures BGP to advertise inactive routes to BGP neighbors. Inactive route advertisement is configured globally, but the global setting can be overridden on a per-VRF basis.

The **no bgp advertise-inactive** and **default bgp advertise-inactive** commands restore the default BGP behavior (advertising only active routes) by removing the corresponding bgp advertise-inactive command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

bgp advertise-inactive

no bgp advertise-inactive

default bgp advertise-inactive

Example

These commands configure BGP to advertise inactive routes.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# bgp advertise-inactive
switch(config-router-bgp)#
```

15.5.5.5 bgp always-compare-med

The **bgp always-compare-med** command configures the switch to always consider Multi-Exit Discriminator (MED) values (also known as “metric”) in best-path selection. By default, this function is disabled, and MED values are compared only if two paths have the same neighbor AS.

When there are two or more links between autonomous systems, MED values may be set by a router in the originating AS to give preferences to certain routes. In comparing MED values, the lower value is preferred.

The **no bgp always-compare-med** and **default bgp always-compare-med** commands restore the default behavior of comparing MED values only on paths with the same neighbor AS.

Command Mode

Router-BGP Configuration

Command Syntax

bgp always-compare-med

no bgp always-compare-med

default bgp always-compare-med

Related Commands

- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**
- **bgp bestpath tie-break router-id**

Example

These commands configure BGP to always consider MED values in best-path comparisons.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# bgp always-compare-med
switch(config-router-bgp)#
```

15.5.5.6 bgp bestpath as-path ignore

The **bgp bestpath as-path ignore** command configures BGP to ignore the length of the Autonomous System (AS) path when comparing routes. This behavior is disabled by default. Normally, the switch compares AS paths as the third step in the best-path selection process (see **Best-Path Selection**), preferring the route with the shorter AS path.

The **no bgp bestpath as-path ignore** and **default bgp bestpath as-path ignore** commands restore the default behavior of considering AS path length in route comparisons.

Command Mode

Router-BGP Configuration

Command Syntax

bgp bestpath as-path ignore

no bgp bestpath as-path ignore

default bgp bestpath as-path ignore

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**
- **bgp bestpath tie-break router-id**

Example

These commands configure BGP to ignore AS path lengths when comparing routes.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# bgp bestpath as-path ignore
switch(config-router-bgp)#
```

15.5.5.7 bgp bestpath as-path multipath-relax

The **bgp bestpath as-path multipath-relax** command allows multiple eBGP routes to a destination to be considered equal in ECMP if their AS paths are the same length despite having different autonomous systems in those paths. The **no bgp bestpath as-path multipath-relax** command configures best-path selection to consider two paths *unequal* if their AS path contents are different, and prefers the first path received.

Multipath-relax is enabled by default. The **bgp bestpath as-path multipath-relax** and **default bgp bestpath as-path multipath-relax** commands restore the default behavior by removing the corresponding **no bgp bestpath as-path multipath-relax** command from ***running-config***.

For BGP to support equal cost multipath (ECMP) routing, the **maximum paths (BGP)** command must be issued in router-BGP configuration mode.

Command Mode

Router-BGP Configuration

Command Syntax

bgp bestpath as-path multipath-relax

no bgp bestpath as-path multipath-relax

default bgp bestpath as-path multipath-relax

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**

- **bgp bestpath tie-break router-id**

Example

These commands configure BGP best-path selection to consider routes unequal if the contents of their AS paths differ.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# no bgp bestpath as-path multipath-relax
switch(config-router-bgp)#
```

15.5.5.8 bgp bestpath ecmp-fast

By default, within an ECMP group the BGP best-path selection process prefers the active path (the first path received by the switch) unless a relevant tie-breaker is enabled. The **no bgp bestpath ecmp-fast** command causes the best-path selection process to ignore order of arrival and continue evaluating paths on other criteria.

The **bgp bestpath ecmp-fast** and **default bgp bestpath ecmp-fast** commands restore the default behavior by removing the corresponding **no bgp bestpath ecmp-fast** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

bgp bestpath ecmp-fast

no bgp bestpath ecmp-fast

default bgp bestpath ecmp-fast

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**
- **bgp bestpath tie-break router-id**

Example

These commands configure BGP to ignore order of arrival in best-path comparisons of paths within an ECMP group.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# no bgp bestpath ecmp-fast
switch(config-router-bgp)#
```

15.5.5.9 bgp bestpath med confed

By default, paths originating within the same confederation as the switch and received from confederation peers do not have their Multi-Exit Discriminator (MED) values compared as part of the best-path selection process. The **bgp bestpath med confed** command causes comparison of MED values in such routes. To ensure that MED values are considered in the best-path selection process for all routes received, use the **bgp always-compare-med** command.

The **no bgp bestpath med confed** and **default bgp bestpath med confed** commands restore the default behavior by removing the corresponding **bgp bestpath ecmp-fast** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

bgp bestpath med confed [missing-as-worst]

no bgp bestpath med confed [missing-as-worst]

default bgp bestpath med confed [missing-as-worst]

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**
- **bgp bestpath tie-break router-id**

Parameters

- **missing as worst** By default, best-path selection considers a missing MED value to be ***0***, so paths with missing MED values is preferred. This option reverses the behavior in comparisons of routes originating within the same confederation as the switch, treating a missing MED as having the highest (least preferred) value.



Note: The **bgp bestpath med missing-as-worst** command controls how best-path selection treats missing MED values for all routes received, and, if configured, overrides the **missing-as-worst** option of this command.

Example

These commands configure the BGP best-path selection process to consider MED values in comparisons between routes originating within the same confederation as the switch.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# bgp bestpath med confed
switch(config-router-bgp)#
```

15.5.5.10 bgp bestpath med missing-as-worst

By default, BGP best-path selection considers a missing MED value to be **0**, so paths with missing MED values will be preferred. The **bgp bestpath med missing-as-worst** command reverses the behavior, treating a missing MED as having the highest (least preferred) value.

The **no bgp bestpath med missing-as-worst** and **default bgp bestpath med missing-as-worst** commands restore the default behavior (giving preference to missing MED values) by removing the corresponding **bgp bestpath med missing-as-worst** command from ***running-config***.



Note: This command overrides the **missing-as-worst** setting of the **bgp bestpath med confed** command.

Command Mode

Router-BGP Configuration

Command Syntax

bgp bestpath med missing-as-worst

no bgp bestpath med missing-as-worst

default bgp bestpath med missing-as-worst

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath tie-break cluster-list-length**
- **bgp bestpath tie-break router-id**

Example

These commands configure the BGP best-path selection process to consider a missing MED value to be considered highest (least preferred) in MED comparisons for all routes received.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# bgp bestpath med missing-as-worst
switch(config-router-bgp)#
```

15.5.5.11 bgp bestpath tie-break cluster-list-length

The **bgp bestpath tie-break cluster-list-length** command causes the best-path selection process to prefer the multipath route with the shortest **CLUSTER_LIST** length in case of a tie in step ***10***. The cluster list length is assumed to be ***0*** if the route does not carry a **CLUSTER_LIST** attribute.

The **no bgp bestpath tie-break cluster-list-length** and **default bgp bestpath tie-break cluster-list-length** commands restore the default behavior by removing the associated **bgp bestpath tie-break cluster-list-length** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

bgp bestpath tie-break cluster-list-length

no bgp bestpath tie-break cluster-list-length

default bgp bestpath tie-break cluster-list-length

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break router-id**

Example

These commands configure the BGP selection process to prefer the multipath route with the shortest **CLUSTER_LIST** length in case of a tie.

```
switch(config)# router bgp 64500
```

```
switch(config-router-bgp)# bgp bestpath tie-break cluster-list-length
switch(config-router-bgp)#
```

15.5.5.12 bgp bestpath tie-break router-id

The **bgp bestpath tie-break router-id** command causes the best-path selection process to prefer the multipath route with the lowest **ROUTER_ID** in case of a tie in step 10. If the route is a reflected route (i.e., if it contains route reflector attributes), the process will use the **ORIGINATOR_ID** as the **ROUTER_ID** for comparison. This behavior is disabled by default.

The **no bgp bestpath tie-break router-id** and **default bgp bestpath tie-break router-id** commands restore the default behavior by removing the associated **bgp bestpath tie-break router-id** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

bgp bestpath tie-break router-id

no bgp bestpath tie-break router-id

default bgp bestpath tie-break router-id

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**

Example

These commands configure the best-path selection process to prefer the multipath route with the lowest **ROUTER_ID** in case of a tie.

```
switch(config)# router bgp 64500
switch(config-router-bgp)# bgp bestpath tie-break router-id
switch(config-router-bgp)#
```

15.5.5.13 bgp client-to-client reflection

By default, routes received from a route reflector client and selected as best routes are propagated to all BGP peers, including other route reflector clients. If the clients are fully meshed, however, routes received from a client do not need to be mirrored to other clients. In this case, client-to-client reflection should be disabled.

The **no bgp client-to-client reflection** command disables client-to-client reflection.

The **bgp client-to-client reflection** and **default bgp client-to-client reflection** commands restore the default behavior by removing the **no bgp client-to-client reflection** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

bgp client-to-client reflection

no bgp client-to-client reflection

default bgp client-to-client reflection

Example

These commands disable client-to-client reflection on the switch.

```
switch(config)# router bgp 1
switch(config-router-bgp)# no bgp client-to-client reflection
switch(config-router-bgp)#
```

15.5.5.14 bgp cluster-id

When using route reflectors, an AS is divided into clusters. A cluster consists of one or more route reflectors and a group of clients to which they re-advertise route information, and for redundancy a single cluster may contain multiple route reflectors. Each route reflector has a cluster ID. If the cluster has only one route reflector the cluster ID is its router ID, but if a cluster has multiple route reflectors a 4-byte cluster ID must be assigned to all route reflectors in the cluster. All must be configured with the same cluster ID to allow them to identify updates from the cluster's other route reflectors.

The **bgp cluster-id command** configures the cluster ID in a cluster with multiple route reflectors.

The **no bgp cluster-id** and **default bgp cluster-id** commands remove the cluster ID by removing the corresponding **bgp cluster-id** command from ***running-config***. Do not remove the cluster ID if there are multiple route reflectors in the cluster.

Command Mode

Router-BGP Configuration

Command Syntax

bgp cluster-id ID_NUM

no bgp cluster-id

default bgp cluster-id

Parameters

ID_NUM cluster ID shared by all route reflectors in the cluster (32-bit dotted-decimal notation). Options include:

- ***0.0.0.1*** to ***255.255.255.255*** valid cluster ID number.
- ***0.0.0.0*** removes the cluster-ID from the switch. Equivalent to **no bgp cluster-id** command.

Example

This command sets the cluster ID for the switch to ***172.22.30.101***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp cluster-id 172.22.30.101
switch(config-router-bgp)#
```

15.5.5.15 bgp confederation identifier

The **bgp confederation identifier** command configures the confederation identifier. Confederation can reduce the number of iBGP connections in a large AS domain. The AS domain is divided into several smaller sub-ASs, and each sub-AS remains fully connected. Devices in a sub-AS exchange information via iBGP, while devices in different sub-ASs use eBGP.

The **no bgp confederation identifier** and **default bgp confederation identifier** commands remove the **bgp confederation identifier command** from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

bgp confederation identifier ***as_number***

no bgp confederation identifier

default bgp confederation identifier

Parameters

as_number the ID of BGP AS confederation. Values range from ***1*** to ***4294967295***.

Example

This command sets the BGP confederation identifier to ***9***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp confederation identifier 9
switch(config-router-bgp)#
```

15.5.5.16 bgp confederation peers

The **bgp confederation peers** command configures a confederation consisting of sub-ASs.

Before this command is executed, the confederation ID should be configured using the **bgp confederation identifier** command. Otherwise this configuration is invalid. The configured ASs in this command are inside the confederation and each AS uses a fully meshed network. The confederation appears as a single AS to the devices outside it.

The **no bgp confederation peers** and **default bgp confederation peers** commands delete the specified sub-AS from the confederation by removing the corresponding **bgp confederation peers** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

bgp confederation peers ***as_range***

no bgp confederation peers ***as_range***

default bgp confederation peers ***as_range***

Parameters

as_range the sub-AS number. Formats include number (from ***1*** to ***4294967295***), number range, or comma-delimited list of numbers and ranges.

Example

This command configures the confederation that contains AS ***1000*** and AS ***1002***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp confederation peers 1000 1002
switch(config-router-bgp)#
```

15.5.5.17 bgp convergence slow-peer time

The **bgp convergence slow-peer time** command configures the idle peer time to wait for the slow peers to establish a session in a BGP convergence state.

The **no bgp convergence slow-peer** time command disables the inheritance of the configuration from the global BGP configuration mode. The **default bgp convergence slow-peer time** command sets the timeout value to the default value.

Command Mode

Router-BGP Configuration

Command Syntax

bgp convergence slow-peer time ***timeout***

no bgp convergence slow-peer time

default bgp convergence slow-peer time

Parameters

timeout the maximum time to wait for the slow peers to establish a session connection. Values range from ***1*** to ***3600*** seconds. The default value is ***90*** seconds.

Example

This command configures an idle peer timeout of ***40*** seconds to wait before establishing a session.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp convergence slow-peer time 40
switch(config-router-bgp)#
```

15.5.5.18 bgp convergence time

The **bgp convergence time** command configures the time to wait before the BGP convergence starts in a session.

The **no bgp convergence time** command removes the configured convergence time to wait. The **default bgp convergence time** command sets the timeout value to the default value.

Command Mode

Router-BGP Configuration

Command Syntax

bgp convergence time ***timeout_range***

no bgp convergence time

default bgp convergence time

Parameters

timeout_range the maximum time to wait for the BGP convergence. Values range from ***1*** to ***3600*** seconds. The default value is ***300*** seconds.

Example

This command configures a convergence time of ***200*** seconds to wait before establishing a session.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp convergence time 200
switch(config-router-bgp)#
```

15.5.5.19 bgp default

The **bgp default** command configures the default address family activation level of all addresses configured as BGP neighbors. The switch sends the following announcements to addresses active in an address family:

- **ipv4 address family**: IPv4 capability and all network advertisements with IPv4 prefixes.
- **ipv6 address family**: IPv6 capability and all network advertisements with IPv6 prefixes.

The following commands configure default address family activation levels for addresses configured as BGP neighbors:

- **bgp default ipv4-unicast**: all addresses are IPv4 address family active.
- **no bgp default ipv4-unicast**: all addresses are *not* IPv4 address family active.
- **bgp default ipv6-unicast**: all addresses are IPv6 address family active
- **no bgp default ipv6-unicast**: all addresses are *not* IPv6 address family active.
- **bgp default ipv4-unicast transport ipv6**: all BGP neighbor addresses are IPv4 address family active and IPv6 neighbors can receive IPv4 NLRIs.



Note: If it is necessary to exchange IPv4 NLRIs over an IPv6 connection, the IPv4 address family must be activated on the IPv6 neighbor. To do this for all IPv6 neighbors, use the command **bgp default ipv4-unicast transport ipv6**. For an individual neighbor, use the **neighbor activate** command for the IPv6 neighbor in the IPv4 address-family configuration mode as described below.

The activation state of an individual BGP neighbor address is configured by the **neighbor activate** command. The **neighbor activate** command overrides the address's default activation state for the address family configuration mode in which the command is issued:

- **neighbor activate**: the specified address is active.
- **no neighbor activate**: the specified address is *not* active.

The **default-default address family activation state** defines the address family activation level of all addresses configured as BGP neighbors when ***running-config*** does not contain any **bgp default** commands. The default state of the BGP default activation level varies by address family.

- **ipv4 address family**: all BGP addresses are IPv4 address family active.
- **ipv6 address family**: all BGP addresses are *not* IPv6 address family active.

The **default bgp default** command restores the default-default activation setting for BGP neighbor addresses in the specified address family:

- **default bgp ipv4-unicast** is equivalent to **bgp ipv4-unicast**.
- **default bgp ipv6-unicast** is equivalent to **no bgp ipv6-unicast**.

Command Mode

Router-BGP Configuration

Command Syntax

bgp default **ADDRESS_FAMILY**

no bgp default **ADDRESS_FAMILY**

default bgp default **ADDRESS_FAMILY**

Parameters

ADDRESS_FAMILY BGP address family. Options include:

- **ipv4-unicast** IPv4-unicast peering sessions.
- **ipv6-unicast** IPv6-unicast peering sessions.

Example

These commands configure the switch to configure all BGP neighbor addresses as IPv4 address-family active and IPv6 address-family active.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp default ipv4-unicast
switch(config-router-bgp)# bgp default ipv6-unicast
switch(config-router-bgp)# show active
router bgp 65533
   bgp log-neighbor-changes
   distance bgp 20 200 200
   neighbor 172.23.254.2 remote-as 65533
   neighbor 172.41.254.78 remote-as 65534
   neighbor 2001:0DB8:52a4:fe01::2 remote-as 65533
   neighbor 2001:0DB8:52a4:fe4c::1 out-delay 10
switch(config-router-bgp)#
```

The **show active** command does not display the **bgp default ipv4-unicast** command because it is the default setting for IPv4 peering sessions.

15.5.5.20 bgp enforce-first-as

The **bgp enforce-first-as** command causes a forced comparison of the first Autonomous System (AS) in the AS path of eBGP routes received from BGP neighbors to the configured remote external peer Autonomous System Number (ASN). Updates from eBGP peers that do not include that ASN as the first item in the AS path (in the **AS_PATH** attribute) are discarded.

This behavior is enabled by default upon BGP configuration, and disabled globally by the **no** form of this command. To configure first-AS enforcement for an individual neighbor or peer group, use the **neighbor enforce-first-as** command.

Command Mode

Router-BGP Configuration

Command Syntax

bgp enforce-first-as

default bgp enforce-first-as

no bgp enforce-first-as

Example

This command configures BGP to enforce the first AS globally.

```
switch(config-router-bgp)# bgp enforce-first-as
switch(config-router-bgp)#
```

15.5.5.21 bgp listen range

The **bgp listen range** command identifies the BGP peering request from a range of IPv4 or IPv6 address, and names the dynamic peer group to which those peers belong to. To create a static peer group, use the neighbor peer group (create) command.

The request can be from a single AS number or from a range of AS numbers configured. To accept the peering request from single ASN use the **remote-as** option, and to accept request from multiple ASNs use the **peer-filter** option.

Members of a dynamic peer group are configured in groups and not as individuals. Once a new peer group is created with a group name, the group name is then used as an argument by the following **neighbor** commands:

- **neighbor ebgp-multihop**
- **neighbor import-localpref**
- **neighbor maximum-routes**
- **neighbor route-map (BGP)**
- **neighbor timers**
- **neighbor update-source**

The **no bgp listen range** and **default bgp listen range** commands remove the dynamic peer group by deleting the corresponding command from ***running-config***. To remove a static peer group, use the **no neighbor** command. All peering relationships with group members are terminated when the dynamic peer group is deleted.

Command Mode

Router-BGP Configuration

Command Syntax

bgp listen range **NET_ADDRESS** [**PEER-ID** include ***router-id***] peer-group ***group_name*** [remote-as ***as_number*** | peer-filter ***filter_name***]

no bgp listen range **NET_ADDRESS** peer-group ***group_name***

default bgp listen range **NET_ADDRESS** peer-group ***group_name***

Parameters

- **NET_ADDRESS** IP address range. Options include:
  - ***IPv4_subnet*** IPv4 subnet (CIDR notation).
  - ***IPv4_address*** **mask** ***subnet*** IPv4 subnet (dotted decimal notation).
  - ***IPv6_prefix*** IPv6 subnet (dotted decimal notation).
- **PEER-ID** Additional specification for identifying a peer.
  - **include** Include following fields as part of peer identifier.
  - **router-id** Include router ID as part of peer identifier.
- ***group_name*** name of the peer group.
- ***as_number*** the autonomous system number, ranges from *1* to ***4294967295***.
- ***filter_name*** name of the peer filter.

Examples

- These commands create a dynamic peer group called ***brazil*** in AS ***5*** which accepts peering requests from the ***192.168.6.0/24*** subnet.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp listen range 192.168.6.0/24 peer-group brazil
 remote-as 5
switch(config-router-bgp)#
```

- These commands create a dynamic peer group called ***brazil*** in a range of AS numbers, which accepts peering requests from the ***192.0.2.0/24*** subnet. The range of AS numbers is defined by peer filter option.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp listen range 192.0.2.0/24 peer-group brazil
 peer-filter group-1
switch(config-router-bgp)#
```

- These commands enable the same address peering.

```
switch(config)# router bgp 1
```

```
switch(config-router-bgp)# bgp listen range 192.0.2.0/24 peer-id include
 router-id peer-group brazil peer-filter group-1
```

15.5.5.22 bgp log-neighbor-changes

The **bgp log-neighbor-changes** command configures the switch to generate a log message when a BGP peer enters or exits the **established** state. This is the default behavior.

The **no bgp log-neighbor-changes** command disables the generation of these log messages. The **default bgp log-neighbor-changes** command enables the generation of these log messages.

Command Mode

Router-BGP Configuration

Command Syntax

bgp log-neighbor-changes

no bgp log-neighbor-changes

default bgp log-neighbor-changes

Example

These commands configure the switch to generate a message when a BGP peer enters or exits the **established** state.

```
switch(config)# router bgp 1
switch(config-router-bgp)# bgp log-neighbor-changes
switch(config-router-bgp)#
```

15.5.5.23 bgp redistribute-internal (BGP)

The **bgp redistribute-internal** command enables the redistribution of iBGP routes into an Interior Gateway Protocol (IGP).

The **no bgp redistribute-internal** command disable route redistribution from the specified domain by removing the corresponding **bgp redistribute-internal** command from ***running-config***. The **default bgp redistribute-internal** command enables the redistribution of iBGP routes into an IGP.

Command Mode

Router-BGP Configuration Router-BGP Address-Family Configuration

Command Syntax

bgp redistribute internal

no bgp redistribute internal

default bgp redistribute internal

Example

This command redistributes internal BGP routes.

```
switch(config)# router bgp 9
switch(config-router-bgp)# bgp redistribute-internal
switch(config-router-bgp)#
```

15.5.5.24 bgp route install-map

The **bgp route install-map** command enables BGP Selective Route Download on the switch and allows the learning and advertising of the BGP routes without installing them in hardware.

The **no bgp route install-map** and **default bgp route install-map** commands delete the BGP Selective Route Download instance.

The **exit** command returns the switch to global configuration mode.

Command Mode

BGP Configuration

Command Syntax

bgp route install-map ***map_name***

Parameter

map_name The name of the route map configured.

Example

These commands configure BGP Selective Route Download for ***test_BGP*** map.

```
switch(config)# router bgp 100
switch(config-router-bgp)# bgp route install-map test_BGP
switch(config-router-bgp)#
```

15.5.5.25 bgp route-reflector preserve-attributes

The **bgp route-reflector preserve-attributes** command configures the switch, when operating as a BGP route reflector, to preserve the BGP attributes of re-advertised routes. By default, BGP attribute preservation is disabled. When attribute preservation is enabled, the BGP attributes (next-hop,local preference, and metric) are preserved in the reflected routes regardless of outbound BGP policies, except when those policies are part of an outbound route map. To override outbound route maps, use the **always** keyword.

The **no bgp route-reflector preserve-attributes** and **default bgp route-reflector preserve-attributes** commands disable BGP attribute preservation.

Command Mode

Router-BGP Configuration

Command Syntax

bgp route-reflector preserve-attributes [**always**]

no bgp route-reflector preserve-attributes

default bgp route-reflector preserve-attributes

Parameter

always Always preserves route attributes, overwriting route map changes.

Related Command

neighbor route-reflector-client

Example

The following commands configure the switch as a route reflector and the neighbor at ***10.5.2.1*** as one of its clients, then configure the switch to preserve the BGP attributes of reflected routes unless overridden by an outbound route map policy.

```
switch(config)# router bgp 10
switch(config-router-bgp)# neighbor 10.5.2.11 route-reflector-client
switch(config-router-bgp)# bgp route-reflector preserve-attributes
switch(config-router-bgp)#
```

15.5.5.26 clear bgp history

To clear all messages for a peer or group of peers, use the **`clear bgp history`** command .

Command Mode

Privileged EXEC

Command Syntax

`clear bgp` [**PEER** | **PREFIX** | peer-group **PEER_GROUP**] **`history`** [connect-failures] [vrf **VRF**]

Parameters

- **PEER** An IPv4 or IPv6 valid address.
- **PREFIX** An IPv4 or IPv6 valid prefix.
- **peer-group PEER_GROUP** A peer group name.
- **connect-failures** Optional and will not affect the result.
- **vrf VRF** A VRF name. If it's not supplied, command will act upon VRF default.

If no peer, prefix, or peer-group is supplied, the **`clear bgp history`** command will clear the history for all peers in the specified VRF.

Related Command

show bgp neighbors history

Example

This example clears the BGP Peer group ***Purple*** history from ***VRF_1***.

```
switch# clear bgp Purple history vrf VRF_1
```

15.5.5.27 clear ip bgp

The **`clear ip bgp`** command removes learned BGP routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required. This command can also clear the switch's BGP sessions with its peers.

Routes that are read or sent are processed through modified route maps or AS-path access lists.

Command Mode

Privileged EXEC

Command Syntax

`clear ip bgp` [**PEERS**] [**RESET_TYPE**] [**DATA_FLOW**] [**VRF_INSTANCE**]

Parameters

- **PEERS** specifies targeted BGP peers. Options include:
  - ***no parameters*** all IPv4 and IPv6 peers.
  - ***** all IPv4 and IPv6 peers.

- ***ipv4_addr*** the IPv4 peer with the specified IPv4 address.
- ***ipv6_addr*** the IPv6 peer with the specified IPv6 address.
- ***intrf_ipv6_addr*** the peer using the specified IPv6 link-local address.
- **peer-group *peer_grp_name*** the peers using the specified BGP peer group.

- **RESET_TYPE** specifies the method used to reset routes. Options include:
  - ***no parameters*** performs a hard reset that terminates current BGP sessions and recreates the local routing information base.
  - **soft** performs a soft reset that maintains current BGP sessions and reconfigures the local routing information base using stored routes.
- **DATA_FLOW** restricts soft reset to inbound or outbound routes. Hard reset is bidirectional. Options include:
  - ***no parameters*** resets inbound and outbound routes.
  - **in** resets inbound peer routes.
  - **out** resets outbound peer routes.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - ***no parameters*** resets matching peers in the context-active VRF.
  - ***vrf_name*** resets matching peers in the specified VRF.
  - all resets matching peers in all VRFs.
  - **default** resets matching peers in the default VRF.

Related Commands

- **`clear ip bgp counters`**
- **`clear ip bgp errors`**
- **`clear ip bgp neighbor *`**

Guidelines

Use the **`clear ip bgp`** command after changing any of the following BGP attributes:

- weights
- distribution lists
- timers
- administrative distance

Examples

- This command performs a hard reset of all IPv4 and IPv6 peers in the context-active VRF.

```
switch# clear ip bgp
! Peerings for all neighbors were hard reset switch#
```

- This command has the same behavior as the above **`clear ip bgp`** command.

```
switch# clear ip bgp *
! Peerings for all neighbors were hard reset switch#
```

15.5.5.28 clear ip bgp counters

The **`clear ip bgp counters`** command resets general statistics of peers. These statistics primarily consist of message-related counts.

Command Mode

Privileged EXEC

Command Syntax

clear ip bgp [**PEERS**] counters [**VRF_INSTANCES**]

Parameters

- **PEERS** specifies targeted BGP peers. Options include:
  - ***no parameters*** all IPv4 and IPv6 peers.
  - ***** all IPv4 and IPv6 peers.
  - ***ipv4_addr*** the IPv4 peer with the specified IPv4 address.
  - ***ipv6_addr*** the IPv6 peer with the specified IPv6 address.
  - ***intrf_ipv6_addr*** the peer using the specified IPv6 link-local address.
  - **peer-group *peer_grp_name*** the peers using the specified BGP peer group.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - ***no parameters*** resets matching peers in the context-active VRF.
  - ***vrf_name*** resets matching peers in the specified VRF.
  - **all** resets matching peers in all VRFs.
  - **default** resets matching peers in the default VRF.

Related Commands

- **clear ip bgp**
- **clear ip bgp errors**
- **clear ip bgp neighbor**

Example

This command resets general statistics of all IPv4 and IPv6 peers in the context-active VRF.

```
switch# clear ip bgp counters
! Counters for all neighbors were reset switch#
```

15.5.5.29 clear ip bgp errors

The **clear ip bgp errors** command resets the error statistics and history of peers. Peer general statistics primarily consist of notification errors, socket errors, and update errors.

Command Mode

Privileged EXEC

Command Syntax

clear ip bgp [**PEERS**] **errors** [**VRF_INSTANCES**]

Parameters

- **PEERS** specifies targeted BGP peers. Options include:
  - ***no parameters*** all IPv4 and IPv6 peers.
  - ***** all IPv4 and IPv6 peers.
  - ***ipv4_addr*** the IPv4 peer with the specified IPv4 address.
  - ***ipv6_addr*** the IPv6 peer with the specified IPv6 address.
  - ***intrf_ipv6_addr*** the peer using the specified IPv6 link-local address.
  - **peer-group *peer_grp_name*** the peers using the specified BGP peer group.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - ***no parameters*** resets matching peers in the context-active VRF.
  - ***vrf_name*** resets matching peers in the specified VRF.
  - **all** resets matching peers in all VRFs.
  - **default** resets matching peers in the default VRF.

Related Commands

- **clear ip bgp**
- **clear ip bgp counters**
- **clear ip bgp neighbor ***

Example

This command resets the error statistics of all IPv4 and IPv6 peers in the context-active VRF.

```
switch# clear ip bgp errors
! Errors for all neighbors were reset switch#
```

15.5.5.30 clear ip bgp neighbor

The **clear ip bgp neighbor** command clears BGP neighbors belonging to the IPv4 transport address family. To clear BGP neighbors in the IPv6 transport address family, use the **clear ipv6 bgp neighbor** command.

Command Mode

Privileged EXEC

Command Syntax

clear ip bgp neighbor [*****] [**vrf** ***vrf_name***] [**reason**

Parameters

***** optional; all neighbors in the address family are cleared with or without this option

vrf ***vrf_name*** specifies a VRF instance for which IPv4 transport address family BGP neighbors will be cleared. If no VRF is specified, the command clears IPv4 BGP neighbors in the context-active VRF.

vrf all clears IPv4 BGP neighbors in all VRFs.

vrf default clears IPv4 BGP neighbors in the default VRF.

reason ***message*** includes the specified message string in the notification sent to neighbors. Maximum string length 250 characters.

Related Commands

- **clear ip bgp**
- **clear ip bgp counters**
- **clear ip bgp errors**

Examples

- This command clears all IPv4 BGP neighbors in the context-active VRF.

```
switch# clear ip bgp neighbor
! Peerings for all ipv4 neighbors were hard reset switch#
```

- This command clears all IPv4 BGP neighbors in VRF ***purple***.

```
switch# clear ip bgp neighbor vrf purple
! Peerings for all ipv4 neighbors were hard reset switch#
```

15.5.5.31 clear ipv6 bgp

The **clear ipv6 bgp** command removes learned BGP routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required. This command can also clear the switch's BGP sessions with its peers.

Routes that are read or sent are processed through modified route maps or AS-path access lists.

Command Mode

Privileged EXEC

Command Syntax

`clear ipv6 bgp` [**PEERS**] [**RESET_TYPE**] [**DATA_FLOW**] [**VRF_INSTANCE**]

Parameters

- **PEERS** specifies targeted BGP peers. Options include:
  - ***no parameters*** all IPv4 and IPv6 peers.
  - ***** all IPv4 and IPv6 peers.
  - ***ipv4_addr*** the IPv4 peer with the specified IPv4 address.
  - ***ipv6_addr*** the IPv6 peer with the specified IPv6 address.
  - ***intrf_ipv6_addr*** the peer using the specified IPv6 link-local address.
  - **peer-group *peer_grp_name*** the peers using the specified BGP peer group.
- **RESET_TYPE** specifies the method used to reset routes. Options include:
  - ***no parameters*** performs a hard reset that terminates current BGP sessions and recreates the local routing information base.
  - **soft** performs a soft reset that maintains current BGP sessions and reconfigures the local routing information base using stored routes.
- **DATA_FLOW** restricts soft reset to inbound or outbound routes. Hard reset is bidirectional. Options include:
  - ***no parameters*** resets inbound and outbound routes.
  - **in** resets inbound peer routes.
  - **out** resets outbound peer routes.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - ***no parameters*** resets matching peers in the context-active VRF.
  - ***vrf_name*** resets matching peers in the specified VRF.
  - **all** resets matching peers in all VRFs.
  - **default** resets matching peers in the default VRF.

Related Commands

- **`clear ipv6 bgp counters`**
- **`clear ipv6 bgp errors`**
- **`clear ipv6 bgp neighbor *`**

Guidelines

Use the **`clear ipv6 bgp`** command after changing any of the following BGP attributes:

- weights
- distribution lists
- timers
- administrative distance

Examples

- This command performs a hard reset of all IPv4 and IPv6 peers in the context-active VRF.

```
switch# clear ipv6 bgp
! Peerings for all neighbors were hard reset switch#
```

- This command has the same behavior as the above **clear ip bgp** command.

```
switch# clear ipv6 bgp *
! Peerings for all neighbors were hard reset switch#
```

15.5.5.32 clear ipv6 bgp counters

The **clear ipv6 bgp counters** command resets general statistics of peers. These statistics primarily consist of message-related counts.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 bgp [**PEERS**] **counters** [**VRF_INSTANCES**]

Parameters

- **PEERS** specifies targeted BGP peers. Options include:
  - ***no parameters*** all IPv4 and IPv6 peers.
  - ***** all IPv4 and IPv6 peers.
  - ***ipv4_addr*** the IPv4 peer with the specified IPv4 address.
  - ***ipv6_addr*** the IPv6 peer with the specified IPv6 address.
  - ***intrf_ipv6_addr*** the peer using the specified IPv6 link-local address.
  - **peer-group *peer_grp_name*** the peers using the specified BGP peer group.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - ***no parameters*** resets matching peers in the context-active VRF.
  - ***vrf_name*** resets matching peers in the specified VRF.
  - **all** resets matching peers in all VRFs.
  - **default** resets matching peers in the default VRF.

Related Commands

- **clear ipv6 bgp**
- **clear ipv6 bgp errors**
- **clear ipv6 bgp neighbor ***

Example

This command resets general statistics of all IPv4 and IPv6 peers in the context-active VRF.

```
switch#clear ipv6 bgp counters
! Counters for all neighbors were reset switch#
```

15.5.5.33 clear ipv6 bgp errors

The **clear ipv6 bgp errors** command resets the error statistics and history of peers. Peer general statistics primarily consist of notification errors, socket errors, and update errors.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6s bgp [**PEERS**] **errors** [**VRF_INSTANCES**]

Parameters

- **PEERS** specifies targeted BGP peers. Options include:
  - ***no parameters*** all IPv4 and IPv6 peers.
  - ***** all IPv4 and IPv6 peers.
  - ***ipv4_addr*** the IPv4 peer with the specified IPv4 address.
  - ***ipv6_addr*** the IPv6 peer with the specified IPv6 address.
  - ***intrf_ipv6_addr*** the peer using the specified IPv6 link-local address.
  - **peer-group *peer_grp_name*** the peers using the specified BGP peer group.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - ***no parameters*** resets matching peers in the context-active VRF.
  - ***vrf_name*** resets matching peers in the specified VRF.
  - **all** resets matching peers in all VRFs.
  - **default** resets matching peers in the default VRF.

Related Commands

- **`clear ipv6 bgp`**
- **`clear ipv6 bgp counters`**
- **`clear ipv6 bgp neighbor *`**

Example

This command resets the error statistics of all IPv4 and IPv6 peers in the context-active VRF.

```
switch# clear ipv6 bgp errors
! Errors for all neighbors were reset switch#
```

15.5.5.34 clear ipv6 bgp neighbor

The **`clear ipv6 bgp neighbor`** command clears BGP neighbors belonging to the IPv6 transport address family. To clear BGP neighbors in the IPv4 transport address family, use the **`clear ip bgp neighbor`** command.

Command Mode

Privileged EXEC

Command Syntax

`clear ipv6 bgp neighbor` [*****] [**vrf *vrf_name***] [**reason *message***]

Parameters

***** optional; all neighbors in the address family are cleared with or without this option

vrf *vrf_name* specifies a VRF instance for which IPv6 transport address family BGP neighbors will be cleared. If no VRF is specified, the command clears IPv6 BGP neighbors in the context-active VRF.

vrf all clears IPv6 BGP neighbors in all VRFs.

vrf default clears IPv6 BGP neighbors in the default VRF.

reason *message* includes the specified message string in the notification sent to neighbors. Maximum string length 250 characters.

Related Commands

- **`clear ipv6 bgp`**
- **`clear ipv6 bgp counters`**
- **`clear ipv6 bgp errors`**

Examples

- This command clears all IPv6 BGP neighbors in the context-active VRF.

```
switch# clear ipv6 bgp neighbor
! Peerings for all ipv6 neighbors were hard reset switch#
```

- This command clears all IPv6 BGP neighbors in VRF ***purple*** and adds a message to the notification.

```
switch# clear ipv6 bgp neighbor vrf purple reason going down for maintenance
! Peerings for all ipv6 neighbors were hard reset switch#
```

15.5.5.35 distance bgp

The **`distance bgp`** command assigns an administrative distance to routes that the switch learns through BGP. Routers use administrative distances to select a route when two protocols provide routing information to the same destination. Distance values range from ***1*** to ***255***; lower distance values correspond to higher reliability. BGP routing tables do not include routes with a distance of ***255***.

The distance command assigns distance values to external, internal, and local BGP routes:

- **external**: Best-path routes learned from a neighbor external to the autonomous system. Default distance is ***200***.
- **internal**: Internal routes are routes learned from a BGP entity within the same autonomous system. Default distance is ***200***.
- **local**: Local routes are networks listed with a network router configuration command for that router or for networks that are redistributed from another process. Default distance is ***200***.

The **`no distance bgp`** and **`default distance bgp`** commands restore the default administrative distances by removing the **`distance bgp`** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

`distance bgp` ***external_dist*** [**INTERNAL_LOCAL**]

`no distance bgp`

`default distance bgp`

Parameters

- ***external_dist*** distance assigned to external routes. Values range from ***1*** to ***255***.
- **INTERNAL_LOCAL** distance assigned to internal and local routes. Values for both routes range from ***1*** to ***255***. Options include:
  - ***no parameter*** the ***external_dist*** value is also assigned to internal and local routes.
  - ***internal_dist local_dist*** values assigned to internal and local routes.

Example

- This command assigns an administrative distance of ***150*** to external routes, ***200*** to internal, and ***150*** to local routes.

```
switch(config)# router bgp 1
switch(config-router-bgp)# distance bgp 150 200 150
switch(config-router-bgp)#
```

15.5.5.36 dynamic peer max

The **`dynamic peer max`** command limits the number of dynamic BGP peers allowed on the switch.

The **no dynamic peer max** and **default dynamic peer max** commands restore the default limit of dynamic BGP peers by removing the **dynamic peer max** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

dynamic peer max ***maximum***

no dynamic peer max

default dynamic peer max

Parameters

maximum the maximum number of dynamic BGP peers to be allowed on the switch. Values range from ***1*** to ***1000***; default value is ***100***.

Example

This command sets the maximum number of dynamic BGP peers allowed on the switch to ***200***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# dynamic peer max 200
switch(config-router-bgp)#
```

15.5.5.37 graceful-restart stalepath-time

The **graceful-restart stalepath-time** command specifies the maximum time that stale routes from a restarting BGP neighbor will be retained after a BGP session is re-established with that peer.

The **no graceful-restart stalepath-time** and **default graceful-restart stalepath-time** commands restore the default value of ***300*** seconds by deleting the **graceful-restart stalepath-time** statement from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

graceful-restart stalepath-time ***interval***

no graceful-restart stalepath-time

default graceful-restart stalepath-time

Parameters

interval Maximum period (in seconds) that stale routes from a restarting BGP neighbor will be retained after the BGP session is re-established. Values range from ***1*** to ***3600*** (***60*** minutes). Default is ***300***.

Example

These commands configure the stale path retention interval to ***15*** minutes.

```
switch(config)# router bgp 1
switch(config-router-bgp)# graceful-restart stalepath-time 900
switch(config-router-bgp)#
```

15.5.5.38 graceful-restart-helper

The **graceful-restart helper** command enables BGP graceful restart helper mode on the switch for all BGP neighbors. When graceful restart helper mode is enabled, the switch will retain routes from neighbors

which are capable of graceful restart while those neighbors are restarting BGP. Graceful restart helper is enabled by default. To configure graceful restart helper mode for a specific neighbor or peer group, use the **neighbor graceful-restart-helper** command. Individual neighbor configuration takes precedence over the global configuration.

The **no graceful-restart helper** command disables graceful restart helper mode on the switch. The **default graceful-restart helper** command enables graceful restart helper mode by removing the corresponding **no graceful-restart helper** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

graceful-restart helper long-lived

no graceful-restart helper long-lived

default graceful-restart helper long-lived

Parameter

long-lived Enables long lived graceful restart helper mode.

Example

These commands disable graceful restart helper mode on the switch.

```
switch(config)# router bgp 1
switch(config-router-bgp)# no graceful-restart-helper
switch(config-router-bgp)#
```

15.5.5.39 ip as-path access-list

The **ip as-path access-list** command creates an access list to filter BGP route updates. If access list ***list_name*** does not exist, this command creates it. If it already exists, this command appends statements to the list.

The **no ip as-path access-list** and **default ip as-path access-list** commands delete the named access list.

Command Mode

Global Configuration

Command Syntax

ip as-path access-list ***list_name*** **FILTER_TYPE** *regex* **ORIGIN**

no ip as-path access-list ***list_name***

default ip as-path access-list ***list_name***

Parameters

- ***list_name*** the name of the AS path access list.
- **FILTER_TYPE** access resolution of the specified AS path. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- **regex** a regular expression describing the AS path being filtered. Regular expressions are pattern-matching strings that are composed of text characters and operators.
- **ORIGIN** the origin of the path information. Values include:
  - ***no parameter*** sets the origin to **any**.

- **any** any BGP origin.
- **egp** EGP origin.
- **igp** IGP origin.
- **incomplete** incomplete origin.

Example

These commands create an AS path access list named ***list1*** which allows all BGP routes except those originating in ***AS 3***.

```
switch(config)# ip as-path access-list list1 deny _3$
switch(config)# ip as-path access-list list1 permit .*
switch(config)#
```

15.5.5.40 ip as-path regex-mode

The **`ip as-path regex-mode`** command specifies how the switch will evaluate regular expressions describing AS paths in ACLs. When the regex mode is set to **asn**, AS numbers in the ACL are interpreted as AS numbers; only complete AS number matches in the AS path return a match. When it is set to **string**, AS numbers in the ACL are interpreted as strings; both complete AS number matches and longer AS numbers that include the target string return a match. The default mode is **asn**.

For example, **asn** mode returns as **`false`** and the **string** mode returns as **`true`** when searching for "***10*** in an AS path of **100 200**.

The **`no ip as-path regex-mode`** and **`default ip as-path regex-mode`** commands restore the regex mode to **asn** by removing the **`ip as-path regex-mode`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip as-path regex-mode` **MODE_SETTING**

`no ip as-path regex-mode`

`default ip as-path regex-mode`

Parameters

MODE_SETTING Specifies how regular expressions describing AS paths in AS path ACLs will be evaluated. Options include:

- **asn** AS numbers in the ACL are interpreted as AS numbers; only complete AS number matches in the AS path return a match.
- **string** AS numbers in the ACL are interpreted as strings; both complete AS number matches and longer AS numbers that include the target string return a match.

Example

This command sets the regex mode to **string**.

```
switch(config)# ip as-path regex-mode string
switch(config)#
```

15.5.5.41 ip community-list

The **`ip community-list`** command creates and configures a BGP access list based on BGP communities.

The **`no ip community-list`** and **`default ip community-list`** commands delete the specified community list by removing the corresponding **`ip community-list`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip community-list` ***list_name*** [permit | deny] [**GSHUT** | ***aa:nn*** | internet | local-as | no-advertise | no-export | *number*]

`no ip community-list` ***list_name***

`default ip community-list` ***list_name***

Parameters

- ***list_name*** name of the community list. Valid input is text.
- **permit** permits access to the specified community.
- **deny** denies access to the specified community.

  Note: The **deny** statements are ignored for all set community/extcommunity/large-community operations.
- **GSHUT** well-known graceful shutdown community.
- ***aa:nn*** AA is ***65535*** and NN specifies the community number (***0-65535***) within the AS.
- **internet** advertises route to the Internet community.
- **local-as** advertises route only to local peers.
- **no-advertise** does not advertise route to any peer.
- **no-export** advertises route only within BGP AS boundary.
- **number** community number. Values ranges from 0 to 4294967040.

Related Commands

- **`route-map`**
- **`match (route-map)`**
- **`show ip community-list`**
- **`show ip extcommunity-list`**

Guideline

EOS does not support disabling the process of graceful shutdown community.

Note: The **`ip community-list`** command with the **permit internet** option permits access to all routes associated with any community.

Examples

- This command creates a BGP community list (named ***list_9***) that does not match members of route maps configured with AS-network number ***100:250***.

```
switch(config)# ip community-list list_9 deny 100:250
switch(config)#
```

- These commands create a BGP community list that permits the graceful shutdown community, then use that list it in a route map to permit routes with that community.

```
switch(config)# ip community-list gshut_list permit GSHUT
switch(config)# route-map map1
switch(config-route-map-map1)# match community gshut_list
switch(config-route-map-map1)# exit
switch(config)# show route-map map1
route-map map1 permit 10
  Description:
  Match clauses:
    match community gshut_list
  SubRouteMap:
```

```
  Set clauses:
switch(config)#
```

- This command permits access to all routes associated with the BGP community list (***CLIST1***).

```
switch(config)# ip community-list CLIST1 permit internet
switch(config)#
```

15.5.5.42 ip community-list regexp

The **ip community-list regexp** command creates and configures a BGP access list based on BGP communities. A BGP community access list filters prefixes based on their BGP communities. The command uses regular expressions to identify the communities specified by the list. To create a community list by explicitly specifying one or more communities, use the **ip community-list** command.

The **no ip community-list regexp** and **default ip community-list regexp** commands delete the specified community list. To delete a specific community-list entry, specify the entry in the **no ip community-list regexp** command.

Command Mode

Global Configuration

Command Syntax

ip community-list regexp ***list_name*** {deny | permit} ***reg_exp***

no ip community-list regexp ***list_name*** {deny | permit} ***reg_exp***

default ip community-list regexp ***list_name***

Parameters

- ***list_name*** name of the community list. Valid input is text.
- **permit** access is permitted for the specified community.
- **deny** access is denied for the specified community.

  Note: The **deny** statements are ignored for all set community/extcommunity/large-community operations.
- ***reg_exp*** list of communities, formatted as a regular expression. Regular expressions are pattern-matching strings that are composed of text characters and operators.

  Note: When using the **no** form of the command, a regular expression can be used to specify a single entry to be removed from the list, leaving the rest of the list intact. If no entry is specified, the **no** form of the command removes the entire list.

Related Commands

- **route-map**
- **match (route-map)**
- **show ip community-list**
- **show ip extcommunity-list**

Guideline

The **ip community-list regexp** command with the **permit internet** option permits access to only those routes that carry the community value of ***0***.

Examples

- This command creates a BGP community list that permits routes from networks ***20-24*** and ***30-34*** in autonomous system ***10***.

```
switch(config)# ip community-list regexp list_2 permit 10:[2-3][0-4]_
```

```
switch(config)#
```

- This command removes the above statement from the community list named ***list_2***, leaving any other statements in the list intact.

```
switch(config)# no ip community-list regexp list_2 permit 10:[2-3][0-4]_
switch(config)#
```

- This command deletes the community list named ***list_2*** entirely.

```
switch(config)# no ip community-list regexp list_2
switch(config)#
```

- This command permits access to all routes associated with the BGP community list (***CLIST1***) that carry the community value ***0***.

```
switch(config)# ip community-list regexp CLIST1 permit internet
switch(config)#
```

15.5.5.43 ip extcommunity-list

The **`ip extcommunity-list`** command creates an extended community list to filter VRF routes or for Link BandWidth (LBW) advertisement.

The following extcommunity-list types are supported:

- **Route Target (RT)** identifies sites that may receive appropriately tagged routes.
- **Site of Origin (SoO)** identifies sites where the switch learned the route.
- **Link Bandwidth (LBW)** advertises BGP link bandwidth.

The **`no ip extcommunity-list`** and **`default ip extcommunity-list`** commands delete the specified extended community list by removing the corresponding **`ip extcommunity-list`** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip extcommunity-list` ***list_name*** {deny | permit} **COMM_1** [**COMM_2...COMM_n**]

`no ip extcommunity-list` ***list_name***

`default ip extcommunity-list` ***list_name***

Parameters

- ***list_name*** name of the extended community list.
- **deny** access is denied for the specified community.
- **permit** access is permitted for the specified community.
- **COMM_x** extended community attribute. Options include:
  - **rt** ***aa:nn*** route target, as specified by autonomous system:network number.
  - **rt** ***ip_addr:nn*** route target, as specified by ip address:network number.
  - **soo** ***aa:nn*** Site of Origin, as specified by autonomous system:network number.
  - **soo** ***ip_addr:nn*** site of origin, as specified by ip address:network number.
  - **lbw** link bandwidth in bits per second.

Related Commands

- **`route-map`**
- **`match (route-map)`**

- **show ip community-list**
- **show ip extcommunity-list**

Example

This command creates a BGP extended community list that denies routes from route target ***100:250***.

```
switch(config)# ip extcommunity-list list_9 deny rt 100:250
switch(config)#
```

15.5.5.44 ip extcommunity-list regexp

The **ip extcommunity-list regexp** command creates an extended community list to filter VRF routes or for link bandwidth (LBW) advertisement. The command uses regular expressions to define the extended communities specified by the list. To specify particular values, use the **ip extcommunity-list** command.

The following extcommunity-list types are supported:

- **Route Target (RT)** identifies sites that may receive appropriately tagged routes.
- **Site of Origin (SoO)** identifies sites where the switch learned the route.
- **Link Bandwidth (LBW)** advertises BGP link bandwidth.

The **no ip extcommunity-list regexp** and **default ip extcommunity-list regexp** commands delete the specified extended community list by removing the corresponding **ip extcommunity-list regexp** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip extcommunity-list regexp ***list_name*** {deny | permit} ***reg_exp***

no ip extcommunity-list regexp ***list_name*** {deny | permit} ***reg_exp***

default ip extcommunity-list regexp ***list_name***

Parameters

- ***list_name*** name of the extended community list. Valid input is text.
- **deny** access is denied for the specified extended community list.

  Note: The **deny** statements are ignored for all set community/extcommunity/large-community operations.
- **permit** access is permitted for the specified extended community list.
- ***reg_exp*** list of communities, formatted as a regular expression. Regular expressions are pattern-matching strings that are composed of text characters and operators.
  - Expressions beginning with **RT:** match the **route target** extended community attribute option.
  - Expressions beginning with **SoO:** match the **site of origin** extended community attribute option.

Related Commands

- **route-map**
- **match (route-map)**
- **show ip community-list**
- **show ip extcommunity-list**

Example

This command creates a BGP extended community list that denies routes from route target networks ***20-24*** and ***30-34*** in autonomous system ***10***.

```
switch(config)# ip extcommunity-list regexp list_1 deny RT:10:[2-3][0-4]_
switch(config)#
```

15.5.5.45 ip large-community-list regexp

The **ip large-community-list regexp** command creates and configures a BGP access list based on BGP large communities. A BGP large-community access list filters prefixes based on their BGP large community values. The command uses regular expressions to match large communities. Multiple large-community lists with the same name may be specified. To create a large-community list by explicitly specifying one or more communities, use the **ip large-community-list** command.

Large-communities are represented as follows: [ASN]:local-part1:local-part2.

The **no ip large-community-list regexp** and **default ip large-community-list regexp** commands delete the specified large community list. To delete a specific community-list entry, specify the entry in the **no ip large-community-list regexp** command.

Command Mode

Global Configuration

Command Syntax

ip large-community-list regexp ***list_name*** {deny | permit} ***reg_exp***

no ip large-community-list regexp ***list_name*** {deny | permit} ***reg_exp***

default ip large-community-list regexp ***list_name***

Parameters

- ***list_name*** name of the community list. Valid input is text.
- **deny** access is denied for the specified community.

  Note: The **deny** statements are ignored for all set community/extcommunity/large-community operations.
- **permit** access is permitted for the specified community.
- ***reg_exp*** list of communities, formatted as a regular expression. Regular expressions are pattern-matching strings that are composed of text characters and operators.

  Note: When using the **no** form of the command, a regular expression can be used to specify a single entry to be removed from the list, leaving the rest of the list intact. If no entry is specified, the **no** form of the command removes the entire list.

Related Commands

- **route-map**
- **match (route-map)**
- **show ip community-list**
- **show ip extcommunity-list**

Examples

- This command creates a BGP large community list that permits routes from autonomous system ***10*** with local-part1 value of ***20-24*** or ***30-34***.

```
switch(config)# ip large-community-list regexp list_2 permit 10:[2-3][0-4]:_
switch(config)#
```

- This command removes the above statement from the large community list named ***list_2***, leaving any other statements in the list intact.

```
switch(config)# no ip large-community-list regexp list_2 permit 10:[2-3]:
[0-4]_
switch(config)#
```

- This command deletes the large community list named ***list_2*** entirely.

```
switch(config)# no ip large-community-list regexp list_2
switch(config)#
```

15.5.5.46 match as-range

The **match as-range** command defines the match statement for the peer-filter, based on the match statement the peer-filter accept or reject the incoming peer request. The match statement includes a sequence number, AS number range and a match condition to accept or reject a peer by comparing its remote AS number to the specified range. A peer filter can consist of a single match statement or multiple match statements. The match statement for the peer filter is configured under peer-filter configuration mode.

The **no match as-range** or **default match as-range** command deletes the peer-filter condition for the group from ***running-config***.

Command Mode

Peer-Filter Configuration

Command Syntax

[***sequence_number***] **match as-range** [***as_number1***] [***as_number2***] result {accept | reject} ***group_name***

no match as-range [***as_number1***] [***as_number2***] result {accept | reject} ***group_name***

default match as-range [***as_number1***] [***as_number2***] result {accept | reject} ***group_name***

Parameters

- ***sequence_number*** optional sequence number for the match statement; one is automatically created if not assigned. Values range from ***0*** to ***65535***.
- ***group_name*** name of the peer filter group.
- ***as_number*** the autonomous system number, values range from ***1*** to ***4294967295***.

Examples

- These commands define a peer filter that accepts any AS number.

```
switch(config)# peer-filter group1
switch(config-peer-filter-group1)# 10 match as-range 1-4294967295 result
 accept
switch(config-peer-filter-group1)#
```

- These commands define a peer filter that accepts any AS number within ***65000*** and ***65100*** (inclusive) except ***65008*** and ***65009***.

```
switch(config)# peer-filter group2
switch(config-peer-filter-group2)# 10 match as-range 65008-65009 result
 reject
switch(config-peer-filter-group2)# 20 match as-range 65000-651000 result
 accept
switch(config-peer-filter-group2)#
```

- These commands define a peer filter that accepts three specific remote AS numbers.

```
switch(config)# peer-filter group3
eswitch(config-peer-filter-group3)# 10 match as-range 65003 result accept
switch(config-peer-filter-group3)# 20 match as-range 65007 result accept
switch(config-peer-filter-group3)# 30 match as-range 65009 result accept
switch(config-peer-filter-group3)#
```

15.5.5.47 maximum-paths (BGP)

The **maximum-paths** command controls the maximum number of parallel BGP routes that the switch supports. The default maximum is one route. The command provides an Equal Cost Multiple Paths (ECMP) parameter that controls the number of equal-cost paths that the switch stores in the routing table for each route.

For paths to be consider equal, they must have the same weight, local preference, AS-path length, and origin. To require that they also have the same Mmulti-Exit Discriminator (MED) value, use the **bgp always-compare-med** command. To require that their AS paths have the same contents, use the **no bgp bestpath as-path multipath-relax** command.

The **no maximum-paths** and **default maximum-paths** commands restore the default values of the maximum number of parallel routes and the maximum number of ECMP paths by removing the corresponding **maximum paths** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

maximum-paths ***paths*** [ecmp ***ecmp_paths***]

no maximum-paths

default maximum-paths

Parameters

- ***paths*** maximum number of parallel routes. Default value is ***1***. Value must be less than or equal to the maximum number of ECMP paths.
- ***ecmp_paths*** maximum number of ECMP paths for each route. Default is maximum value as defined belows.

  Value for each parameter ranges from ***1*** to the number of interfaces available per ECMP group, which is platform dependent.

  - **Arad**: Values range from ***1*** to ***128***. Default value is ***128***.
  - **FM6000**: Values range from ***1*** to ***32***. Default value is ***32***.
  - **PetraA**: Values range from ***1*** to ***16***. Default value is ***16***.
  - **Trident**: Values range from ***1*** to ***32***. Default value is ***32***.
  - **Trident II**: Values range from ***1*** to ***128***. Default value is ***128***.

Examples

- These commands configure the maximum number of BGP parallel paths to ***12*** without changing the ECMP value.

```
switch(config)# router bgp 1
switch(config-router-bgp)# maximum-paths 12
switch(config-router-bgp)#
```

- These commands configure the maximum number of BGP parallel routes to ***2***, with a maximum of ***4*** ECMP paths for each route.

```
switch(config)# router bgp 1
switch(config-router-bgp)# maximum-paths 2 ecmp 4
switch(config-router-bgp)#
```

15.5.5.48 neighbor

Use the **neighbor** command to enable large communities on a 'per-neighbor' or 'per-peer group' basis. This behavior is consistent with all other forms of communities supported by EOS.

Receiving and processing of large communities is enabled by default.

Command Mode

BGP router mode

Command Syntax

neighbor [***A.B.C.D.*** [send-community [large]]| ***A:B:C:D:E:F:G:H*** | *NAME* | default | ***fe80::A:B:C:D% interface*** | interface]

Parameters

- ***A.B.C.D.*** Neighbor IPv4 address
  - **send-community** Enable sending communities.
    - **large** Send large community attribute to this neighbor.
- ***A:B:C:D:E:F:G:H*** Neighbor IPv6 address.
- ***NAME*** Name of the peer-group.
- **default** Apply to all neighbors.
- ***fe80::A:B:C:D% interface*** Neighbor IPv6 link-local address.
- **interface** Interface range to be used for BGP session establishment.

Example

You can enable large communities on a 'per-neighbor' or 'per-peer group' basis.

```
switch(config)# router bgp 1
switch(config-bgp-router)# neighbor 1.1.1.1 send-community large
```

15.5.5.49 neighbor activate

The **neighbor activate** command defines the configuration mode address family activation state of a specified address that is configured as a BGP neighbor. The switch sends the following announcements to addresses active in an address family:

- **IPv4 address family**: IPv4 capability and all network advertisements with IPv4 prefixes.
- **IPv6 address family**: IPv6 capability and all network advertisements with IPv6 prefixes.

The **bgp default** command configures the default address family activation state of addresses configured as BGP neighbors. The **neighbor activate** and **no neighbor activate** commands override the neighbor's default activation state within the ***address family*** configuration mode.

- **neighbor activate**: the specified address is active in the address family.

no neighbor activate: the specified address is not active in the address family.

The **default neighbor activate** command removes the corresponding **neighbor activate** or **no neighbor activate** command from ***running-config***, restoring the default address family activation state for the specified neighbor address.

Command Mode

Router-BGP Address-Family Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`activate`**

`no neighbor` ***neighbor_ID*** **`activate`**

`default neighbor` ***neighbor_ID*** **`activate`**

Parameters

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Limitations

The switch supports the advertisement of networks with IPv6 prefixes to IPv4 transport neighbors. The switch does *not* support the advertisement of networks with IPv4 prefixes to IPv6 transport neighbors.

Example

These commands activate the advertising of specified neighbors during IPv4 peering sessions, then display the result.

```
switch(config)# router bgp 1
switch(config-router-bgp)# no address-family ipv4
switch(config-router-bgp-af)# neighbor 172.41.18.15 activate
switch(config-router-bgp-af)# neighbor 172.49.22.6 activate
switch(config-router-bgp-af)# no neighbor 172.15.21.18 activate
switch(config-router-bgp-af)# show active
   address-family ipv4
      no neighbor 172.15.21.18 activate
      neighbor 172.49.22.6 activate
      neighbor 172.41.18.15 activate
switch(config-router-bgp-af)# exit
switch(config-router-bgp)#
```

15.5.5.50 neighbor allowas-in

By default, BGP drops received routes if their Autonomous System (AS) paths contain the AS Number (ASN) of the switch. The **`neighbor allowas-in`** command configures the switch to accept routes from the specified BGP neighbor even if their AS paths contain the ASN of the switch itself. Optionally, the command can also configure the maximum number of times that the switch's ASN can appear in a route before it is dropped.

The **`no neighbor allowas-in`** command configures the default behavior (dropping BGP routes that contain the ASN of the switch).

The default **`neighbor allowas-in`** command applies the system default configuration for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **`no neighbor`** command removes all configuration commands for the BGP neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`allowas-in`** [***asn_quantity***]

`no neighbor` ***neighbor_ID*** **`allowas-in`**

`default neighbor` ***neighbor_ID*** **`allowas-in`**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***asn_quantity*** number of repetitions of the switch's ASN allowed in the AS path of routes received from the specified BGP neighbor. Values range from ***1*** to ***10***. Default is ***3***.

Related Commands

This command is used on a customer edge router that is part of a split AS; to address the problem at the provider end, use the **`neighbor as-path remote-as replace out`** command.

Example

These commands configure the switch to accept routes from the BGP neighbor at ***192.168.1.30*** which contain the switch's ASN in their AS paths as many as three times.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.168.1.30 allowas-in
switch(config-router-bgp)#
```

15.5.5.51 neighbor as-path remote-as replace out

By default, BGP drops received routes if their Autonomous System (AS) paths contain the AS Number (ASN) of the switch. In a split AS sharing route advertisements through a provider network, this can result in valid routes being dropped. The **`neighbor as-path remote-as replace out`** command configures a provider edge switch to replace the customer's AS with its own in route advertisements sent to neighbors in that AS.

The **`no neighbor as-path remote-as replace out`** command configures the default behavior (leaving the customer's AS in the AS path attribute of routes advertised to the specified neighbor).

The **`default neighbor as-path remote-as replace out`** command applies the system default configuration for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **`no neighbor`** command removes all configuration commands for the BGP neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`as-path remote-as replace out`**

`no neighbor` ***neighbor_ID*** **`as-path remote-as replace out`**

`default neighbor` ***neighbor_ID*** **`as-path remote-as replace out`**

Parameters

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Related Commands

This command is used on a provider edge router forwarding BGP routes to a customer in a split AS; to address the problem at the customer end, use the **`neighbor allowas-in`** command.

Example

These commands configure the switch to substitute its local ASN for the ASN of the BGP neighbor at ***192.168.2.15*** in BGP routes advertised to that neighbor.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.168.2.15 as-path remote-as replace out
```

```
switch(config-router-bgp)#
```

15.5.5.52 neighbor auto-local-addr

The **neighbor auto-local-addr** command configures the switch to automatically determine the local address to be used for the non-transport address family in NLRIs sent to the specified neighbor or peer group. This allows IPv4 NLRIs to be carried over IPv6 transport, or IPv6 NLRIs to be carried over IPv4 transport.

The **no neighbor auto-local-addr** command applies the system default configuration.

The **default neighbor auto-local-addr** command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Note: While this feature works well in eBGP deployments in which the pairing routers are directly connected and have matching IP address configurations, multi-hop eBGP or iBGP deployments may require manual local address configuration.

To explicitly configure a local address for the non-transport address family for a specific neighbor or peer group, use the **neighbor local-v4-addr** command for IPv6 neighbors, or the **neighbor local-v6-addr** for IPv4 neighbors.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **auto-local-addr**

no neighbor ***neighbor_ID*** **auto-local-addr**

default neighbor ***neighbor_ID*** **auto-local-addr**

Parameters

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Example

For the IPv6 neighbor at ***2001:0DB8:c2a4:1761::2***, these commands configure the switch to automatically determine the IPv4 NLRI value to be sent during peering sessions.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 2001:0DB8:c2a4:1761::2 auto-local-addr
switch(config-router-bgp)#
```

15.5.5.53 neighbor default-originate

The **neighbor default-originate** command advertises a default route to a BGP neighbor or peer group. This default route overrides the default route advertised by any other means to the specified neighbor or peer group. However, the update generated by **neighbor default-originate** is not processed by neighbor route map out policies.

If a route map is specified in this command, its set clauses are used to modify attributes of the exported default route, but its match clauses are not used to conditionally advertise the route. The default route is always advertised to the specified neighbor.

The **no neighbor default-originate** command applies the system default configuration.

The **default neighbor default-originate** command applies the system default configuration for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration Router-BGP Address-Family Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`default-originate`** [**MAP**]

`no neighbor` ***neighbor_ID*** **`default-originate`**

`default neighbor` ***neighbor_ID*** **`default-originate`**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- **MAP** specifies route map that modifies attributes of the exported default route. Options include:
  - ***no parameter*** attributes are not modified by a route map.
  - **route-map** ***map_name*** attributes set by specified route map are assigned to the exported default route.

Example

These commands advertise a default route to the BGP neighbor at ***192.168.14.5***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 192.168.14.5 default-originate
switch(config-router-bgp)#
```

15.5.5.54 neighbor description

The **`neighbor description`** command associates descriptive text with the specified peer or peer group.

The **`no neighbor description`** command removes the text association from the specified peer or peer group.

The **`default neighbor description`** command removes the text association from the specified peer for individual neighbors, and applies the peer group's description to neighbors that are members of a peer group.

The **`no neighbor`** command removes all configuration commands for the neighbor at the specified address or for the specified peer group.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`description`** ***description_string***

`no neighbor` ***neighbor_ID*** **`description`**

`default neighbor` ***neighbor_ID*** **`description`**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***description_string*** text string to be associated with the neighbor or peer group.

Example

These commands associate the string ***PEER_1*** with the peer located at ***192.168.1.30***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.168.1.30 description PEER_1
switch(config-router-bgp)#
```

15.5.5.55 neighbor ebgp-multihop

The **neighbor ebgp-multihop** command programs the switch to accept and attempt BGP connections to the external peers residing on networks not directly connected to the switch. The command does not establish the multihop if the only route to the peer is the default route (***0.0.0.0***).

The **no neighbor ebgp-multihop** command applies the system default configuration.

The **default neighbor ebgp-multihop** command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **ebgp-multihop** [***hop_number***]

no neighbor ***neighbor_ID*** **ebgp-multihop**

default neighbor ***neighbor_ID*** **ebgp-multihop**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***hop_number*** time-to-live (hops). Values range from ***1*** to ***255***. Default value is ***255***.

Example

These commands configure the switch to accept and attempt BGP connections to the external peer located at ***192.168.1.30***, setting the hop limit to ***32***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.168.1.30 ebgp-multihop 32
switch(config-router-bgp)#
```

15.5.5.56 neighbor enforce-first-as

The **neighbor enforce-first-as** command causes a forced comparison of the first Autonomous System (AS) in the AS path of eBGP routes received from a specified BGP peer or peer group to the configured remote external peer Autonomous System Number (ASN). Updates from the specified eBGP peers that do not include an ASN as first AS path (in the **AS_PATH** attribute) are discarded.

This behavior is enabled globally by default upon BGP configuration, and disabled for the specified neighbor or peer group by the **no** form of the command. To configure first AS enforcement globally, use the **bgp enforce-first-as** command.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **enforce-first-as**

no neighbor ***neighbor_ID*** **enforce-first-as**

default neighbor ***neighbor_ID*** **enforce-first-as**

Parameters

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Example

This command disables enforcement of the first BGP AS for the neighbors in peer group ***region-3***.

```
switch(config-router-bgp)# no neighbor region-3 enforce-first-as
switch(config-router-bgp)#
```

15.5.5.57 neighbor export-localpref

The **neighbor export-localpref** command determines the **LOCAL_PREF** value that is sent in BGP UPDATE packets to the specified peer or peer group. This command has no effect on external peers.

The **no neighbor export-localpref** command resets the **LOCAL_PREF** value to the system default of ***100*** in packets sent to the specified peer or peer group.

The **default neighbor export-localpref** command resets the **LOCAL_PREF** value to the system default of ***100*** for individual neighbors, and applies the peer groups's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or the specified peer group.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **export-localpref** ***preference***

no neighbor ***neighbor_ID*** **export-localpref**

default neighbor ***neighbor_ID*** **export-localpref**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***preference*** preference value. Values range from ***0*** to ***4294967295***.

Example

This command configures the switch to fill the **LOCAL_PREF** field with ***200*** in UPDATE packets that it sends to the peer located at ***10.1.1.45***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 10.1.1.45 export-localpref 200
switch(config-router-bgp)#
```

15.5.5.58 neighbor graceful-restart

The **neighbor graceful-restart** command enables the BGP graceful restart mode for a specified BGP neighbor or peer group. When graceful restart mode is enabled, the switch retains routes from neighbors that are capable of graceful restart. By default, graceful restart is disabled for all BGP neighbors. Individual neighbor configuration takes precedence over the global configuration.

The **no neighbor graceful-restart** and **default neighbor graceful-restart** commands disable graceful restart mode for the specified BGP neighbor or peer group by removing the corresponding **no neighbor graceful-restart** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **graceful-restart**

no neighbor ***neighbor_ID*** **graceful-restart**

default neighbor ***neighbor_ID*** **graceful-restart**

Parameter

neighbor_ID neighbors's IPv4 or IPv6 address or peer group name.

Example

This command enables BGP graceful restart mode for the neighbor with the IP address ***192.168.12.1***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.168.12.1 graceful-restart
switch(config-router-bgp)#
```

15.5.5.59 neighbor graceful-restart-helper

The **neighbor graceful-restart helper** command enables BGP graceful restart helper mode for the specified BGP neighbor or peer group. When graceful restart helper mode is enabled, the switch will retain routes from neighbors which are capable of graceful restart while those neighbors are restarting BGP. The neighbor graceful-restart-helper is enabled by default for all BGP neighbors. To configure graceful restart helper mode for all BGP neighbors, use the **graceful-restart-helper** command. Individual neighbor configuration takes precedence over the global configuration.

The **no neighbor graceful-restart helper** command disables graceful restart helper mode for the specified BGP neighbor or peer group. The **default neighbor graceful-restart helper** command enables graceful restart helper mode for the specified BGP neighbor or peer group by removing the corresponding **no neighbor graceful-restart helper** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **graceful-restart helper long-lived**

no neighbor ***neighbor_ID*** **graceful-restart helper long-lived**

default neighbor ***neighbor_ID*** **graceful-restart helper long-lived**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- **long-lived** Enables long lived graceful restart helper mode.

Example

These commands disable graceful restart helper mode for the neighbor at ***192.168.12.1***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# no neighbor 192.168.12.1 graceful-restart-helper
switch(config-router-bgp)#
```

15.5.5.60 neighbor import-localpref

The **neighbor import-localpref** command determines the local preference assigned to routes received from the specified external peer or peer group. This command has no effect on routes received from internal peers.**no neighbor import-localpref**

The command resets the local preference to the default of ***100*** for routes received from the specified peer or peer group.

The **default neighbor import-localpref** command resets the local preference to the default of ***100*** for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **import-localpref** ***preference***

no neighbor ***neighbor_ID*** **import-localpref**

default neighbor ***neighbor_ID*** **import-localpref**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***preference*** preference value. Values range from ***0*** to ***4294967295***.

Example

These commands configure the switch to assign a local preference of ***50*** to routes received from the peer located at ***192.168.1.30***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 192.168.1.30 import-localpref 50
switch(config-router-bgp)#
```

15.5.5.61 neighbor local-as

The **neighbor local-as** command changes the local AS value sent to the specified peer in OPEN messages, allowing the switch to appear as a member of a different AS to the selected peer. Arista switches replace the local AS number with the modified value rather than prepending it to routes, so we implement the command only as **neighbor local-as no-prepend replace-as**.

Note: To establish a BGP connection with a static peer, the peer must also be configured to expect the specified ASN. This is done by using the **neighbor remote-as** command on the peer switch.

The **no neighbor local-as** command disables this modification for the specified peer or peer group. The **default neighbor local-as** command disables this modification for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **local-as** ***as_id*** **no-prepend replace-as**

no neighbor ***neighbor_ID*** **local-as**

default neighbor ***neighbor_ID*** **local-as**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***as_id*** AS number that is sent in OPEN messages to the specified peer in place of the actual AS of the switch. Values range from ***1*** to ***4294967295***.

  This parameter cannot be set to the switch's AS number or to any AS number in the peer's network.

Examples

These commands configure the switch to replace its local ASN in OPEN messages sent to the peer at ***10.13.64.1*** with ASN ***64500***, and configure the peer to expect that ASN in messages received from the switch.

Switch Configuration

```
switch(config)# router bgp 64497
switch(config-router-bgp)# neighbor 10.13.64.1 local-as 64500 no-prepend
switch(config-router-bgp)#
```

Peer Configuration

```
peer(config)# router bgp 64502
peer(config-router-bgp)# neighbor 10.4.3.10 remote-as 64500
peer(config-router-bgp)#
```

15.5.5.62 neighbor local-v4-addr

The **`neighbor local-v4-addr`** command specifies the next-hop value that the switch sends as the IPv4 NLRI value to neighbors with whom IPv6 transport peering is established.

The **`no neighbor local-v4-addr`** command applies the system default configuration.

The **`default neighbor local-v4-addr`** command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

To configure the switch to automatically determine the IPv4 address to be sent as the next-hop in IPv4 NLRIs to an IPv6 neighbor, use the **`neighbor auto-local-addr`** command.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`local-v4-addr`** ***ipv4_local***

`no neighbor` ***neighbor_ID*** **`local-v4-addr`**

`default neighbor` ***neighbor_ID*** **`local-v4-addr`**

Parameters

- ***neighbor_ID*** neighbor's IPv6 address or peer group name.
- ***ipv4_local*** next hop address.

Example

For the neighbor at ***2001:0DB8:c2a4:1761::2***, these commands specify an IPv4 NLRI value of ***10.7.5.11*** to be sent during IPv6 transport peering sessions.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 2001:0DB8:c2a4:1761::2 local-v4-addr
 10.7.5.11
switch(config-router-bgp)#
```

15.5.5.63 neighbor local-v6-addr

The **`neighbor local-v6-addr`** command specifies the next-hop value that the switch sends as the IPv6 NLRI value to neighbors with which IPv4 transport peering is established.

In IPv6 peering sessions, the switch sends the global IPv6 address of the interface that is used to transmit BGP updates.

The **no neighbor local-v6-addr** command applies the system default configuration.

The **default neighbor local-v6-addr** command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **local-v6-addr** ***ipv6_local***

no neighbor ***neighbor_ID*** **local-v6-addr**

default neighbor ***neighbor_ID*** **local-v6-addr**

Parameters

- ***neighbor_ID*** neighbor's IPv4 address or peer group name.
- ***ipv6_local*** next hop address (A:B:C:D:E:F:G:H).

Example

For the neighbor at ***10.7.5.11***, these commands specify an IPv6 NLRI value that is sent during IPv4 transport peering sessions.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 10.7.5.11 local-v6-addr 2001:0DB8:c2a
4:1761::2
switch(config-router-bgp)# show active
router bgp 1
   bgp log-neighbor-changes
   bgp default ipv6-unicast
   neighbor 10.7.5.11 local-v6-addr 2001:0DB8:c2a4:1761::2
switch(config-router-bgp)#
```

15.5.5.64 neighbor maximum-routes

The **neighbor maximum-routes** command determines the number of BGP routes the switch accepts from a specified neighbor and defines an action when the limit is exceeded. The default value is ***12000***. To remove the maximum routes limit, select a limit of zero.

When the number of routes received from a peer exceeds the limit, the switch generates an error message. This command can also configure the switch to disable peering with the neighbor. In this case, the neighbor state is reset only through a **clear ip bgp** command.

The **no neighbor maximum-routes** command applies the system default maximum-routes value of ***12000*** for the specified peer.

The **default neighbor maximum-routes** command applies the system default value for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **maximum-routes** ***quantity*** [**ACTION**]

no neighbor ***neighbor_ID*** **maximum-routes**

default neighbor ***neighbor_ID*** **maximum-routes**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***quantity*** maximum number of routes. Values include:
  - ***0*** the switch does not define a route limit.
  - ***1 to 4294967294*** maximum number of routes.
- **ACTION** switch action when the route limit is exceeded. Values include:
  - ***no parameter*** peering is disabled and an error message is generated.
  - **warning-only** peering is not disabled, but an error message is generated.

Example

This command configures the switch to accept ***15000*** routes for the neighbor at ***10.3.16.210***. If the neighbor exceeds ***15000*** routes, the switch disables peering with the neighbor.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 110.3.16.210 maximum-routes 15000
switch(config-router-bgp)#
```

15.5.5.65 neighbor next-hop-peer

The **neighbor next-hop-peer** command configures the switch to list the peer address as the next hop in routes that it receives from the specified peer BGP-speaking neighbor or members of the specified peer group. This command overrides the next hop for all routes received from this neighbor or peer group.

The **no neighbor next-hop-peer** command applies the system default (no next-hop override) for the specified peer.

The **default neighbor next-hop-peer** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or the specified peer group.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor *neighbor_ID* next-hop-peer

no neighbor *neighbor_ID* next-hop-peer

default neighbor *neighbor_ID* next-hop-peer

Parameters

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Example

This command configures the peer address of ***10.3.2.24*** as the next hop for routes advertised to the switch from the peer BGP neighbor.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.3.2.24 next-hop-peer
switch(config-router-bgp)#
```

15.5.5.66 neighbor next-hop-self

The **neighbor next-hop-self** command configures the switch to list its address as the next hop in routes that it advertises to the specified BGP-speaking neighbor or neighbors in the specified peer group. This is used in networks where BGP neighbors do not directly access all other neighbors on the same subnet.

The **no neighbor next-hop-self** command applies the system default (no next-hop override) for the specified peer.

The **default neighbor next-hop-self** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or for the specified peer group.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **next-hop-self**

no neighbor ***neighbor_ID*** **next-hop-self**

default neighbor ***neighbor_ID*** **next-hop-self**

Parameters

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Example

This command configures the switch as the next hop for the peer at ***10.4.1.30***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 10.4.1.30 next-hop-self
switch(config-router-bgp)#
```

15.5.5.67 neighbor next-hop resolution v4-mapped-v6 translation

The **neighbor next-hop resolution v4-mapped-v6 translation** command configures the switch to enable translation of IPv4-mapped IPv6 addresses to IPv4 addresses. With this setting enabled, when the switch receives an IPv4-mapped IPv6 address for a next hop, it will translate it to an IPv4 address. This allows the next hop to be resolved in an IPv4 network.

The **no neighbor next-hop resolution v4-mapped-v6 translation** and **default neighbor next-hop resolution v4-mapped-v6 translation** commands disable the translation from IPv4-mapped IPv6 addresses to IPv4 addresses.

Command Mode

BGP IPv6 Labeled-Unicast Address Family Configuration

Command Syntax

neighbor {***neighbor_ID***} **next-hop resolution v4-mapped-v6 translation**

no neighbor {***neighbor_ID***} **next-hop resolution v4-mapped-v6 translation**

default neighbor {***neighbor_ID***} **next-hop resolution v4-mapped-v6 translation**

Parameters

- ***neighbor_ID*** a neighboring peer or peer group that may send IPv4-mapped IPv6 addresses to this switch.

Guidelines

- This command is active only if the multi-agent routing protocol model is running.
- This command requires an IPv6 labeled-unicast address family.
- This command applies to the default VRF.

Example

These commands enter BGP IPv6 Labeled-Unicast Address Family Configuration mode for AS ***64510*** (creating the BGP instance if it does not exist) and enable the translation of IPv4-mapped IPv6 addresses to IPv4 addresses for neighbors in the ***v6_pg*** peer group.

```
switch(config)# router bgp 64510
switch(config-router-bgp)# address-family ipv6 labeled-unicast
switch(config-router-bgp-af-label)# neighbor v6_pg next-hop resolution v4-
mapped-v6 translation
switch(config-router-bgp-af-label)#
```

15.5.5.68 neighbor out-delay

The **`neighbor out-delay`** command sets the period of time that a route update for the specified neighbor must be in the routing table before the switch exports it to BGP. The out delay interval is used for bundling routing updates.

The **`no neighbor out-delay`** command applies the system default (out-delay value of zero) for the specified peer.

The **`default neighbor out-delay`** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **`no neighbor`** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`out-delay`** ***delay_time***

`no neighbor` ***neighbor_ID*** **`out-delay`** ***delay_time***

`default neighbor` ***neighbor_ID*** **`out-delay`** ***delay_time***

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***delay_time*** the out delay period (seconds). Values range from ***0*** to ***600***. Default value is ***0***.

Example

These commands set the out delay period to ***5*** seconds for the connection with the peer at ***10.24.15.9***.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 10.24.15.9 out-delay 5
switch(config-router-bgp)#
```

15.5.5.69 neighbor passive

The **`neighbor passive`** command sets the TCP connection for the specified BGP neighbor or peer group to passive mode. When the peer's transport connection mode is set to passive, it accepts TCP connections for BGP but does not initiate them.

The **no neighbor passive** command sets the specified BGP neighbor or peer group to active connection mode. BGP peers in active mode can both accept and initiate TCP connections for BGP. This is the default behavior.

The **default neighbor passive** command restores the default connection mode. The default mode is **active** for individual BGP peers, or the mode inherited from the peer group for peer group members.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor *neighbor_ID* passive

no neighbor *neighbor_ID* passive

default neighbor *neighbor_ID* passive

Parameter

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Example

These commands configure the neighbor at IP address ***10.2.2.14*** to not initiate TCP connections for BGP peering.

```
switch(config)# router bgp 300
switch(config-router-bgp)# neighbor 10.2.2.14 passive
switch(config-router-bgp)#
```

15.5.5.70 neighbor password

The **neighbor password** command enables authentication on a TCP connection with a BGP peer. The plain-text version of the password is a string, up to ***8*** bytes in length. Peers must use the same password to ensure proper communication.

The ***running-config*** displays the encrypted version of the password. The encryption scheme is not strong by cryptographic standards; encrypted passwords should be treated in the same manner as plain-text passwords.

The **no neighbor password** command applies the system default for the specified peer, removing the neighbor password from the configuration and disabling authentication with the specified peer.

The **default neighbor password** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor password** and **default neighbor password** commands remove the neighbor password from the configuration, disabling authentication with the specified peer.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor *neighbor_ID* password [**ENCRYPT_LEVEL**] ***key_text***

no neighbor *neighbor_ID* password

default neighbor *neighbor_ID* password

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.

- **ENCRYPT_LEVEL** the encryption level of the ***key_text*** parameter. Values include:
  - ***no parameter*** the ***key_text*** is in clear text.
  - ***0*** the ***key_text*** is in clear text. Equivalent to the ***no parameter*** case.
  - ***7*** the ***key_text*** is MD5-encrypted.
- ***key_text*** the password.

Example

This command specifies a password in clear text.

```
switch(config)# router bgp 1
switch(config-router-bgp)# neighbor 10.25.25.13 password 0 code123
switch(config-router-bgp)#
```

Running-config stores the password as an encrypted string.

15.5.5.71 neighbor peer group (create)

Peer groups allow the user to apply settings to a group of BGP neighbors simultaneously. Once a peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Settings applied to an individual neighbor in the peer group override group settings.

The **neighbor peer group (create)** command is used to create static BGP peer groups. Static peer groups are peer groups whose members are added manually. To assign BGP neighbors to a static peer group, use the **neighbor peer group (neighbor assignment)** command. To create a dynamic peer group, use the **bgp listen range** command.

The **no neighbor peer group (create)** and **default neighbor peer group (create)** commands remove the specified static peer group from ***running-config***. When a static peer group is deleted, the neighbors that were members of that peer group lose any configuration that was inherited from the peer group. The **no** form of the **bgp listen range** command removes a dynamic peer group.

The **no neighbor** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***group_name*** **peer group**

no neighbor ***group_name*** **peer group**

default neighbor ***group_name*** **peer group**

Parameters

group_name peer group name.

Examples

- These commands create a BGP peer group called ***bgpgroup1***, assign several neighbors to the group, apply a route map, and adjust the configuration for one group member.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor bgpgroup1 peer group
switch(config-router-bgp)# neighbor 10.1.1.1 peer group bgpgroup1
switch(config-router-bgp)# neighbor 10.2.2.2 peer group bgpgroup1
switch(config-router-bgp)# neighbor 10.3.3.3 peer group bgpgroup1
switch(config-router-bgp)# neighbor bgpgroup1 route-map corporate in
switch(config-router-bgp)# neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)# show active
```

```
router bgp 9
bgp log-neighbor-changes
   neighbor bgpgroup1 peer group
   neighbor bgpgroup1 route-map corporate in
   neighbor bgpgroup1 maximum-routes 12000
   neighbor 10.1.1.1 peer group bgpgroup1
   neighbor 10.2.2.2 peer group bgpgroup1
   neighbor 10.3.3.3 peer group bgpgroup1
   neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)#
```

- This command removes peer group ***bgpgroup1*** from ***running-config***. The group members remain, but all settings that group members inherited from the peer group are removed.

```
switch(config-router-bgp)# no neighbor bgpgroup1 peer group
switch(config-router-bgp)# show active
router bgp 9
bgp log-neighbor-changes
   neighbor 10.1.1.1 maximum-routes 12000
   neighbor 10.2.2.2 maximum-routes 12000
   neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)#
```

15.5.5.72 neighbor peer group (neighbor assignment)

Peer groups allow the user to apply settings to a group of BGP neighbors simultaneously. Once a peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Settings applied to an individual neighbor in the peer group override group settings.

The **`neighbor peer group (neighbor assignment)`** command is used to assign BGP neighbors to an existing static peer group. To create a static peer group, use the **`neighbor peer group (create)`** command. A neighbor can only belong to one peer group, so issuing this command for a neighbor that is already a member of another group will remove it from that group.

The **`no neighbor peer group`** and **`default neighbor peer group`** commands remove the specified neighbor from all peer groups. When a neighbor is removed from a peer group, the neighbor retains the configuration inherited from the peer group.

The **`no neighbor`** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` **NEIGHBOR_ADDR** **`peer group`** ***group_name***

`no neighbor` **NEIGHBOR_ADDR** **`peer group`**

`default neighbor` **NEIGHBOR_ADDR** **`peer group`**

Parameters

- **NEIGHBOR_ADDR** address of a neighbor being added to peer group. Values include:
  - ***ipv4_addr*** neighbor's IPv4 address.
  - ***ipv6_addr*** neighbor's IPv6 address.
- ***group_name*** peer group name.

Examples

- These commands create a BGP peer group called ***bgpgroup1***, assign several neighbors to the group, and apply a route map.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor bgpgroup1 peer group
switch(config-router-bgp)# neighbor 10.1.1.1 peer group bgpgroup1
switch(config-router-bgp)# neighbor 10.2.2.2 peer group bgpgroup1
switch(config-router-bgp)# neighbor 10.3.3.3 peer group bgpgroup1
switch(config-router-bgp)# neighbor bgpgroup1 route-map corporate in
switch(config-router-bgp)#
```

- This command removes the neighbor at ***1.1.1.1*** from the peer group. All settings that neighbor ***10.1.1.1*** inherited from the peer group are maintained.

```
switch(config-router-bgp)# no neighbor 10.1.1.1 peer group
switch(config-router-bgp)#
```

15.5.5.73 neighbor remote-as

The **`neighbor remote-as`** command configures the expected AS Number for a neighbor (peer). This configuration is required to establish a static peer connection. Internal neighbors have the same AS Number (ASN); external neighbors have different ASNs.

Note: To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (e.g., in an IPv6-only environment), configure an appropriate router ID using the **`router-id (BGP)`** command.

When a static peer is using the **neighbor local-as** command to replace its local ASN with a configured ASN in OPEN messages, use the **`neighbor remote-as`** command to configure the switch to expect the configured ASN for that peer.

The **`no neighbor remote-as`** command applies the system default for the specified peer or peer group.

The **`default neighbor remote-as`** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **`no neighbor`** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`remote-as`** ***as_id***

`no neighbor` ***neighbor_ID*** **`remote-as`**

`default neighbor` ***neighbor_ID*** **`remote-as`**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***as_id*** Autonomous System (AS) of the peer. Values range from ***1*** to ***4294967295***.

Example

These commands establish an eBGP connection with the router at ***10.4.3.10*** in ***AS 64500***.

```
switch(config)# router bgp 64497
switch(config-router-bgp)# neighbor 10.4.3.10 remote-as 64500
switch(config-router-bgp)#
```

15.5.5.74 neighbor remove-private-as

The **neighbor remove-private-as** command removes private autonomous system numbers from outbound routing updates for external BGP (eBGP) neighbors. When the Autonomous System (AS) path includes only private autonomous system numbers, the **REMOVAL** parameter specifies how the private autonomous system number is removed.

The **no neighbor remove-private-as** command applies the system default (preserves private AS numbers) for the specified peer.

The **default neighbor remove-private-as** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **remove-private-as** [**REMOVAL**]

no neighbor ***neighbor_ID*** **remove-private-as**

default neighbor ***neighbor_ID*** **remove-private-as**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- **REMOVAL** specifies removal of all private AS numbers when the AS path contains only private AS numbers. Values include:
  - **all** removes all private AS numbers from AS path in outbound updates.
  - **all replace-as** all private AS numbers in AS path are replaced with router's local AS number.



Note: This command does not support a mix of public and private AS numbers.

Examples

- These commands program the switch to remove all private AS numbers from outbound routing updates for the eBGP neighbor at ***10.5.2.11*** only if the AS path does not contain any public AS number.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.11 remove-private-as
switch(config-router-bgp)#
```

- This command replaces all private AS numbers in the AS path with the switch's local AS number.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.11 remove-private-as all replace-
as
switch(config-router-bgp)#
```

15.5.5.75 neighbor rib-in pre-policy retain

By default, inbound BGP routes that are filtered out by the inbound policy are still stored on the switch. Because all routes are retained, this allows policies to be changed without the need to reset the BGP sessions. All routes received by the switch (including those that were filtered out by the inbound policy) can be seen by issuing the **show ip bgp neighbor received-routes** command.

The **no neighbor rib-in pre-policy retain** command configures the switch to discard those routes received from the specified neighbor (or peer group) that are filtered out by the inbound policy.

The **neighbor rib-in pre-policy retain** command restores the system default behavior (retaining routes from the specified neighbor or group regardless of inbound policy).

The **default neighbor rib-in pre-policy retain** command applies the system default (retaining policy-rejected routes) for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **rib-in pre-policy retain** [**all**]

no neighbor ***neighbor_ID*** **rib-in pre-policy retain**

default neighbor ***neighbor_ID*** **rib-in pre-policy retain**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- **all** the command applies to all routes from the specified neighbor or peer group, including those that would otherwise be discarded as invalid (because their AS-Path contains the switch's own ASN, for example). Without this keyword, the command applies only to routes that were filtered out by the inbound policy.

Examples

These commands configure the switch to discard routes received from from the neighbor at ***10.5.2.23*** which are filtered out by the switch's inbound policies.

```
switch(config)# router bgp 9
switch(config-router-bgp)# no neighbor 10.5.2.23 rib-in pre-policy retain
switch(config-router-bgp)#
```

These commands configure the switch to retain all routes received from the neighbor at ***10.5.2.23*** (including invalid routes).

```
switch(config)# router bgp 9
switch(config-router-bgp)# no neighbor 10.5.2.23 rib-in pre-policy retain all
switch(config-router-bgp)#
```

15.5.5.76 neighbor route-map (BGP)

The **neighbor route-map** command applies a route map to inbound or outbound BGP routes. When a route map is applied to outbound routes, the switch will advertise only routes matching at least one section of the route map. Only one outbound route map and one inbound route map can be applied to a given neighbor. A new route map applied to a neighbor will replace the previous route map.

The command is available in the ***router-bgp*** and the ***router-bgp-address-family*** configuration modes. The mode in which the command is executed determines the scope of the command:

- In the ***router-bgp*** mode, the route map is applied to the specified neighbor in all peering sessions where it is advertised.
- In the ***router-bgp-address-family*** mode, the route map is applied to the neighbors only in peering sessions corresponding to the configuration-mode address family.

The **no neighbor route-map** command discontinues the application of the specified route map for the specified neighbor and direction. Removing a route map from one direction does not remove it from the other if it has been applied to both.

The **default neighbor route-map** command applies the system default (no route map) for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Router-BGP Address-Family Configuration

Command Syntax

`neighbor` *neighbor_ID* `route-map` *map_name* DIRECTION

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***map_name*** name of a route map.
- **DIRECTION** routes to which the route map is applied. Options include:
  - **in** route map is applied to inbound routes.
  - **out** route map is applied to outbound routes.

Example

This command applies a route map named ***inner-map*** to a BGP inbound route from ***10.5.2.11***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.11 route-map inner-map in
switch(config-router-bgp)#
```

15.5.5.77 neighbor route-reflector-client

Participating BGP routers within an AS communicate eBGP-learned routes to all of their peers, but to prevent routing loops they must not re-advertise iBGP-learned routes within the AS. To ensure that all members of the AS share the same routing information, a fully meshed network topology (in which each member router of the AS is connected to every other member) can be used, but this topology can result in high volumes of iBGP messages when it is scaled. Instead, in larger networks one or more routers can be configured as route reflectors.

A route reflector is configured to re-advertise routes learned through iBGP to a group of BGP neighbors within the AS (its clients), eliminating the need for a fully meshed topology.

The **`neighbor route-reflector-client`** command configures the switch to act as a route reflector and configures the specified neighbor as one of its clients. Additional clients are specified by re-issuing the command.

The **`no neighbor route-reflector-client`** and **`default neighbor route-reflector-client`** commands disable route reflection by deleting the **`neighbor route-reflector-client`** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` *neighbor_ID* `route-reflector-client`

`no neighbor` *neighbor_ID* `route-reflector-client`

`default neighbor` *neighbor_ID* `route-reflector-client`

Parameters

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Related Commands

- **`bgp client-to-client reflection`**

- **bgp route-reflector preserve-attributes**

Example

This command configures the switch as a route reflector and the neighbor at ***10.5.2.1*** as one of its clients.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.11 route-reflector-client
switch(config-router-bgp)#
```

15.5.5.78 neighbor route-to-peer

The **`neighbor route-to-peer`** command allows BGP to establish a connection to reach the specified peer using kernel routing table information. By default, route-to-peer configuration is enabled for a peer or a peer group.

The **`no neighbor route-to-peer`** command prevents BGP from using kernel routing table information to establish a BGP connection to reach a peer and the **`default neighbor route-to-peer`** command enables route-to-peer configuration for a peer or a peer group by removing the corresponding **`no neighbor route-to-peer`** command from the ***running-config***.

If the peer is directly connected, BGP instead uses ARP table or neighbor table information to establish a BGP connection to reach the peer.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`route-to-peer`**

`no neighbor` ***neighbor_ID*** **`route-to-peer`**

`default neighbor` ***neighbor_ID*** **`route-to-peer`**

Parameter

neighbor_ID neighbor's IPv4 or IPv6 address or the peer group name.

Example

These commands establish a connection between the switch and the BGP peer located at IP address ***172.16.1.1***, and prevent BGP from using kernel routing table information to establish a route to that peer.

```
switch(config)# router bgp 64496
switch(config-router-bgp)# no neighbor 172.16.1.1 route-to-peer
switch(config-router-bgp)# neighbor 172.16.1.1 remote-as 100
switch(config-router-bgp)#
```

15.5.5.79 neighbor send-community

The **`neighbor send-community`** command configures the switch to include community path attributes for routes in the UPDATE messages advertised to the specified BGP neighbor. By default, the command enables the switch to send all community attributes: standard, extended, and large. To advertise *only a subset* of community attributes, use the keyword(s) for the community attribute(s) to be included. To add additional community attributes in a separate command, or to remove specific community attributes from advertised routes, use the **`neighbor send-community add / remove`** command.

Note: The **`neighbor send-community link-bandwidth`** command will override this command and vice-versa.

The **`no neighbor send-community`** command applies the system default (not sending community attributes in BGP UPDATE messages) for the specified peer.

The **default neighbor send-community** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor *neighbor_ID* send-community [extended] [large][standard]

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- **extended** includes extended community attributes.
- **large** includes large community attributes.
- **standard** includes standard community attributes.

Examples

- These commands configure the switch to send all community attributes to the neighbor at address ***10.5.2.23***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.23 send-community
switch(config-router-bgp)#
```

- These commands configure the switch to include only large community attributes in the routes sent to the neighbor at address ***10.5.2.24***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.24 send-community large
switch(config-router-bgp)#
```

- These commands configure the switch to send only standard and large community attributes to the neighbor at address ***10.5.2.25***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.25 send-community standard large
switch(config-router-bgp)#
```

15.5.5.80 neighbor send-community add / remove

The **neighbor send-community add / remove** command modifies the types of community path attributes included for routes in the UPDATE messages advertised to the specified BGP neighbor without having to issue the **no neighbor send-community** command.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor *neighbor_ID* send-community {add | remove}{extended | large | standard}

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- **add** appends the specified community path attribute type to the list of community path attribute types sent to the specified neighbor.
- **remove** removes the specified community path attribute type from the list of community path attribute types sent to the specified neighbor.

- **extended** enables (or disables) sending of the extended community path attribute to the specified neighbor.
- **large** enables (or disables) sending of the large community path attribute to the specified neighbor.
- **standard** enables (or disables) sending of the standard community path attribute to the specified neighbor.
- **link-bandwidth** see **neighbor send-community link-bandwidth** for a description of this parameter.

Guidelines

- If the **neighbor send-community** command has been issued for the neighbor without specifying any community types, that neighbor will receive all community attributes in the routes advertised to it. Using the **neighbor send-community add** command then to add an attribute will cause the switch to send *only* the specified community types in advertised routes. This results in the other community path attributes no longer being advertised to that BGP peer.
- If all community types are removed using the **neighbor send-community remove** command, the switch will then send routes with all community types. (This behavior is maintained for backward compatibility.) To remove all community path attributes from routes sent to the specified neighbor, use the **no neighbor send-community** command.
- After using this command, issue the **show active** command in ***router-bgp*** configuration mode to ensure that the intended attributes are being sent to the specified neighbor.

Examples

- These commands configure the switch to send large community attributes in the routes sent to the neighbor at address ***10.5.2.24***, then add extended community attributes as well.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.24 send-community large
switch(config-router-bgp)# neighbor 10.5.2.24 send-community add extended
switch(config-router-bgp)# show active
switch(config-router-bgp)# neighbor 10.5.2.24 send-community add extended
switch(config-router-bgp)# show active
router bgp 9
   neighbor 10.5.2.24 send-community extended large
   neighbor 10.5.2.24 maximum-routes 12000
switch(config-router-bgp)#
```

- These commands configure the switch to include extended and large community attributes in the routes sent to the neighbor at address ***10.5.2.27***, then remove the large attribute from the list of community types to be included.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.27 send-community extended large
switch(config-router-bgp)# neighbor 10.5.2.27 send-community remove large
switch(config-router-bgp)# show active
router bgp 600
   neighbor 10.5.2.27 send-community extended
   neighbor 10.5.2.27 maximum-routes 12000
switch(config-router-bgp)#
```

- These commands attempt to configure the switch to remove large community attributes from routes sent to the neighbor at address ***10.5.2.28***, but send all others. However, because the original command did not specify a list of attributes, the **remove** command has no effect, and all community path attributes are still included.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.28 send-community
switch(config-router-bgp)# neighbor 10.5.2.28 send-community remove large
switch(config-router-bgp)# show active
```

```
router bgp 600
   neighbor 10.5.2.28 send-community
   neighbor 10.5.2.28 maximum-routes 12000
switch(config-router-bgp)#
```

- These commands configure the switch to send only large community attributes in routes sent to the neighbor at address ***10.5.2.29***, then attempt to remove the large attribute from sent routes. However, because this removes the last specified attribute, *all* community path attributes (including large) will now be included.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.29 send-community large
switch(config-router-bgp)# neighbor 10.5.2.28 send-community remove large
switch(config-router-bgp)# show active
router bgp 600
   neighbor 10.5.2.29 send-community
   neighbor 10.5.2.29 maximum-routes 12000
switch(config-router-bgp)#
```

15.5.5.81 neighbor send-community link-bandwidth

The **`neighbor send-community link-bandwidth`** command is used to locally regenerate the link-bandwidth value to be advertised to a specific BGP neighbor or peer group. When this command is configured the regenerated link-bandwidth value is included in the extended community path attribute in UPDATE messages.

This command is used specifically for local regeneration of the link-bandwidth value. To send an explictly-configured link-bandwidth value, add an extended community to a route map instead. (see **`set extcommunity (route-map)`**) and include extended community attributes in UPDATE messages sent to that neighbor.

Note: The **`neighbor send-community`** command will override this command and vice-versa.

The **`no neighbor send-community`** command applies the system default (not sending community attributes in **`BGP UPDATE`** messages) for the specified peer.

The **`default neighbor send-community`** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`send-community link-bandwidth`** {**aggregate** [***reference_speed***] | **divide** {**equal** | **ratio**}}

`no neighbor` ***neighbor_ID*** **`send-community`**

`default neighbor` ***neighbor_ID*** **`send-community`**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- **aggregate** sends the sum of all link-bandwidth values for all paths toward a prefix to the specified neighbor or to each member of the specified peer group.
  - ***reference_speed*** optional value to specify a reference link speed in bits/second. Values range from ***0.0*** to ***4294967295.0***; larger values can also be expressed using the multiplier K (*10^3), M (*10^6), or G (*10^9). The link speed of the connection to the peer is divided by this value, and the resulting ratio is used to scale down the link-bandwidth advertised to the peer. If the result is >***1***, the multiplier is ignored and the full aggregate value is advertised.

- **divide** divides the cumulative link-bandwidth value described above among the peers in an Adj-RIB-Out either equally or proportionally.
  - **equal** divides the cumulative total link-bandwidth value equally among all peers in the same Adj-RIB-Out.
  - ***ratio*** divides the cumulative total link-bandwidth value among peers proportionally according to the speed of the connection to each peer in the Adj-RIB-Out.

Examples

- These commands configure the switch to locally regenerate the link-bandwidth value, dividing the bandwidth proportionally and including it in UPDATE messages to all peers in the peer group ***idaho***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor idaho send-community link-bandwidth
 divide ratio
switch(config-router-bgp)#
```

- These commands configure the switch to locally regenerate the link-bandwidth value, scale it down with a reference link speed of ***20*** gigabits/second, and include it in UPDATE messages to the neighbor at address ***10.5.2.24***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.24 send-community link-bandwidth
 aggregate 20G
switch(config-router-bgp)#
```

15.5.5.82 neighbor shutdown

The **`neighbor shutdown`** command disables the specified neighbor. Disabling a neighbor also terminates all of its active sessions and removes associated routing information.

The **`no neighbor shutdown`** command enables the specified peer.

The default neighbor shutdown command enables individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`shutdown`** reason ***REASON***

Parameter

neighbor_ID neighbor's IPv4 or IPv6 address or peer group name.

Examples

- This command disables the neighbor at ***10.5.2.23***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.23 shutdown
switch(config-router-bgp)#
```

- This command disables the neighbor at ***10.5.2.23*** with a reason - planned upgrade. The reason parameter is optional.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.5.2.23 shutdown reason Planned
 upgrade
switch(config-router-bgp)#
```

15.5.5.83 neighbor timers

The **neighbor timers** command configures the BGP keepalive and hold times for a specified peer connection. The **timers bgp** command configures the times on all peer connections for which an individual command is not specified.

- **Keepalive time** is the period between the transmission of consecutive keepalive messages.
- **Hold time** is the period the switch waits for a KEEPALIVE or UPDATE message before it disables peering.

The hold time must be at least ***3*** seconds and should be three times longer than the keepalive setting.

The **no neighbor timers** command applies the system default for the specified peer or group (the timers specified by the **timers bgp** command).

The **default neighbor timers** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **timers** ***keep_alive hold_time***

no neighbor ***neighbor_ID*** **timers**

default neighbor ***neighbor_ID*** **timers**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***keep_alive*** keepalive period, in seconds. Values include:
  - ***0*** keepalive messages are not sent.
  - ***1 to 3600*** keepalive time (seconds).
- ***hold_time*** hold time. Values include:
  - ***0*** peering is not disabled by timeout expiry; keepalive packets are not sent.
  - ***3 to 7200*** hold time (seconds).

Example

This command sets the keepalive time to ***30*** seconds and the hold time to ***90*** seconds for the connection with the peer at ***10.24.15.9***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.24.15.9 timers 30 90
switch(config-router-bgp)#
```

15.5.5.84 neighbor ttl maximum-hops

The **neighbor ttl maximum-hops** command configures the Generalized TTL Security Mechanism (GTSM) for the specified neighbor(s).

The **no neighbor ttl maximum-hops** command disables the GTSM configuration in the specified neighbor.

The **default neighbor ttl maximum-hops** command applies the system default configuration for individual neighbors; and applies the peer group's setting for neighbors that are members of a peer group.

Command-Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`ttl maximum-hops`** ***hop_number***

`no sneighbor` ***neighbor_ID*** **`ttl maximum-hops`**

`default neighbor` ***neighbor_ID*** **`ttl maximum-hops`**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***hop_number*** maximum count of hops from a BGP peer. Values range from ***0*** to ***254***.

Example

This command configures the TTL security for ***10.20.20.30*** with a maximum of ***4*** hops.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.20.20.30 ttl maximum-hops 4
switch(config-router-bgp)#
```

15.5.5.85 neighbor update-source

The **`neighbor update-source`** command specifies the interface that BGP sessions use for TCP connections. By default, BGP sessions use the neighbor's closest interface (also known as the best local address).

The **`no neighbor update-source`** command applies the system default (using best local address for TCP connections) for the specified peer or group.

The default **`neighbor update-source`** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **`no neighbor`** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

`neighbor` ***neighbor_ID*** **`update-source`** **INTERFACE**

`no neighbor` ***neighbor_ID*** **`update-source`**

`default neighbor` ***neighbor_ID*** **`update-source`**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- **INTERFACE** interface type and number. Options include:
  - **ethernet** ***e_num*** Ethernet interface specified by ***e_num***.
  - **loopback** ***l_num*** loopback interface specified by ***l_num***.
  - **management** ***m_num*** management interface specified by ***m_num***.
  - **port-channel** ***p_num*** port-channel interface specified by ***p_num***.
  - **vlan** VLAN interface specified by ***v_num***.

Example

This command configures the switch to use ***ethernet 10*** for TCP connections for the neighbor at ***10.2.2.14***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# neighbor 10.2.2.14 update-source ethernet 10
```

```
switch(config-router-bgp)#
```

15.5.5.86 neighbor weight

The **neighbor weight** command assigns a weight attribute value to paths from the specified neighbor. Weight is the first parameter that the BGP best-path selection algorithm considers. When multiple paths to a destination prefix exist, the best-path selection algorithm prefers the path with the highest weight. Other attributes are used only when all paths to the prefix have the same weight.

Weight values range from ***0*** to ***65535*** and are not propagated to other switches through route updates. The default weight for paths that the router originates is ***32768***; the default weight for routes received through BGP is ***0***.

A path's BGP weight is also configurable through route maps. Weight values set through route-map commands have precedence over neighbor weight command values.

The **no neighbor weight** command applies the system default (***32768*** for router-originated paths, ***0*** for routes received through BGP) for the specified peer or group.

The **default neighbor weight** command applies the system default for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***neighbor_ID*** **weight** ***weight_value***

no neighbor ***neighbor_ID*** **weight**

default neighbor ***neighbor_ID*** **weight**

Parameters

- ***neighbor_ID*** neighbor's IPv4 or IPv6 address or peer group name.
- ***weight_value*** weight value. Values range from ***1*** to ***65535***.

Example

This command specifies a weight of ***4000*** for all paths from the neighbor at ***10.1.2.5***.

```
switch(config)# router bgp 9
eswitch(config-router-bgp)#neighbor 10.1.2.5 weight 4000
switch(config-router-bgp)#
```

15.5.5.87 network (BGP)

The **network** command specifies a network for advertisement through UPDATE packets to BGP peers. The configuration zeros the host portion of the specified network address; for example, ***192.0.2.4/24*** is stored as ***192.0.2.0/24***. A route map option is available for assigning attributes to the network.

The command is available in Router-BGP and Router-BGP-Address-Family configuration modes. The mode in which the command is issued does not affect the command. The scope of the command depends on the specified network address:

- commands with an IPv4 address are advertised to peers activated in the IPv4 address family.
- commands with an IPv6 address are advertised to peers activated in the IPv6 address family.

The **no network** and **default network** commands remove the network from the routing table, preventing its advertisement.

Command Mode

Router-BGP Configuration

Router-BGP Address-Family Configuration

Command Syntax

`network` **NET_ADDRESS** [**ROUTE_MAP**]

`no network` **NET_ADDRESS**

`default network` **NET_ADDRESS**

Parameters

- **NET_ADDRESS** IP address range. Entry options include:
  - ***ipv4_subnet*** IPv4 subnet (CIDR notation).
  - ***ipv4_addr*** **mask** ***subnet*** IPv4 subnet (address-mask notation).
  - ***ipv6_prefix*** neighbor's IPv6 prefix (CIDR notation).
- **ROUTE_MAP** specifies route map that assigns attribute values to the network. Options include:
  - ***no parameter*** attributes are not assigned through a route map.
  - **route-map** ***map_name*** attributes listed by specified route map are assigned to the network.

Example

This command enables BGP advertising for the network located at ***10.1.2.5***. The configuration stores the network as ***10.1.2.5***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# network 10.1.2.5/24
switch(config-router-bgp)#
```

15.5.5.88 no neighbor

The **`no neighbor`** command removes all neighbor configuration commands for the specified neighbor.

Neighbor settings can also be removed individually; refer to the command description page of the desired command for details. Neighbor settings for a peer group *must* be removed individually.

Command Mode

Router-BGP Configuration

Command Syntax

`no neighbor` ***neighbor_ID***

`default neighbor` ***neighbor_ID***

Parameter

neighbor_ID neighbor's IPv4 or IPv6 address. This command does not accept a peer group name as an argument; peer group settings must be removed individually.

Example

This command removes all neighbor configuration commands for the neighbor at ***10.1.1.1***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# no neighbor 10.1.1.1
switch(config-router-bgp)#
```

15.5.5.89 peer-filter

The **peer-filter** command creates a peer filter group and places the switch in peer-filter configuration mode for that group. The peer-filter group parameters are defined using the **match as-range** command.

The **no peer-filter** and **default peer-filter** commands remove the peer-filter group from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

peer-filter ***filter_name***

no peer-filter ***filter_name***

default peer-filter ***filter_name***

Parameters

filter_name name of the peer filter.

Example

This command creates a peer filter called ***group1*** and places the switch in peer-filter configuration mode for that filter.

```
switch(config-router-bgp)# peer-filter group1
switch(config-peer-filter-group1)#
```

15.5.5.90 rd (Router-BGP VRF and VNI Configuration Modes)

The **rd** command adds a Route Distinguisher (RD) to VRF and VNI configuration modes. RDs internally identify routes belonging to a VRF or VNI to distinguish overlapping or duplicate IP address ranges. This allows the creation of distinct routes to the same IP address for different VPNs. The RD is a 64-bit number made up of an AS number or IPv4 address followed by a user-selected ID number.

If the switch is not running EVPN, an RD is not required for a VRF or VNI to function. Use **no** or **default** command forms to remove an RD from a VRF or VNI.

Note: Legacy RDs that were assigned in VRF Configuration Mode appear in **show vrf** outputs if an RD has not been configured using this command, but they no longer have an effect on the system. RDs assigned in the VNI Configuration Mode are displayed in the output of **show bgp evpn** command.

Command Modes

Router-BGP VRF Configuration

Router-BGP VNI Configuration

Command Syntax

rd ***admin_ID***:***local_assignment***

no rd

default rd

Parameters

- ***admin_ID*** an AS number or globally assigned IPv4 address identifying the entity assigning the RD. This should be an IANA-assigned identifying number.

- ***local_assignment*** a locally assigned number distinguishing the VRF. Values range from ***0***-***65535*** if the ***admin_ID*** is an IPv4 address, or from ***0***-***4294967295*** if the ***admin_ID*** is an AS number. If the ***admin_ID*** is an AS number, the ***local_assignment*** can also be entered in the form of an IPv4 address.

Examples

- These commands identify the administrator of the VRF named ***purple*** as AS ***530*** and assign ***12*** as its local number.

```
switch(config)# router bgp 50
switch(config-router-bgp)# vrf purple
switch(config-router-bgp-vrf-purple)# rd 530:12
switch(config-router-bgp-vrf-purple)#
```

- These commands identify the administrator of the MAC-VRF named ***bundle1*** as AS ***530*** and assign ***12*** as its local number.

```
cvx(config)# router bgp 100
cvx(config-router-bgp)# vni-aware-bundle bundle1
cvx(config-macvrf-bundle1)# rd 530:12
cvx(config-macvrf-bundle1)#
```

15.5.5.91 redistribute (BGP)

The **`redistribute`** command enables the redistribution of specified routes to the BGP domain.

The **`no redistribute`** and **`default redistribute`** commands disable route redistribution from the specified domain by removing the corresponding **`redistribute`** command from ***running-config***.

Note: Aggregate routes are redistributed automatically, and their redistribution cannot be disabled.

Command Mode

Router-BGP Configuration

Command Syntax

`redistribute` **ROUTE_TYPE** [**ROUTE_MAP**]

`no redistribute` **ROUTE_TYPE**

`default redistribute` **ROUTE_TYPE**

Parameters

- **ROUTE_TYPE** source from which routes are redistributed. Options include:
  - **connected** routes that are established when IP is enabled on an interface.
  - **match nssa-external** all OSPF NSSA external routes.
  - **match nssa-external 1** type 1 OSPF NSSA external routes.
  - **match nssa-external 2** type 2 OSPF NSSA external routes.
  - **ospf** internal routes from an OSPF domain.
  - **ospf match external** routes external to the AS, but imported from OSPF.
  - **ospf match internal** OSPF routes that are internal to the AS.
  - **ospf match nssa-external** all OSPF NSSA external routes.
  - **ospf match nssa-external 1** type 1 OSPF NSSA external routes.
  - **ospf match nssa-external 2** type 2 OSPF NSSA external routes.
  - **ospf3** routes from an OSPFv3 domain.
  - **ospf3 match external** routes external to the AS, but imported from OSPFv3.
  - **ospf3 match internal** OSPFv3 routes that are internal to the AS.
  - **rip** routes from a RIP domain.

- **static** IP static routes.
- **isis** IS-IS routes. Sub-options include:
  - **level-1** redistribute IS-IS level-1 routes.
  - **level-1-2** redistribute IS-IS level-1 and level-2 routes.
  - **level-2** redistribute IS-IS level-2 routes.
  - **route-map** route map reference.

  Note: While redistributing IS-IS routes into BGP, the **level-1** or **level-2** keyword can be used to selectively redistribute level-1 routes or level-2 routes into BGP. The **level-1** or **level-2** keyword is optional, and the command defaults to **level-2** when it is not configured.
- **ROUTE_MAP** route map that determines the routes that are redistributed. Options include:
  - ***no parameter*** all routes are redistributed.
  - **route-map *map_name*** only routes in the specified route map are redistributed.

Examples

- These commands redistribute internal OSPF routes into the BGP domain.

```
switch(config)# router bgp 1
switch(config-router-bgp)# redistribute ospf
switch(config-router-bgp)#
```

- These commands redistribute ISIS routes into the BGP domain in the ***address-family*** mode.

```
switch(config)# router bgp 1
switch(config-router-bgp)# address-family ipv4
switch(config-router-bgp-af)# redistribute isis level-1 route-map isis-to-
bgp-v4
switch(config-router-bgp-af)#
```

- These commands redistribute ISIS routes into the BGP domain in the ***router-bgp*** mode.

```
switch(config)# router bgp 1
switch(config-router-bgp)# redistribute isis level-1 route-map isis-to-bgp
switch(config-router-bgp)#
```

15.5.5.92 rib fib fec ecmp ordered

The **`rib fib fec ecmp ordered`** command is configured to enforce ordering of next hops as determined by the protocol agents in the FEC programmed for the route.

The **`no rib fib fec ecmp ordered`** command removes the Ordered FEC configuration from the **`running-config`**.

Command Mode

Router General Configuration Mode

Command Syntax

`rib fib fec ecmp ordered`

`no rib fib fec ecmp ordered`

Example

The **`rib fib fec ecmp ordered`** command configures the Ordered FEC feature on the switch.

```
switch(config)# router general
switch(config-router-general)# rib fib fec ecmp ordered
switch(config-router-general)#
```

15.5.5.93 router bgp

The **router bgp** command places the switch in router-BGP configuration mode. If BGP was not previously instantiated, this command creates a BGP instance with the specified AS number. Router-BGP configuration mode is not a group-change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

When a BGP instance exists, the command must include the AS number of the existing BGP instance. Running this command with a different AS number generates an error message.

The **no router bgp** and **default router bgp** commands delete the BGP instance.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

router bgp ***as_id***

no router bgp

default router bgp

Parameters

as_id Autonomous System (AS) number. Values range from ***1*** to ***4294967295***.

Examples

- This command creates a BGP instance with AS number ***64500***.

```
switch(config)# router bgp 64500
switch(config-router-bgp)#
```

- This command attempts to open a BGP instance with a different AS number from that of the existing instance. The switch displays an error and stays in the ***global*** configuration mode.

```
switch(config)# router bgp 64501
% BGP is already running with AS number 64500
switch(config)#
```

- This command exits the ***bgp*** configuration mode.

```
switch(config-router-bgp)# exit
switch(config)#
```

- This command deletes the BGP instance.

```
switch(config)# no router bgp
switch(config)#
```

15.5.5.94 router-id (BGP)

The **router-id** command sets the local router BGP router ID.

When no ID has been specified, the local router ID is set to the following:

- the loopback IP address when a single loopback interface is configured.
- the loopback with the highest IP address when multiple loopback interfaces are configured.
- the highest IP address on a physical interface when no loopback interfaces are configured.



Note: The router ID must be specified if the switch has no IPv4 addresses configured.

The **no router-id** and **default router-id** commands remove the **router-id** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

router-id ***id_num***

no router-id [***id_num***]

default router-id [***id_num***]

Parameter

id_num router ID number (32-bit dotted decimal notation).

Example

This command configures the fixed router ID address of ***10.10.4.11***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# router-id 10.10.4.11
switch(config-router-bgp)#
```

15.5.5.95 set large-community

Large communities are an optional transitive attribute of variable length. There are no predefined large-community types or values. Large communities may be configured alongside standard and extended communities within route-maps using additional configuration commands.

Large community values (***aa:nn:nn***) must consist of three decimal values each in the range (***0-4294967295***). All three sub-values of a large community value must be present. As-plain and As-dot notation are supported for the leading ASN value.

The **no** and **default** versions of the command return the command to the original configuration.

Command Mode

Route map configuration

Command Syntax

set large-community [large-community-list **LIST1** [[**LIST2**] [additive | delete]]]

no set large-community [large-community-list **LIST1** [[**LIST2**] [additive | delete]]]

default set large-community [large-community-list **LIST1** [[**LIST2**] [additive | delete]]]

Parameters

large-community-list Add a large community list entry.

- **LIST1** Name of large community list.
  - **additive** Adds to the existing community.
  - **delete** Deletes matching communities.

Examples

- The following route-map sets a number of large-community values using both as-plain and as-dot notation.

```
switch(config)# route-map LC permit 10
switch(config-route-map-LC)# set large-community 10.10:20:30 40.40:50:60
 1000:80:90
```

- The following route-map adds additional large-community values.

```
switch(config)# route-map LC permit 10
switch(config-route-map-LC)# set large-community 50:50:50 51:51:51 additive
```

- The following route-map removes the specified large-community values if they are present.

```
switch(config)# route-map LC permit 10
switch(config-route-map-LC)# set large-community 60:60:60 61:61:61 delete
```

- The following route-map matches multiple large-community values from large-community lists (***LC_a1*** **and** ***LC_a2***) and sets **local-pref** accordingly.

```
switch(config)# ip large-community-list LC_1 permit 10:20:30 40:50:60
switch(config)# ip large-community-list LC_2 permit 70:80:90
switch(config)# route-map LC permit 10
switch(config-map-LC)# match large-community LC_1 LC_2 exact_match
switch(config-map-LC)# set local-pref 111
```

15.5.5.96 show bgp labeled-unicast tunnel

The **show bgp labeled-unicast tunnel** command displays the contents of the BGP Labeled-Unicast (LU) tunnel table. The user can optionally specify a tunnel index parameter to view the specific single tunnel information.

Command Mode

EXEC

Command Syntax

show bgp labeled-unicast tunnel ***tunnel_index***

Parameters

tunnel_index index to view single tunnel information.

Examples

- This command displays the BGP LU tunnel table.

```
switch# show bgp labeled-unicast tunnel
Index Endpoint   Nexthop  Interface   Labels          Contributing Metric Metric 2 Pref Pref 2
----- ---------- -------- ----------- --------------- ------------ ------ -------- ---- ------
5     2.0.0.0/24 10.1.1.2 'Ethernet3' [ 123 899 900 ] Yes          0      100      200  0
6     2.0.1.0/24 10.1.1.2 'Ethernet3' [ 400 500 600 ] Yes          0      100      200  0
7     2.0.2.0/24 10.1.1.2 'Ethernet3' [ 400 500 600 ] Yes          0      100      200  0 switch#
```

- This command displays the BGP LU tunnel table for tunnel index ***4***.

```
switch# show bgp labeled-unicast tunnel 4
Index   Endpoint        Nexthop/Tunnel Index   Interface       Labels   Contributing Metric Metric 2  Pref  Pref 2
------  --------------- ---------------------- --------------- -------- ------------ ------ --------- ----- ------
4       10.253.0.10/32  10.1.0.0               Port-Channel111 [ 3 ]     Yes          0      0         200    0 switch#
```

15.5.5.97 show bgp convergence

The **show bgp convergence** command displays information about the Border Gateway Protocol (BGP) convergence state and other statistics about the BGP instance in the specified VRF or in all VRFs.

Command Mode

EXEC

Command Syntax

show bgp convergence [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE specifies VRF instances. Options include:

- ***no parameter*** displays BGP information for the context-active VRF.
- **vrf *vrf_name*** displays BGP information for the specified VRF.
- **vrf all** displays BGP information for all VRFs.
- **vrf default** displays BGP information for the default VRF.

Examples

- This command displays the output when no peers have joined before convergence.

```
switch# show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:02:30
Configured convergence slow peer timeout: 00:00:55
Convergence based update synchronization is enabled
Last Bgp convergence event : None
Bgp convergence state : Not Initiated (Waiting for the first peer to join)
 Convergence timer is not running
 Convergence timeout in use: 00:02:30
 Convergence slow peer timeout in use: 00:00:55
 First peer is not up yet
 All the expected peers are up: no
 All IGP protocols have converged: yes
 Outstanding EORs: 0, Outstanding Keepalives: 0
 Pending Peers: 2
 Total Peers: 2
 Established Peers: 0
 Disabled Peers: 0
 Peers that have not converged yet:
 IPv4 peers:
 201.1.1.1 (Session : Connect)
 202.1.1.1 (Session : Connect)
 IPv6 peers:
 None switch#
```

- This command displays the output when the first peer has joined before convergence.

```
switch# show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:02:30
Configured convergence slow peer timeout: 00:00:55
Convergence based update synchronization is enabled
Last Bgp convergence event 00:00:40 ago
Bgp convergence state : Pending (Waiting for EORs/Keepalives from peer(s)
 and IGP
convergence)
 Convergence timer running, will expire in 00:01:50
 Convergence timeout in use: 00:02:30
 Convergence slow peer timeout in use: 00:00:55
 First peer came up 00:00:13 ago
 All the expected peers are up: no
 All IGP protocols have converged: yes
 Outstanding EORs: 0, Outstanding Keepalives: 0
 Pending Peers: 1
 Total Peers: 2
 Established Peers: 1
 Disabled Peers: 0
 Peers that have not converged yet:
 IPv4 peers:
 201.1.1.1 (Session : Active)
 IPv6 peers:
 None switch#
```

- This command displays the output when the convergence timeout value is reached.

```
switch# show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:02:30
Configured convergence slow peer timeout: 00:00:55
Convergence based update synchronization is enabled
Last Bgp convergence event 00:02:44 ago
Bgp convergence state : Timeout reached
 Time taken to converge 00:02:30
 Pending Peers: 1
 Total Peers: 2
 Established Peers: 1
 Disabled Peers: 0
 Peers that did not converge before local bgp convergence:
 IPv4 peers:
 201.1.1.1 (Session : Active)
 202.1.1.1 (Session : Established)
 IPv6 peers:
 None switch#
```

- This command displays the output during the converged state.

```
switch#show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:05:00
Configured convergence slow peer timeout: 00:01:30
Convergence based update synchronization is enabled
Last Bgp convergence event 00:00:05 ago
Bgp convergence state : Converged
 Time taken to converge 00:00:02
 First peer came up 00:00:05 ago
 Pending Peers: 0
 Total Peers: 3
 Established Peers: 3
 Disabled Peers: 0
 Peers that did not converge before local bgp convergence:
 IPv4 peers:
 None
 IPv6 peers:
 None switch#
```

15.5.5.98 show bgp flow-spec

The **show bgp flow-spec ipv4** displays a brief description of each flowspec rule, including the matching rule and actions.

Command Mode

EXEC

Command Syntax

show bgp flow-spec [ipv4 | ipv6] [summary | detail] [vrf ***VRFNAME***]

Parameters

- **ipv4** Displays information releated to IPv4.
- **ipv6** Displays information releated to IPv6.
- **summary** Displays summarized BGP information.
- **detail** Displays detailed information.
- **vrf *VRFNAME*** Displays flow-spec information in the named VRF.

Related Command

show flow-spec

Examples

- The **show bgp flow-spec ipv4 summary** command displays the count of flowspec rules received from each peer:

```
switch(config)# show bgp flow-spec ipv4 summary
BGP summary information for VRF default
Router identifier 0.0.0.1, local AS number 10
Neighbor Status Codes: m - Under maintenance
  Neighbor     V  AS    MsgRcvd   MsgSent  InQ OutQ  Up/Down State  RulesRcd RulesAcc
  10.0.0.2     4  10         12         4    0    0 00:02:18 Estab  2        2
  10.0.1.2     4  10          6         4    0    0 00:02:18 Estab  0        0
```

- The **show bgp flow-spec detail** displays the full details of each flowspec rule including the peer(s) it was received from, BGP properties, and an expanded description of the matching rule:

```
switch(config)# show bgp flow-spec ipv4 detail
BGP Flow Specification rules for VRF default
Router identifier 0.0.0.1, local AS number 10
BGP Flow Specification Matching Rule for 10.2.3.0/24;*;
 Rule identifier: 3882065752
 Matching Rule:
   Destination Prefix: 10.2.3.0/24
   Source Prefix: *
 Paths: 1 available
  Local
    from 10.0.0.2 (10.1.1.2)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Actions: Drop
BGP Flow Specification Matching Rule for 10.2.4.0/24;10.2.0.0/16;IP:=6|
=17;DP:>1010&<1024;
 Rule identifier: 3882090640
 Matching Rule:
   Destination Prefix: 10.2.4.0/24
   Source Prefix: 10.2.0.0/16
   IP Protocol: =6 | =17
   Destination Port: >1010 & <1024
 Paths: 1 available
  Local
    from 10.0.0.2 (10.1.1.2)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Actions: Drop
```

15.5.5.99 show bgp instance

The **show bgp instance** command displays summary Border Gateway Protocol (BGP) information about the BGP instance in the specified VRF or in all VRFs.

Command Mode

EXEC

Command Syntax

show bgp instance [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE specifies VRF instances. Options include:

- ***no parameter*** displays BGP information for the context-active VRF.
- **vrf *vrf_name*** displays BGP information for the specified VRF.

- **vrf all** displays BGP information for all VRFs.
- **vrf default** displays BGP information for the default VRF.

Examples

- This command displays information about the BGP instance in the context-active VRF.

```
switch# show bgp instance
BGP instance information for VRF purple
BGP Local AS:  64497, Router ID: 1.2.3.5
Total peers:              5
Configured peers:         3
  UnConfigured peers:     2
  Disabled peers:         0
  Established peers:      3
Graceful restart helper mode enabled
End of rib timer timeout: 00:05:00
BGP Convergence timer is inactive
BGP Convergence information:
    BGP has converged:no
    Outstanding EORs:0,Outstanding Keepalives: 0
    Convergence timeout: 00:10:00 switch#
```

- This command displays information about the BGP instance in the default VRF.

```
switch# show bgp instance vrf default
BGP instance information for VRF default
BGP Local AS:  64503, Router ID: 1.2.3.5
Total peers:              1
Configured peers:         1
  UnConfigured peers:     0
  Disabled peers:         0
  Established peers:      0
Graceful restart helper mode enabled
End of rib timer timeout: 00:05:00
BGP Convergence timer is inactive
BGP Convergence information:
    BGP has converged:no
    Outstanding EORs:0,Outstanding Keepalives: 0
    Convergence timeout: 00:10:00 switch#
```

15.5.5.100 show bgp neighbors history

The **show bgp neighbors history** command stores and displays a list of failed BGP connection attempts for each peer. This may be particularly useful while troubleshooting flappy connections. If dynamic peering is enabled, the failure history will be remembered even after the peers are no longer present.

Command Mode

EXEC

Command Syntax

show bgp neighbors [**PEER** | **PREFIX** | peer-group **PEER_GROUP**] **history** [connect-failures][vrf ***VRF***

Parameters

- **PEER** An IPv4 or IPv6 valid address.
- **PREFIX** An IPv4 or IPv6 valid prefix.
- **peer-group PEER_GROUP** A peer group name.
- **connect-failures** Optional and will not affect the result.
- **vrf *VRF*** A VRF name. If it is not supplied, the command will act upon the VRF default.

Guidelines

Relevant error messages are recorded by default, without any configuration. To clear all messages for a peer or group of peers, though, it is necessary to use the command **`clear bgp history`**. The syntax for this command is described as:

```
switch# clear bgp [PEER|PREFIX|peer-group PEER_GROUP] history [connect-fail
ures][vrf VRF]
```

If no peer, prefix, or peer-group is supplied, this command will clear the history for all peers in the specified VRF.

- The number of recorded messages is limited to eight per peer.
- Only errors that occur prior to session establishment will be recorded.
- The **`show bgp neighbors history`** is available only with the multi-agent protocol model.

Related Command

clear bgp history

Example

For each peer, its address will be printed at the first line, along with the VRF it is part of. Then, a table prints with the following columns:

- **Type**: the peer connection type. May be Static or Dynamic.
- **AS**: the remote Autonomous System number.
- **Time**: the time of the failure, using the local timezone.
- **Event**: a description related to the cause of the failed BGP connection.

```
switch> show bgp neighbors history
1.1.1.2 VRF default
Type      AS       Time                      Event
Static    65538    Mon 2019-05-13 04:16:24  Connect (No route to host)
Static    65538    Mon 2019-05-13 04:16:31  Connect (No route to host)
Static    65538    Mon 2019-05-13 04:16:39  Connect (No route to host)
Static    65538    Mon 2019-05-13 04:16:47  Connect (No route to host)
Static    65538    Mon 2019-05-13 04:16:55  Connect (No route to host)
Static    65538    Mon 2019-05-13 04:17:03  Connect (No route to host)
Static    65538    Mon 2019-05-13 04:18:17  bad AS number
Static    65538    Mon 2019-05-13 04:19:40  bad AS number
```

15.5.5.101 show bgp update-group

The **`show bgp update-group`** command displays how peers are grouped into update groups and can be used to verify that peers with different RCF functions with identical contents are grouped together.

Command Mode

EXEC

Command Syntax

`show bgp update-group`

Examples

This command displays information about how BGP peers are grouped into update groups.

```
switch# show bgp update-group
switch#
```

15.5.5.102 show flow-spec

The **show flow-spec** command displays an overall status of how many flowspec rules were received and how many were installed.

Command Mode

EXEC

Command Syntax

show flow-spec (ipv4 | ipv6) [summary][vrf ***VRFNAME***]

Parameters

- **ipv4** Displays information releated to IPv4.
- **ipv6** Displays information releated to IPv6.
- **summary** Displays summary of flow-spec rule.
- **vrf *VRFNAME*** Displays flow-spec information in the named VRF.

Related Command

show bgp flow-spec

Examples

- The **show flow-spec ipv4 summary** command displays an overall status of how many flowspec rules were received and how many were installed:

```
switch(config)# show flow-spec ipv4 summary
Flow specification rules summary for VRF default
  Total number of rules: 2
  Number of installed rules: 2
```

- The **show flow-spec ipv4** displays the installation status of the rule, and a counter of how many hits it has accumulated. This command also compiles the received flowspec rules into rules that can be programmed into the TCAM. For example, logical expressions on values such as the destination port are converted to ranges, as shown below:

```
switch(config)# show flow-spec ipv4
Flow specification rules for VRF default
Applied on: Ethernet47/1
  Flow-spec rule: 10.2.3.0/24;*;
    Rule identifier: 3882065752
    Matches:
      Destination prefix: 10.2.3.0/24
    Actions:
      Police: 80 Mbps (10 MBps)
      Redirect: VRF customer1
                Route via LDP tunnel index 4, MPLS label 100123
                Route via LDP tunnel index 1, MPLS label 116507
    Status:
      Installed: yes
      Counter: 312 packets
  Flow-spec rule: 10.2.4.0/24;10.2.0.0/16;IP:=6|=17;DP:>1010&<1024;
    Rule identifier: 3882090640
    Matches:
      Destination prefix: 10.2.4.0/24
      Source prefix: 10.2.0.0/16
      Next protocol: 17
                     6
      Destination port: 1011-1023
    Actions:
      Police: 80 Mbps (10 MBps)
      Redirect: VRF customer1
```

```
                Route via LDP tunnel index 4, MPLS label 100123
                Route via LDP tunnel index 1, MPLS label 116507
    Status:
      Installed: yes
      Counter: 0 packets
```

15.5.5.103 show ip as-path access-list

The **show ip as-path access-list** command displays BGP filters on the switch. Specifying an access list displays the statements from that access list. Entering the command without parameters displays the statements from all access lists on the switch.

Command Mode

EXEC

Command Syntax

show ip as-path access-list [***list_name***]

Parameters

list_name the name of an AS path access list.

Example

This command displays the contents of the AS path access list named ***list1***.

```
switch# show ip as-path access-list list1
ip as-path access-list list1 deny _3$
ip as-path access-list list1 permit .*
switch#
```

15.5.5.104 show ip bgp

The **show ip bgp** command displays Border Gateway Protocol (BGP) IPv4 routing table entries. The output format depends on the command parameters:

- **data-block format** displays comprehensive information for each specified BGP routing-table entry.
- **tabular format** displays routing-table entries for the specified IPv4 addresses.

Command Mode

EXEC

Command Syntax

show ip bgp [**FILTER**][**VRF_INSTANCE**]

Parameters

- **FILTER** routing-table entries to display. Options include:
  - ***no parameter*** displays all routing-table entries in tabular format.
  - **detail** displays all routing-table entries in data-block format.
  - ***ipv4_addr*** displays IPv4 host address in data-block format.
  - **PREFIX** displays the route information of the specified IPv4 prefix in data block format. Options include:
    - **detail *ipv4_prefix***displays the detailed route information of specified IPv4 prefix in data block format.
    - **longer-prefixes *ipv4_prefix*** displays the route information of IPv4 prefix in tabular block format.
    - **longer-prefixes detail *ipv4_prefix*** displays the detailed route information of specified IPv4 prefix in data block format.
  - **community-list *cmnty_list_name*** displays BGP routes filtered by the specified community list.

- **installed** displays the information of installed BGP routes.
- **labeled-unicast** displays the information of labeled-unicast BGP routes only.
- **not-installed** displays the information of BGP routes that are not installed.

- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Guidelines

You must provide the IPv4 prefix in CIDR notation.

Examples

- This command displays the BGP routing table with prefix “L” flag for all BGP LU route entries.

```
switch# show ip bgp
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 100
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

        Network          Next Hop     Metric     LocPref     Weight     Path
* > L   2.0.0.1/32       1.1.1.2      0          100         0          300 i
* #     2.0.0.1/32       1.0.0.2      0          100         0          200 ?
* > L   2.0.0.2/32       1.1.1.2      0          100         0          300 i
* #     2.0.0.2/32       1.0.0.2      0          100         0          200 ?
* > L   2.0.0.3/32       1.1.1.2      0          100         0          300 i
* #     2.0.0.3/32       1.0.0.2      0          100         0          200 ?
* > L   2.0.0.4/32       1.1.1.2      0          100         0          300 i
* #     2.0.0.4/32       1.0.0.2      0          100         0          200 ?
* > L   2.0.0.5/32       1.1.1.2      0          100         0          300 i
* #     2.0.0.5/32       1.0.0.2      0          100         0          200 ?
switch#
```

- This command displays the routing-table information of unicast routes for a default VRF.

```
switch# show ip bgp
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 100
BGP routing table entry for 2.0.0.1/32
    Paths: 2 available
    300
        1.1.1.2 labels [ 101 102 103 104 ] from 1.1.1.2 (1.1.1.2)
        Origin IGP, metric 0, localpref 100, weight 0, valid, external, best
        Rx path id: 0x0
    200
        1.0.0.2 from 1.0.0.2 (0.0.1.1)
        Origin INCOMPLETE, metric 0, localpref 100, weight 0, valid,
 external,
not installed (labeled-route present)
switch#
```

- This command displays the BGP routing-table entry for the ***10.100.1.0/24*** network.

```
switch# show ip bgp 10.100.1.0/24
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
BGP routing table entry for 10.100.1.0/24
 Paths: 1 available
  64496 64497 65536
```

```
    10.1.0.100 from 10.1.0.100 (10.0.0.100)
      Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received
01:57:33 ago, valid, external, best
      Community: 655:23590 64496:1000
      Rx SAFI: Unicast switch#
```

- This command displays the label stack associated with the route for a default VRF.

```
switch# show ip bgp detail
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 100
BGP routing table entry for 2.0.0.1/32
    Paths: 2 available
    200
        1.0.0.2 from 1.0.0.2 (0.0.1.1)
        Origin INCOMPLETE, metric 0, localpref 100, weight 0, valid,
 external, best
    300
        1.1.1.2 labels [ 101 102 103 104 ] from 1.1.1.2 (1.1.1.2)
        Origin IGP, metric 0, localpref 100, weight 0, valid, external
        Rx path id: 0x0
        Rx SAFI: Labels
        Tunnel RIB eligible switch#
```

- This command displays the BGP routing-table entry for the ***10.105.1.1/24*** network, including the reason why the route was discarded by the best-path algorithm. The reason for discarding a route is preceded by the label "**Not best:**".

```
switch# show ip bgp 10.105.1.1/24 detail
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status: [a.b.c.d] - Route is  queued for advertisement to peer.
BGP routing table entry for 10.105.1.0/24
 Paths: 2 available
  64510
    10.2.0.101 from 10.2.0.101 (12.0.0.101)
      Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received
00:00:58 ago, valid, external, best
      Rx SAFI: Unicast
  64496
    10.1.0.100 from 10.1.0.100 (10.0.0.100)
      Origin INCOMPLETE, metric 42, localpref 100, IGP metric 1, weight 0,
 received
00:00:33 ago, valid, external
      Rx SAFI: Unicast
      Not best: Origin
 Advertised to 2 peers:
  peer-group EXTERNAL:
    10.1.0.100
  peer-group INTERNAL:
    10.3.0.103 switch#
```

15.5.5.105 show ip bgp community

The **`show ip bgp community`** command displays Border Gateway Protocol (BGP) routing table entries, filtered by community.

Command Mode

EXEC

Command Syntax

show ip bgp community **COMM_1** [**COMM_2... COMM_n**][**MATCH_TYPE**][**DATA_OPTION**] [**VRF_INSTANCE**]

Parameters

- **COMM_x** community number or name, as specified in the route map that sets the community list number.
  - **GSHUT** well-known graceful shutdown community.
  - ***aa:nn*** AS and network number, separated by colon. Each value ranges from ***1*** **to** ***4294967295*****.**
  - ***comm_num*** community number. Values range from ***1*** to ***4294967040***.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise the route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- **MATCH_TYPE** routes are filtered based on their communities. Options include:
  - ***no parameter*** routes must match at least one community in the list.
  - **exact** route must match all communities and include no other communities.
  - **regex** display routes matching the regular expression of communities.
- **DATA_OPTION** type of information the command displays. Options include:
  - ***no parameter*** displays table of the routing entry line items.
  - **detail** displays data block for each routing table entry.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Guidelines

The interpretation of regular expressions is always based on string mode but not on the ACL configuration.

Example

This command displays the BGP routing table entries with the community ***64496:1000***.

```
switch# show ip bgp community 64496:1000 detail
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
BGP routing table entry for 10.100.1.0/24
 Paths: 1 available
  64496 64497 65536
    10.1.0.100 from 10.1.0.100 (10.0.0.100)
      Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 00:03:16 ago, valid, external, best
      Community: 655:23590 64496:1000
      Rx SAFI: Unicast switch#
```

15.5.5.106 show ip bgp installed

The **show ip bgp installed** command displays the list of installed routes in the RIB.

Command Mode

EXEC

Command Syntax

show ip bgp installed

Example

This command displays the list of installed routes in the RIB.

```
switch# show ip bgp installed
BGP routing table information for VRF default
Router identifier 1.0.0.2, local AS number 100
Route status codes: s - suppressed, * - valid, > - active, # - not installed,
 E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

        Network             Next Hop         Metric  LocPref Weight Path
 * >    6.0.0.0/24          1.0.0.1          0       100     0      ?
switch#
```

15.5.5.107 show ip bgp neighbors (route type)

The **show ip bgp neighbors (route type)** command displays information for next-hop routes to a specified IPv4 neighbor. The **show ip bgp neighbors (route-type) community** command displays the same information for routes filtered by communities.

The output format depends on the selected **FILTER** parameter:

- data-block format displays comprehensive information for each specified route.
- tabular format displays routing table entries in tabular format for the specified IP addresses.

Commands that do not include a route type revert to the **show ip bgp neighbors** command.

Command Mode

EXEC

Command Syntax

show ip bgp neighbors ***neighbor_addr*** **HOPDIRECT** [**FILTER**] [**VRF_INSTANCE**]

show ip bgp neighbors ***neighbor_addr*** [**ROUTE_TYPE**] **HOPDIRECT** [**detail**]

Related Commands

- **show ip bgp neighbors**
- **show ip bgp neighbors (route-type) community**

Parameters

- ***neighbor_addr*** location of the neighbor.
- **ROUTE_TYPE** filters route on route type. Options include:
  - **ipv4 unicast** displays IPv4 unicast routes.
  - **ipv6 unicast** displays IPv6 unicast route
- **HOPDIRECT** filters route on the basis of direction from neighbor. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- **FILTER** routing table entries that the command displays. Values include:
  - ***no parameter*** displays all routing table entries in tabular format.
  - **detail** displays all routing table entries in data-block format.
  - ***ipv4_addr*** displays IPv4 host address in data-block format.

- ***ipv4_prefix*** displays the route information of specified IPv4 prefix in data-block format. Option includes:
  - **longer-prefixes** displays the route information of IPv4 prefix in data-block format.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

This command displays information for routes advertised to the neighbor at ***10.3.0.103***.

```
switch# show ip bgp neighbors 10.3.0.103 advertised-routes
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

         Network                Next Hop            Metric  LocPref Weight  Path
 * >     10.1.0.0/24            10.3.0.102            -       100     -       i
 * >     10.2.0.0/24            10.3.0.102            -       100     -       i
 * >     10.3.0.0/24            10.3.0.102            -       100     -       i
 * >     10.100.0.0/24          10.1.0.100            200     100     -       64496 i
 * >     10.100.1.0/24          10.1.0.100            -       100     -       64496 64497 65536 i
 * >     10.100.2.0/24          10.1.0.100            42      100     -       64496 ?
 * >     10.101.0.0/24          10.2.0.101            -       100     -       64510 i
 * >     10.101.1.0/24          10.2.0.101            -       100     -       64510 i
 * >     10.101.2.0/24          10.2.0.101            -       100     -       64510 i switch#
```

15.5.5.108 show ip bgp neighbors (route-type) community

The **`show ip bgp neighbors (route type) community`** command displays information for next-hop routes to a specified neighbor. Routes are filtered by community.

The **`show ip bgp neighbors (route type)`** command displays the same information for routes filtered by IP addresses and subnets.

Command Mode

EXEC

Command Syntax

`show ip bgp neighbors` ***addr*** **RTE `community` CM_1 [CM_2...CM_n][MATCH][INFO][VRF_INST]**

Related Commands

- **`show ip bgp neighbors`**
- **`show ip bgp neighbors (route type)`**

Parameters

- ***addr*** location of the neighbor.
- **RTE** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- **CM_x** community number or name, as specified in the route map that sets the community list number. The command must list at least one of the following community identifiers:
  - **GSHUT** well-known graceful shutdown community.

  - ***aa:nn*** AS and network number, separated by colon. Each value ranges from ***1*** to ***4294967295***.
  - ***comm_num*** community number. Values range from ***1*** to ***4294967040***.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- **MATCH** routes are filtered based on their communities.
  - ***no parameter*** routes must match at least one community in the list.
  - **exact** route must match all communities and include no other communities.
- **INFO** type of information the command displays. Values include:
  - ***no parameter*** displays table of routing entry line items.
  - **detail** displays data block for each routing table entry.
- **VRF_INST** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

This command lists the routes advertised to the neighbor at ***10.3.0.103*** with community ***655:23590***.

```
switch# show ip bgp neighbors 10.3.0.103 advertised-routes community 655:23590
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

         Network                Next Hop            Metric  LocPref Weight  Path
 * >     10.100.1.0/24          10.1.0.100            -       100     -       64496 64497 65536 i
 switch#
```

15.5.5.109 show ip bgp neighbors regexp

The **`show ip bgp neighbors regexp`** command displays information for next-hop routes to a specified IPv4 neighbor that match the AS path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

`show ip bgp neighbors` ***addr*** **RTE** **`regexp`** ***as_paths*** [**VRF_INST**]

Related Commands

- **`show ip bgp neighbors`**
- **`show ip bgp neighbors (route type)`**

Parameters

- ***addr*** location of the neighbor.
- **RTE** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.

- ***as_paths*** list of AS paths, formatted as a regular expression. Regular expressions are pattern-matching strings that are composed of text characters and operators.
- **VRF_INST** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

This command lists the routes advertised to the neighbor at ***10.3.0.103*** where the AS path is ***64496***.

```
switch# show ip bgp neighbors 10.3.0.103 advertised-routes regex ^64496$
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP,  b - backup, L = labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST -Cluster List, LL Nexthop - Link Local Nexthop

        Network               Next Hop              Metric  LocPref Weight  Path
 * >     10.100.0.0/24       10.1.0.100       200     100     -       64496 i
 * >     10.100.2.0/24       10.1.0.100       42      100     -       64496 ?
switch#
```

15.5.5.110 show ip bgp neighbors

The **`show ip bgp neighbors`** command displays Border Gateway Protocol (BGP) and TCP-session data for a specified IPv4 BGP neighbor, or for all IPv4 BGP neighbors if an address is not specified.

Command Mode

EXEC

Command Syntax

`show ip bgp neighbors` [**NEIGHBOR_ADDR**] [**VRF_INSTANCE**]

Parameters

- **NEIGHBOR_ADDR** location of the neighbors. Options include:
  - ***no parameter*** command displays information for all IPv4 BGP neighbors.
  - ***ipv4_addr*** command displays information for specified neighbor.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for the default VRF.

Related Commands

- **`show ip bgp neighbors (route type)`**
- **`show ip bgp neighbors (route-type) community`**

Examples

- This command displays information of the neighbor at ***10.1.0.100***.

```
switch# show ip bgp neighbors 10.1.0.100
BGP neighbor is 10.1.0.100, remote AS 64496, external link
  BGP version 4, remote router ID 10.0.0.100, VRF default
```

```
  Inherits configuration from and member of peer-group EXTERNAL
  Negotiated BGP version 4
  Member of update group 3
  Last read 00:00:17, last write 00:00:18
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Connect timer is inactive
  Idle-restart timer is inactive
  BGP state is Established, up for 00:05:17
  Number of transitions to established: 1
  Last state was OpenConfirm
  Last event was RecvKeepAlive
  Neighbor Capabilities:
    Multiprotocol IPv4 Unicast: advertised and received and negotiated
    Four Octet ASN: advertised and received and negotiated
    Route Refresh: advertised and received and negotiated
    Send End-of-RIB messages: advertised and received and negotiated
    Additional-paths recv capability:
      IPv4 Unicast: advertised
    Additional-paths send capability:
      IPv4 Unicast: received
  Restart timer is inactive
  End of rib timer is inactive
  Message Statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                4          4
    Keepalives:             7          7
    Route-Refresh:          0          0
    Total messages:        12         12
  Prefix Statistics:
                         Sent       Rcvd
    IPv4 Unicast:           9          4
    IPv6 Unicast:           0          0
    IPv4 SR-TE:             0          0
    IPv6 SR-TE:             0          0
  Inbound updates dropped by reason:
    AS path loop detection: 0
    Enforced First AS: 0
    Originator ID matches local router ID: 0
    Nexthop matches local IP address: 0
    Unexpected IPv6 nexthop for IPv4 routes: 0
    Nexthop invalid for single hop eBGP: 0
  Inbound updates with attribute errors:
    Resulting in removal of all paths in update (treat-as-withdraw): 0
    Resulting in AFI/SAFI disable: 0
    Resulting in attribute ignore: 0
  Inbound paths dropped by reason:
    IPv4 labeled-unicast NLRIs dropped due to excessive labels: 0
    IPv6 labeled-unicast NLRIs dropped due to excessive labels: 0
  Outbound paths dropped by reason:
    IPv4 local address not available: 0
    IPv6 local address not available: 0
Local AS is 64500, local router ID 10.0.0.102
TTL is 255, BGP neighbor may be upto 1 hops away
Local TCP address is 10.1.0.102, local port is 179
Remote TCP address is 10.1.0.100, remote port is 33171
Auto-Local-Addr is disabled
TCP Socket Information:
  TCP state is ESTABLISHED
  Recv-Q: 0/32768
```

```
  Send-Q: 0/32768
  Outgoing Maximum Segment Size (MSS): 1448
  Total Number of TCP retransmissions: 0
  Options:
    Timestamps enabled: yes
    Selective Acknowledgments enabled: yes
    Window Scale enabled: yes
    Explicit Congestion Notification (ECN) enabled: no
  Socket Statistics:
    Window Scale (wscale): 9,9
    Retransmission Timeout (rto): 204.0ms
    Round-trip Time (rtt/rtvar): 3.0ms/5.4ms
    Delayed Ack Timeout (ato): 40.0ms
    Congestion Window (cwnd): 10
    TCP Throughput: 39.20 Mbps
    Advertised Recv Window (rcv_space): 28960 switch#
```

- This command displays neighbor information for all neighbors.

```
switch# show ip bgp neighbors
BGP neighbor is 172.24.77.5, remote AS 100, external link
    BGP version 4, remote router ID 172.24.77.5, VRF default
...
    Neighbor Capabilities:
        Multiprotocol IPv4 Unicast: advertised
        Multiprotocol IPv4 Labeled Unicast: advertised and received and
 negotiated
        Four Octet ASN: advertised and received
        Route Refresh: advertised
        Send End-of-RIB messages: advertised
        Additional-paths Receive:
            IPv4 Unicast: advertised
            IPv4 Labeled Unicast: advertised
...
    Inbound updates dropped by reason:
        AS path loop detection: 0
        Enforced First AS: 0
        Malformed MPBGP routes: 0
        Originator ID matches local router ID: 0
        Nexthop matches local IP address: 0
        Unexpected IPv6 nexthop for IPv4 routes: 0
    Inbound paths dropped by reason:
        IPv4 labeled-unicast NLRIs dropped due to excessive labels: 0
 switch#
```

15.5.5.111 show ip bgp not-installed

The **show ip bgp not-installed** command displays the list of non-installed routes in the RIB.

Command Mode

EXEC

Command Syntax

show ip bgp not-installed

Example

This command displays the list of non-installed routes in the RIB.

```
switch# show ip bgp not-installed
BGP routing table information for VRF default
Router identifier 1.0.0.2, local AS number 100
```

```
Route status codes: s - suppressed, * - valid, > - active, # - not installed,
 E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

        Network             Next Hop         Metric  LocPref Weight Path
 * #    7.0.0.0/24          1.0.0.1          0       100     0      ?
switch#
```

15.5.5.112 show ip bgp paths

The **show ip bgp paths** command displays all BGP AS paths in the database.

Command Mode

EXEC

Command Syntax

show ip bgp paths [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE specifies VRF instances.

- ***no parameter*** displays routing table for context-active VRF.
- **vrf *vrf_name*** displays routing table for the specified VRF.
- **vrf all** displays routing table for all VRFs.
- **vrf default** displays routing table for default VRF.

Display Values

- **Refcount**: number of routes using a listed path.
- **Metric**: the path's Multi Exit Discriminator (MED).
- **Path**: the route's AS path and its origin code.

Example

This command displays all BGP AS paths in the switch's database.

```
switch# show ip bgp paths
Refcount Metric     Path
6        0          64510 64505 64506 64507 i (HashID 9)
6        0          64510 ? (HashID 8)
12       0          65530 65531 65532 e (HashID 5)
12       0          i (HashID 6)
6        0          64100 64200 i (HashID 4)
28       0          i (HashID 1)
7        0          ? (HashID 2)
40       0          64510 i (HashID 10)
19       0          64510 i (HashID 7)
2        0          i (HashID 3)
switch#
```

15.5.5.113 show ip bgp peer-group

The **show ip bgp peer-group** command displays the BGP version, address family, and group members for all BGP peer groups defined on the switch.

Command Mode

EXEC

Command Syntax

show ip bgp peer-group [**GROUP**][**VRF_INSTANCE**]

Parameters

- **GROUP** peer group for which command displays information. Options include:
  - ***no parameter*** command displays information for all peer groups.
  - ***group_name*** name of peer group for which command displays information.
- **VRF_INSTANCE** specifies VRF instances.
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

This command displays BGP peer group information for all peer groups on the switch.

```
switch# show ip bgp peer-group
BGP peer-group is EXTERNAL
  BGP version 4
  Static peer-group members:
    VRF default:
      10.1.0.100, state: Connect
        Negotiated MP Capabilities:
            IPv4 Unicast: No
            IPv6 Unicast: No
            IPv4 SR-TE: No
            IPv6 SR-TE: No
      10.2.0.101, state: Connect
        Negotiated MP Capabilities:
            IPv4 Unicast: No
            IPv6 Unicast: No
            IPv4 SR-TE: No
            IPv6 SR-TE: No
BGP peer-group is INTERNAL
  BGP version 4
  Listen-range subnets:
    VRF default:
      10.3.0.0/24, remote AS 64500
  Dynamic peer-group members:
    VRF default:
switch#
```

15.5.5.114 show ip bgp regexp

The **show ip bgp regexp** command displays Border Gateway Protocol (BGP) IPv4 routing-table entries that match the AS path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

show ip bgp regexp ***as_paths*** [**VRF_INSTANCE**]

Parameters

- ***as_paths*** list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.

  Note: The AS delimiter (_) regular expression is not supported when BGP routes are filtered by community lists and the command output does not display BGP route information.
- **VRF_INSTANCE** specifies the VRF instance of the BGP routing table to be displayed. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

This command displays information about the BGP IPv4 routes in the context-active VRF where the AS path is ***64510***.

```
switch# show ip bgp regex ^64510$
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP,  b - backup, L = labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST -Cluster List, LL Nexthop - Link Local Nexthop

        Network               Next Hop              Metric  LocPref Weight  Path
 *       10.2.0.0/24         10.2.0.101       0       100     0       64510 i
 * >     10.101.0.0/24       10.2.0.101       0       100     0       64510 i
 * >     10.101.1.0/24       10.2.0.101       0       100     0       64510 i
 * >     10.101.2.0/24       10.2.0.101       0       100     0       64510 i switch#
```

15.5.5.115 show ip bgp summary

The **`show ip bgp summary`** command displays the summary of all IPv4 and IPv6 BGP neighbors based on exchanged Address Family Identifiers (AFI) and Subsequent Address Family Identifiers (SAFI) negotiations where AFI is “IP” and SAFI is “unicast” information.

Command Mode

EXEC

Command Syntax

`show ip bgp summary` [**VRF_INSTANCE**]

Parameters

- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Display Values

Header Row

- **`BGP router identifier`**: the router identifier loopback address or highest IP address.
- **`Local AS Number`**: AS number assigned to the switch.

Neighbor Table Columns

- **(First) `Neighbor`**: neighbor’s IP address.
- **(Second) `V`**: BGP version number.

- **(Third) `AS`**: neighbor's AS number.
- **(Fourth) `MsgRcvd`**: messages received from the neighbor.
- **(Fifth) `MsgSent`**: messages sent to neighbor.
- **(Sixth) `InQ`**: messages queued from neighbor.
- **(Seventh) `OutQ`**: messages queued to send neighbor.
- **(Eighth) `Up/Down`**: period the BGP session has been **`Established`**, or its current status.
- **(Ninth) `State`**: State of the BGP session and the number of routes received from a neighbor.

After the maximum number of routes are received, the ninth field displays **`PfxRcd`**, and the connection becomes **`Idle`**. Maximum number of routes is set using the **`maximum paths (BGP)`** command.

Related Command

`show ipv6 bgp summary`

Example

This command displays the status of the switch's BGP connections.

```
switch# show ip bgp summary
BGP summary information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS       MsgRcvd   MsgSent  InQ OutQ  Up/Down State   PfxRcd PfxAcc
  10.1.0.100       4  64496      1075      1083    0    0 00:04:04 Connect
  10.2.0.101       4  64510      1079      1088    0    0 00:04:14 Connect switch#
```

15.5.5.116 show ip community-list

The **`show ip community-list`** command displays the BGP community lists configured on the switch.

Command Mode

EXEC

Command Syntax

`show ip community-list` [**COMMUNITY_LIST**]

Parameters

COMMUNITY_LIST community list for which command displays information. Options include:

- ***no parameter*** command displays information for all community lists.
- ***listname*** name of the community list (text string).

Example

This command displays the BGP paths in the switch's database.

```
switch# show ip community-list hs-comm-list
ip community-list hs-comm-list permit 0:10 switch#
```

15.5.5.117 show ip extcommunity-list

The **`show ip extcommunity-list`** command displays the BGP extended community lists configured on the switch.

Command Mode

EXEC

Command Syntax

`show ip extcommunity-list` [**COMMUNITY_LIST**]

Parameters

COMMUNITY_LIST extended community list for which command displays information. Options include:

- ***no parameter*** command displays information for all extended community lists.
- ***listname*** command displays information for the specified extended community list.

Example

This command displays information for all extended extcommunity lists on the switch.

```
switch# show ip extcommunity-list
ip extcommunity-list hs-extcomm-list permit rt 3050:20
ip extcommunity-list hs-extcomm-list permit soo 172.17.52.2:30
ip extcommunity-list hs-extcomm-list permit rt 3050:70000 switch#
```

15.5.5.118 show ipv6 bgp

The **show ipv6 bgp** command displays IPv6 Border Gateway Protocol (BGP) routing-table entries. The output format depends on the command parameters:

- **data-block format** displays comprehensive information for each specified BGP routing-table entry.
- **tabular format** displays routing-table entries for specified IPv6 addresses.

Command Mode

EXEC

Command Syntax

show ipv6 bgp [**FILTER**][**VRF_INSTANCE**]

Parameters

- **FILTER** routing table entries that the command displays. Options include:
  - ***no parameter*** displays all routing-table entries in tabular format.
  - **detail** displays all routing-table entries in data-block format.
  - ***ipv6_addr*** displays IPv6 host address in data-block format.
  - ***ipv6_prefix*** displays the route information of specified IPv6 prefix address in data-block format. Options include:
    - **detail** displays the detailed route information of specified IPv6 prefix address in data-block format.
    - **longer-prefixes** displays the route information of IPv6 prefix in data-block format.
    - **longer-prefixes detail** displays detailed route information of specified IPv6 prefix in data-block format.
  - **community-list *cmnty_list_name*** displays BGP routes filtered by the specified community list.
  - **installed** displays the information of installed BGP routes.
  - **labeled-unicast** displays the information of labeled-unicast BGP routes only.
  - **not-installed** displays the information of BGP routes that are not installed.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Guidelines

You must provide the IPv6 prefix in CIDR notation.

Related Command

show ip bgp

Example

This command displays the route information of ***2001:10:1:0::102/64*** in data-block format.

```
switch# show ipv6 bgp 2001:10:1:0::102/64
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
BGP routing table entry for 2001:10:1::/64
 Paths: 2 available
  Local
    - from - (10.0.0.102)
      Origin IGP, metric 1, localpref 0, IGP metric -, weight -, received
 00:16:27 ago, valid, local, best,
redistributed (Connected)
      Rx SAFI: Unicast
  64496
    2001:10:1::100 from 2001:10:1::100 (10.0.0.100)
      Origin INCOMPLETE, metric 42, localpref 100, IGP metric 1, weight 0,
 received 00:10:09 ago, valid,
external
      Rx SAFI: Unicast switch#
```

15.5.5.119 show ipv6 bgp match community

The **show ipv6 bgp match community** command displays IPv6 Border Gateway Protocol (BGP) routing-table entries, filtered by community.

Command Mode

EXEC

Command Syntax

show ipv6 bgp match community [**COMM_1 ... COMM_n**][**MATCH_TYPE**][**INFO**][**VRF_INSTANCE**]

Parameters

- **COMM_x** community number or name, as specified in the route map that sets the community-list number. Options include:
  - ***aa:nn*** AS and network number, separated by colon. Each value ranges from ***1*** to ***4294967295***.
  - ***comm_num*** community number. Values range from *1* to ***4294967040***.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- **MATCH_TYPE** routes are filtered based on their communities. Options include:
  - ***no parameter*** routes must match at least one community in the list.
  - **exact** route must match all communities and include no other communities.
- **INFO** type of information the command displays. Options include:
  - ***no parameter*** displays table of the routing entry-line items.
  - **detail** displays data block for each routing-table entry.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing-table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

This command displays information in data-block format for each routing-table entry with community ***655:23590***.

```
switch(config)# show ipv6 bgp match community 655:23590 detail
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
BGP routing table entry for 2001:10:100:1::/64
 Paths: 1 available
  64496 64497 65536
    2001:10:1::100 from 2001:10:1::100 (10.0.0.100)
      Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received
 01:09:29 ago, valid, external, best
      Community: 655:23590 64496:1000
      Rx SAFI: Unicast
switch(config)#
```

15.5.5.120 show ipv6 bgp peers

The **show ipv6 bgp peers** command displays IPv6 Border Gateway Protocol (BGP) and TCP session data for a specified neighbor. Command displays data for all neighbors if an address is not included.

Command Mode

EXEC

Command Syntax

show ipv6 bgp peers [**NEIGHBOR_ADDR**] [**VRF_INSTANCE**]

Parameters

- **NEIGHBOR_ADDR** location of the neighbors. Options include:
  - ***no parameter*** command displays information for all neighbors.
  - ***ipv6_addr*** command displays information for the specified neighbor.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for the context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for the default VRF.

Related Command

show ip bgp peer-group

Example

This command displays information for the neighbor at ***2001:10:1:0::100***.

```
switch# show ipv6 bgp peers 2001:10:1:0::100
BGP neighbor is 2001:10:1::100, remote AS 64496, external link
  BGP version 4, remote router ID 10.0.0.100, VRF default
  Inherits configuration from and member of peer-group EXTERNAL
  Negotiated BGP version 4
  Member of update group 3
  Last read 00:00:01, last write 00:00:01
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Connect timer is inactive
  Idle-restart timer is inactive
  BGP state is Established, up for 00:12:01
```

```
  Number of transitions to established: 1
  Last state was OpenConfirm
  Last event was RecvKeepAlive
  Neighbor Capabilities:
    Multiprotocol IPv6 Unicast: advertised and received and negotiated
    Four Octet ASN: advertised and received and negotiated
    Route Refresh: advertised and received and negotiated
    Send End-of-RIB messages: advertised and received and negotiated
    Additional-paths recv capability:
      IPv6 Unicast: advertised
    Additional-paths send capability:
      IPv6 Unicast: received
  Restart timer is inactive
  End of rib timer is inactive
  Message Statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                4          5
    Keepalives:            14         14
    Route-Refresh:          0          0
    Total messages:        19         20
  Prefix Statistics:
                         Sent       Rcvd
    IPv4 Unicast:           0          0
    IPv6 Unicast:           6          4
    IPv4 SR-TE:             0          0
    IPv6 SR-TE:             0          0
  Inbound updates dropped by reason:
    AS path loop detection: 0
    Enforced First AS: 0
    Originator ID matches local router ID: 0
    Nexthop matches local IP address: 0
    Unexpected IPv6 nexthop for IPv4 routes: 0
    Nexthop invalid for single hop eBGP: 0
  Inbound updates with attribute errors:
    Resulting in removal of all paths in update (treat-as-withdraw): 0
    Resulting in AFI/SAFI disable: 0
    Resulting in attribute ignore: 0
  Inbound paths dropped by reason:
    IPv4 labeled-unicast NLRIs dropped due to excessive labels: 0
    IPv6 labeled-unicast NLRIs dropped due to excessive labels: 0
  Outbound paths dropped by reason:
    IPv4 local address not available: 0
    IPv6 local address not available: 0
Local AS is 64500, local router ID 10.0.0.102
TTL is 1
Local TCP address is 2001:10:1::102, local port is 45983
Remote TCP address is 2001:10:1::100, remote port is 179
Auto-Local-Addr is disabled
TCP Socket Information:
  TCP state is ESTABLISHED
  Recv-Q: 0/32768
  Send-Q: 0/32768
  Outgoing Maximum Segment Size (MSS): 1428
  Total Number of TCP retransmissions: 0
  Options:
    Timestamps enabled: yes
    Selective Acknowledgments enabled: yes
    Window Scale enabled: yes
    Explicit Congestion Notification (ECN) enabled: no
  Socket Statistics:
```

```
    Window Scale (wscale): 9,9
    Retransmission Timeout (rto): 204.0ms
    Round-trip Time (rtt/rtvar): 1.4ms/2.7ms
    Delayed Ack Timeout (ato): 40.0ms
    Congestion Window (cwnd): 10
    TCP Throughput: 80.00 Mbps
    Advertised Recv Window (rcv_space): 28800 switch#
```

15.5.5.121 show ipv6 bgp peers (route type)

The **show ipv6 bgp peers (route type)** command displays information about the routes either advertised to or received from a specified IPv6 BGP neighbor. The **show ipv6 bgp peers (route type) community** command displays the same information for routes filtered by communities. Commands that do not include a route type revert to the **show ipv6 bgp peers** command.

The output format depends on the selected **FILTER** parameter:

- **data-block format** displays comprehensive information for each specified route.
- **tabular format** displays routing table entries in tabular format for the specified IP addresses.

Output produced by the **longer-prefixes** option includes the specified route and all more specific routes.

Command Mode

EXEC

Command Syntax

show ipv6 bgp peers ***neighbor_addr*** **HOPDIRECT** [**FILTER**] [**VRF_INSTANCE**]

show ipv6 bgp peers ***neighbor_addr*** [**ROUTE_TYPE**] **HOPDIRECT** [detail]

Parameters

- ***neighbor_addr*** location of the neighbor.
- **ROUTE_TYPE** filters route on route type. Options include:
  - **ipv4 unicast** displays IPv4 unicast routes.
  - **ipv6 unicast** displays IPv6 unicast routes.
- **HOPDIRECT** filters route on the basis of direction from neighbor. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- **FILTER** routing table entries that the command displays. Options include:
  - ***no parameter*** displays all routing table entries in tabular format.
  - **detail** displays all routing table entries in data-block format.
  - ***ipv6_addr*** displays the IPv6 host address in data-block format.
  - ***ipv6_prefix*** displays the route information of specified IPv6 prefix in data-block format. Additional option:
    - **longer-prefixes** displays the route information of IPv4 prefix in data-block format.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Related Commands

show ipv6 bgp peers (route type) community

Example

This command displays information of all routes advertised to the neighbor at ***2001:10:1:0::100***.

```
switch# show ipv6 bgp peers 2001:10:1:0::100 advertised-routes
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

         Network                Next Hop            Metric  LocPref Weight  Path
 * >     2001:10:1::/64         2001:10:1::102        -       -       -       64500 i
 * >     2001:10:2::/64         2001:10:1::102        -       -       -       64500 i
 * >     2001:10:3::/64         2001:10:1::102        -       -       -       64500 i
 * >     2001:10:101::/64       2001:10:1::102        -       -       -       64500 64510 i
 * >     2001:10:101:1::/64     2001:10:1::102        -       -       -       64500 64510 i
 * >     2001:10:101:2::/64     2001:10:1::102        -       -       -       64500 64510 i
 switch#
```

15.5.5.122 show ipv6 bgp peers (route type) community

The **`show ipv6 bgp peers (route type) community`** command displays information about the routes either advertised to or received from a specified IPv6 BGP neighbor. The routes are filtered by community.

The **`show ipv6 bgp peers (route type)`** command displays the same information for routes filtered by IP addresses and prefixes.

Command Mode

EXEC

Command Syntax

`show ipv6 bgp peers` ***addr*** **RTE** **`community`** **CM_1** [**CM_2...CM_n**] [**MATCH**] [**INFO**] [**VRF_INST**]

Parameters

- ***addr*** neighbor location (IPv6 address).
- **RTE** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- **CM_x** community number or name, as specified in the route map that sets the community list number. The command must list at least one of the following community identifiers:
  - ***GSHUT*** well-known graceful shutdown community.
  - ***aa:nn*** AS and network number, separated by colon. Each value ranges from ***1*** to ***4294967295***.
  - ***comm_num*** community number. Values range from ***1*** to ***4294967040***.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- **MATCH** routes are filtered based on their communities. Options include:
  - ***no parameter*** routes must match at least one community in the list.
  - **exact** route must match all communities and include no other communities.
- **INFO** type of information the command displays. Values include:
  - ***no parameter*** displays table of the routing entry line items.
  - **detail** displays data block for each routing table entry.
- **VRF_INST** specifies VRF instances. Options include:

- ***no parameter*** displays routing table for context-active VRF.
- **vrf *vrf_name*** displays routing table for the specified VRF.
- **vrf all** displays routing table for all VRFs.
- **vrf default** displays routing table for default VRF.

Related Command

`show ipv6 bgp peers`

Example

This command lists the routes advertised to the neighbor at ***2001:10:1:0::102*** with the community ***64496:1000***.

```
switch# show ipv6 bgp peers 2001:10:1:0::102 advertised-routes community 64496:1000
BGP routing table information for VRF default
Router identifier 10.0.0.100, local AS number 64496
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

         Network                Next Hop            Metric  LocPref Weight  Path
 * >     2001:10:100:1::/64     2001:10:1::100        -       -       -       64496 64497 65536 i
 switch#
```

15.5.5.123 show ipv6 bgp peers regexp

The **`show ipv6 bgp peers regexp`** command displays information about routes (advertised or received) from a specified IPv6 neighbor that match the AS-path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

`show ipv6 bgp peers` ***addr*** **ROUTE regexp** ***as_paths*** [**VRF_INST**]

Parameters

- ***addr*** neighbor location (IPv6 address).
- **ROUTE** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***as_paths*** list of AS paths, formatted as a regular expression. Regular expressions are pattern-smatching strings that are composed of text characters and operators.
- **VRF_INST** specifies VRF instances. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Related Commands

- **`show ip bgp regexp`**
- **`show ipv6 bgp peers`**

Example

This command displays information for routes received from the neighbor at ***2001:10:1:0::100*** which include AS number ***64496*** in their AS paths.

```
switch# show ipv6 bgp peers 2001:10:1:0::100 received-routes regex 64496
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

         Network                 Next Hop             Metric  LocPref Weight Path
 *       2001:10:1::/64          2001:10:1::100       42      -       -      64496 ?
 * >     2001:10:100::/64        2001:10:1::100       200     -       -      64496 i
 * >     2001:10:100:1::/64      2001:10:1::100       -       -       -      64496 64497 65536 i
 * >     2001:10:100:2::/64      2001:10:1::100       42      -       -      64496 ?
switch#
```

15.5.5.124 show ipv6 bgp regexp

The **`show ipv6 bgp regexp`** command displays Border Gateway Protocol (BGP) IPv6 routing-table entries that match the AS-path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

`show ipv6 bgp regexp` ***as_paths*** [**VRF_INSTANCE**]

Parameters

- ***as_paths*** slist of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- **VRF_INSTANCE** specifies the VRF instance of the BGP routing table to be displayed. Options include:
  - ***no parameter*** displays routing table for context-active VRF.
  - **vrf *vrf_name*** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Related Command

`show ip bgp regexp`

Examples

This command displays information about the BGP IPv6 routes in the context-active VRF that pass through AS ***64496***.

```
switch# show ipv6 bgp regex _64496_
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e -
 ECMP
                    S - Stale, c - Contributing to ECMP,  b - backup, L = labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST -Cluster List, LL Nexthop - Link Local Nexthop

        Network                Next Hop              Metric  LocPref Weight  Path
 *       2001:10:1::/64      2001:10:1::100   42      100     0       64496 ?
 * >     2001:10:100::/64    2001:10:1::100   200     100     0       64496 i
 * >     2001:10:100:1::/64  2001:10:1::100   0       100     0       64496 64497 65536 i
 * >     2001:10:100:2::/64  2001:10:1::100   42      100     0       64496 ?
switch#
```

15.5.5.125 show ipv6 bgp summary

The **`show ipv6 bgp summary`** command displays the summary of all IPv4 and IPv6 BGP neighbors based on Address Family Identifier (AFI) and Subsequent Address Family Identifier (SAFI) negotiations where AFI is “IPv6” and SAFI is “Unicast” information.

Command Mode

EXEC

Command Syntax

`show ipv6 bgp summary` [**VRF_INSTANCE**]

Parameters

VRF_INSTANCE specifies VRF instances. Options include:

- ***no parameter*** displays routing table for context-active VRF.
- **vrf *vrf_name*** displays routing table for the specified VRF.
- **vrf all** displays routing table for all VRFs.
- **vrf default** displays routing table for default VRF.

Display Values

Header Row

- **`BGP router identifier`**: the router identifier; sloopback address or highest IP address.
- **`Local AS number`**: AS number assigned to switch.

Neighbor Table Columns

- **(First) `Neighbor`**: neighbor’s IP address.
- **(Second) `V`**: BGP version number.
- **(Third) `AS`**: neighbor’s AS number.
- **(Fourth) `MsgRcvd`**: messages received from the neighbor.
- **(Fifth) `MsgSent`**: messages sent to neighbor.
- **(Sixth) `InQ`**: messages queued from neighbor.
- **(Seventh) `OutQ`**: messages queued to send neighbor.
- **(Eighth) `Up/Down`**: period the BGP session has been **Established**, or its current status.
- **(Ninth) `State`**: state of the BGP session and the number of routes received from a neighbor.
- **(Tenth) `PfxRcd`**: the count of prefixes received by BGP per neighbor.
- **(Eleventh) `PfxAcc`**: the count of prefixes added to the BGP RIB among all received prefixes.

Related Command

`show ip bgp summary`

Example

This command displays the status of the switch’s BGP connections.

```
switch# show ipv6 bgp summary
BGP summary information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State   PfxRcd PfxAcc
  2001:10:1::100   4  64496             37        36    0    0 00:29:33 Estab   4      4
  2001:10:2::101   4  64510             35        38    0    0 00:29:37 Estab   4      4 switch#
```

15.5.5.126 show peer-filter

The **`show peer-filter`** command displays the definition of a peer filter.

Command Mode

EXEC

Command Syntax

show peer-filter ***filter_name***

Parameters

filter_name name of the peer-filter group.

Example

This command displays the peer-filter group information for ***group3***.

```
switch# show peer-filter group3
peer-filter group3
   10 match as-range 65003 result accept
   20 match as-range 65007 result accept
   30 match as-range 65009 result accept switch#
```

15.5.5.127 show run|section bgp

When using the **show run** command, it displays the entire running configuration. Sometimes this is unnecessary, so to target your output you can use the **show run|section bgp** command which will display only the BGP section.

Command Mode

bgp-router

Command Syntax

show run | section bgp [***name***]

Parameter

name name of the peer-group.

Example

Once the peer group request are completed, then run the **show run|section bgp** command to display only the BGP section of the running configuration.

```
switch(config-router-bgp)# show run|section bgp router bgp 300
switch(config-router-bgp)# neighbor interface Et1-2,4-6 peer-group PG1 remote-
as 100
switch(config-router-bgp)# neighbor interface Et3 peer-group PG2 remote-as 200
switch(config-router-bgp)# neighbor interface vlan2000-2002 peer-group PG1
 remote-as 100
```

15.5.5.128 show tunnel rib brief

The **show tunnel rib brief** command displays the preferred tunnels for various IP endpoints, optionally filtered by endpoint. Each tunnel RIB entry in the output displays the type of the tunnel (such as BGP LU) and a numerical index uniquely identifying that tunnel within the type-specific tunnel table.

Command Mode

EXEC

Command Syntax

show bgp tunnel rib brief

Example

This command displays the tunnel type and the index value.

```
switch# show tunnel rib brief
Endpoint           Tunnel Type        Indexes
-----------------  -----------------  -------
10.1.1.0/32        BGP LU             2
11.1.1.0/32        BGP LU             1, 3 switch#
```

15.5.5.129 shutdown (BGP)

The **shutdown** command disables BGP on the switch without modifying the BGP configuration.

The **no shutdown** and **default shutdown** commands enable the BGP instance by removing the **shutdown** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Examples

- These commands disable BGP on the switch.

```
switch(config)# router bgp 9
switch(config-router-bgp)# shutdown
switch(config-router-bgp)#
```

- These commands enable BGP on the switch.

```
switch(config)# router bgp 9
switch(config-router-bgp)# no shutdown
switch(config-router-bgp)#
```

15.5.5.130 timers bgp

The **timers bgp** command configures the BGP keepalive and hold times.Timer settings apply to each peer connection. The **neighbor timers** command configures the times on a specified peer connection.

- **Keepalive time**: period between the transmission of consecutive keepalive messages.
- **Hold time**: period the switch waits for a keepalive or UPDATE message before it disables peering.

The hold time must be at least ***3*** seconds and should be three times longer than the keepalive setting.

The **no timers bgp** and **default timers bgp** commands return the time settings to their default values by removing the **timers bgp** command from ***running-config***. The default values are:

- **keepalive**: ***60*** seconds.
- **hold time**: ***180*** seconds.

Command Mode

Router-BGP Configuration

Command Syntax

timers bgp ***keep_alive hold_time***

no timers bgp

default timers bgp

Parameters

- ***keep_alive*** keepalive period, in seconds. Values include:
  - ***0*** keepalive messages are not sent.
  - ***1*** to ***3600*** keepalive time (seconds).
- ***hold_time*** hold time. Values include:
  - ***0*** peering is not disabled by timeout expiry; keepalive packets are not sent.
  - ***3*** to ***7200*** hold time (seconds).

Example

This command sets the keepalive time to ***30*** seconds and the hold time to ***90*** seconds.

```
switch(config)# router bgp 9
switch(config-router-bgp)# timers bgp 30 90
switch(config-router-bgp)#
```

15.5.5.131 update wait-for-convergence

The **update wait-for-convergence** command disables FIB updates and route advertisement when the BGP instance is initiated until the BGP convergence state is reached.

The **no update wait-for-convergence** command allows FIB updates and route advertisement irrespective of the BGP convergence state.

Command Mode

Router-BGP Configuration

Command Syntax

update wait-for-convergence

no update wait-for-convergence

default update wait-for-convergence

Related Commands

- **clear ip bgp** removes learned BGP routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required.
- **bgp convergence slow-peer time** configures the BGP convergence idle peer timeout value.
- **bgp convergence time** configures the BGP convergence timeout value.
- **show bgp convergence** displays information about the BGP convergence state; and other statistics about the BGP instance in either the specified VRF or all VRFs.

Guidelines

The initiation of BGP instance includes the following scenarios:

- the BGP instance starts for the first time after a switch is reloaded.
- the BGP instance restarts.
- all sessions are cleared by using the **clear ip bgp *** command.

Configuration changes made by using this command are effective from the next initiation of a BGP instance.

Example

This command disables FIB updates and route advertisement when the BGP instance is initiated until the BGP convergence state is reached.

```
switch(config)# router bgp 9
switch(config-router-bgp)# update wait-for-convergence
switch(config-router-bgp)#
```

15.5.5.132 vrf

The **vrf** command places the switch in BGP VRF configuration mode for the specified VRF. Commands issued in this mode will override global BGP configuration for the specified VRF.

Command Mode

Router-BGP Configuration

Command Syntax

vrf ***vrf_instance***

Parameters

vrf_instance VRF to be configured.

Example

These commands place the switch in BGP VRF configuration mode for VRF ***purple***.

```
switch(config)# router bgp 9
switch(config-router-bgp)# vrf purple
switch(config-router-bgp-vrf-purple)#
```

15.6 Maintenance Mode

This section describes configuration for performing maintenance of switch elements.

This section contains these topics:

- Overview
- Maintenance Mode Elements
- Maintenance Mode Features
- Maintenance Mode Configuration
- Maintenance Mode Commands

15.6.1 Overview

Using maintenance mode, you can perform several maintenance activities such as:

- EOS image upgrade.
- Initial configuration or reconfiguration of a production system.
- Replacement of hardware.
- Changing linecards or transceiver modules.
- Replace, reattach, and reroute cables.

Maintenance mode uses BGP to divert traffic away from the switch on which the maintenance tasks need to be performed, minimizing traffic impact. You can set the traffic thresholds and time limits at which the switch, or parts of the switch, is considered to be available for maintenance tasks.

Maintenance mode can be activated on a switch at boot-up or during operation. The mode provides the following benefits:

- Rerouting of traffic when the mode is activated during operation and other routes are present.
- Replacement of hardware in modular systems or systems with redundant hardware.

The switch is placed into maintenance mode, serviced, and then returned to normal operation.

15.6.2 Maintenance Mode Elements

Maintenance mode elements include Units, Groups of Interfaces and BGP Peers, and Profiles. Arista Network switches provide maintenance mode operations performed on a fundamental, configurable element, referred to as a Unit. Maintenance mode will quiesce a unit, which places the unit into maintenance mode by gracefully transitioning traffic away from it.

The most common maintenance mode operations such as removing from service an entire switch system or individual components of the switch, including a single linecard, interface, or BGP peer, can be achieved using minimal configuration.

15.6.2.1 Units

Units are configurable maintenance mode elements that comprise a collection of various groups. In addition, units contain policies which decide whether the member groups should be put into maintenance mode automatically upon boot. Built-in units are configured by default, such as the System unit representing the entire system. All maintenance mode operations are executed at the unit level.

An interface, interface range, and BGP peer (or peer-group) can be directly put under maintenance.

15.6.2.1.1 Built-in Units

There are various built-in units such as **System** and **Linecard<n>**. Fixed systems contain only one built-in unit called **System**, which comprises the interface group containing all Ethernet interfaces and sub-interfaces; and BGP groups per VRF containing all the peers in the respective VRF.

Modular Systems have both **System** and **Linecard<n>** units. **Linecard<n>** units are present for each linecard which comprises the **Linecard<n>** groups containing all Ethernet interfaces and sub-interfaces of that linecard.

15.6.2.1.2 User-configured Units

You can also configure customized units containing user-defined groups and policies as shown in the following example. A custom group called ***BG1*** with a custom interface ***IG1*** and a unit profile ***UP1*** is created. The show command displays the details.

```
switch(config)# maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group bgp BG1
switch(config-unit-UNIT1)# group interface IG1
switch(config-unit-UNIT1)# profile unit UP1
switch(config-unit-UNIT1)# exit
switch(config-maintenance)# show maintenance units
Unit Name: System
Origin: Built-in
Status: Not Under Maintenance
Unit Profile: Default
Time Since Last State Change: never
Bgp Groups:
AllBgpNeighborVrf-default
Interface Groups:
AllEthernetInterface
Unit Name: UNIT1
Origin: User Configured
Status: Under Maintenance
```

```
Unit Profile: UP1
Time Since Last State Change: 0:00:08 ago
Bgp Groups:
BG1
Interface Groups:
IG1
```

15.6.2.2 Groups of Interfaces and BGP Peers

Maintenance mode group types include the groups for interfaces and BGP peers. Groups are identified by a group name unique to a particular group type.

By default, several built-in groups are available on the device such as **linecard** groups containing physical interfaces.

15.6.2.2.1 Built-in Groups

There are several built-in groups such as **AllEthernetInterface**, **Linecard1**, **Linecard2**, etc., **AllBgpNeighborVrf-<vrf_name>**. **AllEthernetInterface** is the built-in interface group which contains all physical Ethernet interfaces and sub-interfaces on the switch, and is a part of **System** unit. Whereas on modulars **Linercard1**, **Linecard2**, etc., are the built-in groups which contain respective linecard interfaces and sub-interfaces; and are part of the **Linecard1** and **Linecard2** units respectively. **AllBgpNeighborVrf-<vrf_name>** is the built-in BGP group which contains all the BGP peers in that particular VRF.

15.6.2.2.2 User-defined Groups

The following set of commands sets up a custom group (***IG1***) of interfaces, which includes physical ports, port-channels and SVIs.

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# interface Ethernet1
switch(config-group-if-IG1)# interface Port-Channel1,20
switch(config-group-if-IG1)# interface Vlan1-20
switch(config-group-if-IG1)# exit
switch(config)#
```



Note: User-defined interface groups do not contain sub-interfaces.

The following set of commands sets up a custom group (***BG1***) of BGP peers.

```
switch(config)# group bgp BG1
switch(config-group-bgp-BG1)# neighbor 10.0.0.1
switch(config-group-bgp-BG1)# neighbor BGP_PG1
switch(config-group-bgp-BG1)# vrf vrf1
switch(config-group-bgp-BG1)# exit
switch(config)#
```



Note: BGP groups are specific to VRF.

15.6.2.3 Profiles

Profiles are configurable maintenance mode elements that define policies for related software or hardware components to carry out maintenance mode operations.

15.6.2.3.1 Default Profiles

Default profiles are the built-in policies which are applied to groups interface/BGP and unit.

The default profile is used in the absence of an explicit interface/BGP profile associated with the group, or explicit unit profile associated with the unit.

BGP Profile

Default BGP profile has route-map with set clausesset community GSHUT additive and set local-preference 0.

```
switch(config-maintenance)# show maintenance profile bgp default
Bgp Profile: Default
Initiator route-map: SystemGenerated
route-map SystemGenerated permit 10
Description:
description System generated initiator route-map
Match clauses:
SubRouteMap:
Set clauses:
set local-preference 0
set community GSHUT additive
```

Interface Profile

Default interface profile has rate-monitoring load-interval set to ***60*** seconds, threshold set to ***100*** kbps, and shutdown disabled as shown. The max-delay parameter is set to ***300*** seconds but is not enabled.

```
switch(config-maintenance)# show maintenance profile interface default
Interface Profile: Default
Rate Monitoring:
load-interval: 60 seconds
threshold (in/out): 100 kbps
shutdown:
enabled: no
max-delay: 300 seconds
```

Unit Profile

Default unit profile has on-boot setting disabled.

```
switch(config-maintenance)# show maintenance profiles unit default
Unit Profile: Default
On-boot:
enabled: no
duration: 300 seconds
```

15.6.2.3.2 User-defined Profiles

You can define your own profiles which can be associated to groups or set as default profiles.

Interface Profile: The following set of commands sets up an Interface Profile (***IP1***) with load interval set to ***10*** seconds, rate-monitoring threshold set to ***100kbps*** and the maximum delay for shutting down the interface set to ***100*** seconds. The interface will be shutdown with cause maint-down if traffic does not drain below the threshold even after the specified maximum delay period of ***100*** seconds.

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1
switch(config-profile-intf-IP1)# rate-monitoring load-interval 10
switch(config-profile-intf-IP1)# rate-monitoring threshold 100
switch(config-profile-intf-IP1)# shutdown max-delay 100
switch(config-profile-intf-IP1)# exit
```

```
switch(config-maintenance)#
```

An interface profile can be associated to only interface groups using the following set of commands.

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# maintenance profile interface IP1
switch(config-group-if-IG1)# exit
switch(config)#
```

You can set the interface profile as the default interface profile using the following set of commands.

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1 default
switch(config-maintenance)# exit
switch(config)#
```

BGP Profile: The following set of commands sets up a BGP profile (***BP1***) with initiator route-map called RM which will be applied for both inbound and outbound directions.

```
switch(config)# maintenance
switch(config-maintenance)# profile bgp BP1
switch(config-profile-bgp-BP1)# initiator route-map RM inout
switch(config-profile-bgp-BP1)# exit
switch(config-maintenance)#
```

A BGP profile can be associated to both interface and bgp groups using the following commands.

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# maintenance profile bgp BP1
switch(config-group-if-IG1)# exit
switch(config)# group bgp BG1
switch(config-group-bgp-BG1)# maintenance profile bgp BP1
switch(config-group-bgp-BG1)# exit
switch(config)#
```

You can set the bgp profile as the default bgp profile using the following set of commands.

```
switch(config)# maintenance
switch(config-maintenance)# profile bgp BP1 default
switch(config-maintenance)# exit
switch(config)#
```

Unit Profile: The following set of commands sets up a Unit profile (***UP1***) with on-boot duration of ***300*** seconds. The unit will enter into maintenance mode at boot-up and exit maintenance mode at the end of ***5*** minutes (***300sec***) after boot-up.

```
switch(config-maintenance)# profile unit UP1
switch(config-profile-unit-UP1)# on-boot duration 300
switch(config-profile-unit-UP1)# exit
switch(config-maintenance)#
```

A Unit profile can be associated to a Unit using the following commands.

```
switch(config)# maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# profile unit UP1
switch(config-unit-UNIT1)# exit
switch(config-maintenance)#
```

You can set the Unit profile as the default Unit profile using the following set of commands.

```
switch(config)# maintenance
switch(config-maintenance)# profile unit UP1 default
switch(config-maintenance)# exit
switch(config)#
```

15.6.3 Maintenance Mode Features

Arista Network switches provide maintenance mode features including rate monitoring, BGP maintenance route map, on-boot maintenance, and EventMgr integration.

15.6.3.1 Rate Monitoring

Rate monitoring provides a mechanism to monitor traffic on interfaces identified for maintenance. You can set the traffic threshold and a time limit for the interface to be shutdown for maintenance tasks.

A shutdown parameter can be configured in the interface profile that signals the interface to be shutdown after it has entered maintenance mode.

The max-delay parameter specifies the maximum number of seconds to allow for traffic to dissipate from the interface before the interface is shutdown. The default interface profile settings are shown in the output of the **`show maintenance profile interface default`** command.



Note: The exclusive rate monitoring of sub-interfaces is not supported. Sub-interfaces inherit the interface profile from its parent interface. In case of multiple sub-interfaces configured for single parent interface, rate monitoring of parent interface include aggregate values of all respective sub-interfaces.

15.6.3.2 BGP Maintenance Route Map

Route-maps are used within a BGP maintenance profile to tag the inbound and outbound routes in order to direct traffic away from the unit.

The default profile tags the inbound and outbound routes with the global shutdown community. Other methods can be configured under the route-map such as alternate communities, or by using **AS_PATH** prepend operations.

15.6.3.3 On-boot Maintenance

There are two ways of placing a unit in maintenance mode on switch boot-up:

- The unit is placed into maintenance mode prior to the switch reboot, and the running-config is saved prior to switch boot-up.
- The on-boot property in the unit maintenance profile specifies that the unit will be placed into maintenance mode as part of boot-up, and remains so for the specified duration.



Note: The duration value in the on-boot unit maintenance profile starts as soon as the unit is put into maintenance mode on boot-up.

15.6.3.4 Single Event Upset handling

All electronic devices are subject to interference from cosmic radiation. Arista products use a combination of hardware and software to automatically detect and correct the results of this interference. For instance, many chip memories contain parity or Error Correcting Code (ECC) bits. However, Single Event Handling (SEU) is a random event, and following configuration determines the handling behavior.

```
switch(config)# platform sand seu
switch(config-sand-seu)#
```

The system, by default corrects the first instance of an ECC or parity event without any logging. If a further error occurs within a 4 hour time window, related to the first or not, a log message will be emitted.

The default 4 hours logging window can be changed as following. For example, a second SEU is detected within ***3*** hours of a prior SEU.

```
switch(config-sand-seu)# log window 10800 seconds
```

Static memories are used by hardware to hold configuration to determine switching behaviour. When SEUs occur, repairs are made automatically. The following command disables automatic repair by a specific agent.

```
switch(config-sand-seu)# repair table static manager SandFap disabled
```

The following command disables automatic repair by a specific memories, overriding any specific configurations.

```
switch(config-sand-seu)# repair table static disabled
```

Note: Repair of static tables should only be disabled in consultation with the Support team.

The following command disables automatic repair of fabric chip memories on the modular or fixed systems which use fabric chip.

```
switch(config-sand-seu)# repair table fabric manager SandFabric disabled
switch(config-sand-seu)# repair table fabric disabled
```

The following command disables dynamic memories globally. Dynamic memories are internal memories utilized by hardware to hold transient data, such as packet header encapsulations.

```
switch(config-sand-seu)# repair table dynamic disabled
```

The following command disables SEU resets without affecting other dynamic table repairs.

```
switch(config-sand-seu)# repair table dynamic action reset full disabled
```

The following command configures the SEU to ***12*** hours, by default the minimum interval between SEUs is ***24*** hours.

```
switch(config-sand-seu)#repair action reset full interval 43200 seconds
```

Show Command

SEU events generate interrupts, it can be seen along with all other interrupts.

```
switch# show platform fap interrupts
 Jericho0
 ---------------------------------------------------------------------------------------
 | Interrupt Bit                | Count | First Occurrence    | Last Occurrence     |
 ---------------------------------------------------------------------------------------
 | ...                          |   ... |                     |                     |
 | CFC_ECC_Ecc_2bErrInt[0]      |     2 | 2020-10-15 04:27:59 | 2020-10-15 04:31:41 |
 | ...                          |   ... |                     |                     |
 ---------------------------------------------------------------------------------------
```

All SEU interrupt names take the form **<block>_ECC_<type>Int**. The block indicates the part of a switch chip affected. The type can be one of the following:

- **Ecc_1bErr** - Single bit error in ECC protected memory, corrected automatically in hardware.
- **Ecc_2bErr** - Two bit error in ECC protected memory detected, requires software correction.

- **ParityErr** - Single bit error in parity protected memory detected, requires software correction.

Single bit ECC errors do not affect correct operation of the switch. Two bit ECC and parity errors can disrupt correct operation, for example, by dropping one or more packets, or by mis-forwarding packets. The exact effect depends on the memory and location affected by the SEU.

15.6.4 Maintenance Mode Configuration

You can configure maintenance mode for the entire device, specific linecards, or any other Unit. You can set up configuration for maintenance mode for the device at boot-up or while it is running.



Note: Explicit maintenance of sub-interfaces is not supported. Sub-interfaces are put into maintenance implicitly in case of built-in unit maintenance and interface maintenance but not in case of user-configured units.

15.6.4.1 Unit Configuration

Arista Network switches provide the ability to place the switch in maintenance mode, and configuration options for groups, profiles, associating profiles with groups, units, and maintenance mode operations. **System** is a predefined (built-in) unit on all switches. Built-in groups include **AllEthernetInterface**, **AllBgpNeighborVRF-<vrf_name>**, and **Linecardn**. **Linecardn** can also be a built-in unit and can be differentiated depending on the command being used as shown.

- ```
  switch(config-maintenance)# unit Linecardn
  ```
- ```
  switch(config)# group interface Linecardn
  ```

Built-in unit **System** comprises the following groups:

- **AllEthernetInterface** - a built-in interface group which contains all physical Ethernet interfaces on the switch on a fixed system.
- **Linecardn** - a built-in interface group which contains all interfaces for the linecard numbered n for modular systems.
- **AllBgpNeighborVRF-<vrf_name>** - a built-in BGP group which contains all the BGP peers in the named VRF.

For each Linecard n, there is a built-in unit which consists of all the **Linecardn** groups.

By default, the default interface and BGP profiles are applied to the built-in interface and BGP groups and the default built-in unit profile is applied to the built-in unit. You can also configure your own profiles and choose a default.

In the following example, traffic is flowing through multiple switches in the spine to and from one switch to another, when you elect to put one of the Units (entire switch or parts thereof) in the spine switch in maintenance mode. The traffic is then gracefully steered away from the Unit, provided other paths are available. Traffic will continue to flow through the Unit placed into maintenance mode, if no other path is available.

Example

Note: The illustration shows an entire switch as the Unit. You can replace switch with Linecardn or another relevant Unit as appropriate.

Figure 15-18: Traffic flow pattern between TOR and Core Before Maintenance

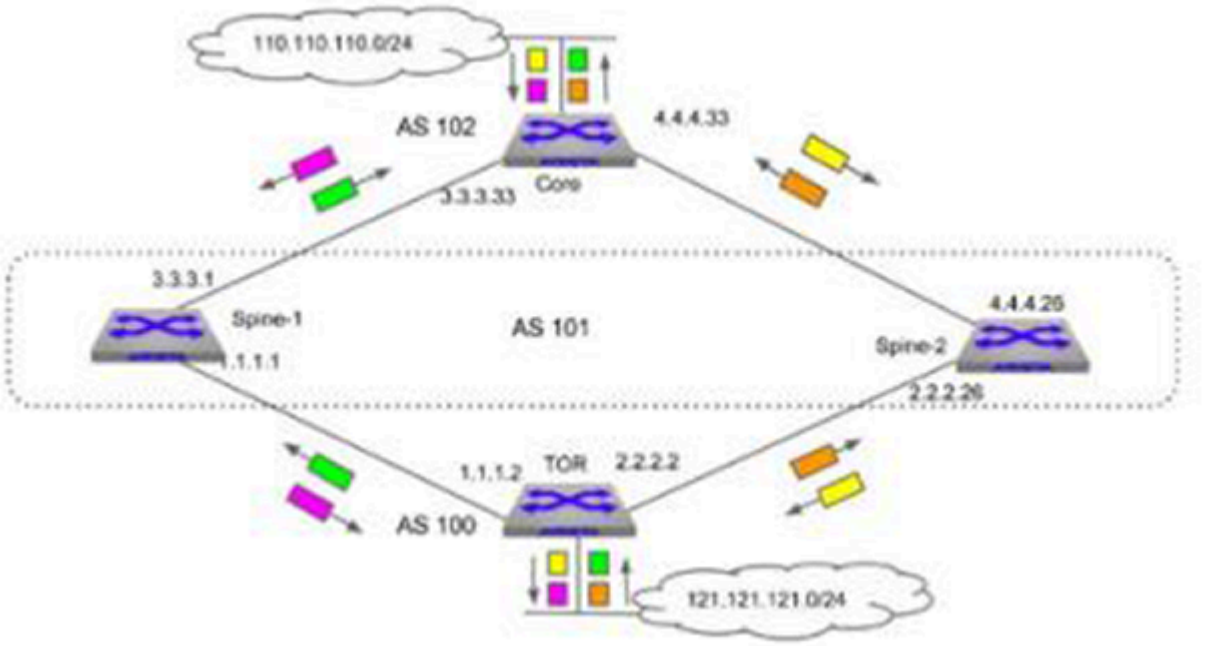
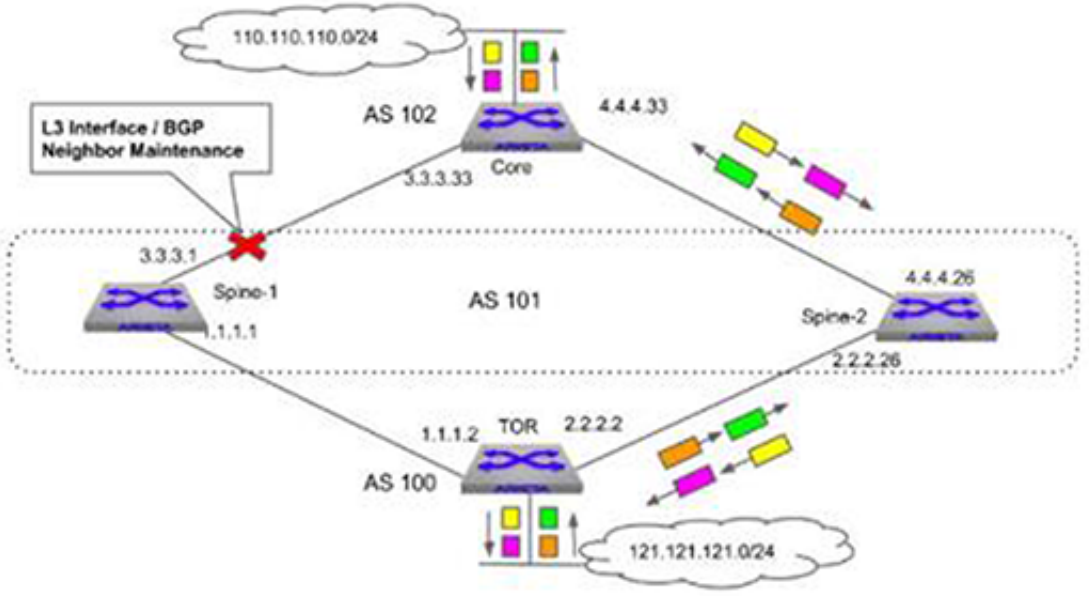


Figure 15-19: Traffic flow pattern between TOR and Core After unit on Spine-1 is put into Maintenance



You can see the status of the Unit (System) using the **`show maintenance units System`** command for the example above before the system is placed into maintenance mode. If the device being placed into maintenance mode is modular and the Unit is a linecard, replace the argument **System** with **Linecardn** to see the status of the Unit (Linecardn).

```
switch(config)# show maintenance units System
Unit Name: System
```

```
   Origin: Built-in
   Status: Not Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: never
   Bgp Groups:
     AllBgpNeighborVrf-default
   Interface Groups:
     AllEthernetInterface
```

You can then place the Unit (System) into maintenance mode and recheck the status using the sequence of commands shown.

```
switch(config-maintenance)# unit System
switch(config-builtin-unit-System)# quiesce
switch(config-builtin-unit-System)# exit
switch(config-maintenance)# show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold


Unit Name              Status                   Time since last change    Flags
---------------------- ------------------------ ------------------------- -----
System                 Under Maintenance              0:02:03 ago

switch(config-maintenance)# show ip bgp summary
BGP summary information for VRF default
Router identifier 1.1.1.1, local AS number 101
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State
PfxRcd PfxAcc
m 1.1.1.2          4  100              24        17    0    0 00:00:40 Estab  5      5
m 3.3.3.33         4  102              15        16    0    0 00:06:23 Estab  1      1
```

Note: The **o** flag is shown for on-boot maintenance in the **show maintenance** command and the **m** neighbor status flag in the **show ip bgp summary** command indicates that the peer is in maintenance mode.

15.6.4.2 On-boot Maintenance Mode Configuration

To configure on-boot maintenance, you can use one of two methods:

- Use **quiesce config**, or
- Use on-boot profile

15.6.4.2.1 Using quiesce config

You must perform the following tasks to place the Unit in maintenance mode on boot-up using the **quiesce** command.

1. Place the unit into maintenance mode prior to switch reboot using the following commands.

```
switch(config)# maintenance
switch(config-maintenance)# unit System
switch(config-unit-System)# quiesce
switch(config-unit-System)# exit
switch(config-maintenance)# show maintenance

Flags:
o - On-boot maintenance
v - Violating traffic threshold
Unit Name Status Time since last change Flags
--------- ------ ---------------------- -----
System Under Maintenance 00:01:10 ago
```

2. Save the ***running-config*** using the following command.

```
switch(config)# copy running-config startup-config
Copy completed successfully
switch(config)#
```

3. Reload the device.

```
switch(config)# reload
Proceed with reload? [Confirm] Yes
Connection to switch closed.
```

4. After the device comes up, you must execute the **no quiesce** command for the Unit to come out of maintenance mode. You can check the status of the device after it comes up using the **show maintenance** command.

```
switch# show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold
Unit Name Status Time since last change Flags
--------- ------ ---------------------- -----
System Under Maintenance 00:03:10 ago
```

15.6.4.2.2 Using on-boot profile

The on-boot property in the Unit maintenance profile specifies that the Unit will be placed into maintenance mode as part of boot-up for the specified duration. You must perform the following tasks to use this method.

1. Check to see if the on-boot maintenance mode is enabled using the **show maintenance profiles unit default**.

```
switch# show maintenance profiles unit default
Unit Profile: Default
On-boot:
enabled: no
duration: 300 seconds
```

2. Configure an on-boot profile with on-boot enabled and a duration specified. Make this the default Unit profile. The following code example shows the creation of an on-boot duration of ***300*** seconds in the profile unit ***UP1***.

```
switch(config)# maintenance
switch(config-maintenance)# profile unit UP1
switch(config-profile-unit-UP1)# on-boot duration 300
switch(config-profile-unit-UP1)# exit
switch(config-maintenance)# profile unit UP1 default
switch(config-maintenance)# show maintenance profiles unit default
Unit Profile: UP1
On-boot:
enabled: yes
duration: 300 seconds
switch(config-maintenance)#
```

3. Save the ***running-config*** and reload the device.

```
switch(config)# copy running-config startup-config
Copy completed successfully
switch(config)# reload
Connection to switch closed.
```

4. After the device comes up, execute the **show maintenance** and **show maintenance units System** commands.

```
switch(config)# show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold
Unit Name Status Time since last change Flags
--------- ------ ---------------------- -----
System Under Maintenance 00:00:08 ago o
switch(config)# show maintenance units System
Unit Name: System
Origin: Built-in
Status: Under Maintenance (on-boot)
Unit Profile: UP1
Time Since Last State Change: 0:00:16 ago
Will come out of on-boot Maintenance after 0:04:43
Interface Groups:
AllEthernetInterface
History:
2017-01-18 00:44:39 old state: 'maintenanceModeEnter' to new state:
'underMaintenance' 0:00:16 ago
2017-01-18 00:43:54 old state: 'active' to new state: 'maintenanceModeEnter'
0:01:01 ago
```

The o - flag shows that unit System is under maintenance due to on-boot profile. Also, **show maintenance units System** output shows the following - Will come out of on-boot Maintenance after ***0:04:43***, which is the time remaining of the specified duration of ***5*** minutes.

The Unit will come up in maintenance mode when the device boots up and will exit maintenance mode once the specified duration of ***300*** seconds in the default profile is completed. The BGP sessions will remain under maintenance for the duration and will resume after the specified duration is over.

15.6.4.3 Interface-level Maintenance Mode Configuration

To configure the maintenance mode at interface-level, you must perform the following tasks:

1. Configure an interface-level profile (or use a pre-configured one). The following code example creates a user-defined interface profile IP1 with a rate-monitoring load-interval of ***100*** seconds, a rate-monitoring threshold of ***500*** kbps and a maximum shutdown delay of ***100*** seconds.

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1
switch(config-maint-if-Et5)# rate-monitoring load-interval 100
switch(config-maint-if-Et5)# rate-monitoring threshold 500
switch(config-maint-if-Et5)# shutdown max-delay 100
```

2. Make the user-defined interface profile IP1 as the default interface profile.

```
switch(config-maintenance)# profile interface IP1 default
```

3. Place the interface into maintenance mode.

```
switch(config)# maintenance
switch(config-maintenance)# interface Ethernet 1
switch(config-maint-if-Et1)# quiesce
```

4. Remove the interface from maintenance mode once the service has been performed.

```
switch(config-maintenance)# interface Ethernet 1
switch(config-maint-if-Et1)# no quiesce
```

Note: If interface ***Et1*** has sub-interfaces (***Et1.1***, ***Et1.2***,...) with BGP peers on these sub-interfaces, then these sub-interfaces are also placed into maintenance mode. The **`show maintenance interface sub-interface detail`** command displays the maintenance state of sub-interfaces.

15.6.4.4 Entering Maintenance Mode

Enter configuration commands **`unit`** and **`quiesce`** using the **`maintenance profile bgp`** mode command to place the switch into maintenance mode. The following code sequence places unit foo, the ***interface 3/3***, and BGP **`1.1.1.1`** in maintenance mode.

Example

```
switch(config)# maintenance
switch(config-maintenance)# unit foo
switch(config-unit-foo)# quiesce
switch(config-unit-foo)# exit
switch(config-maintenance)# interface ethernet 3/3
switch(config-maint-if-Et3/3)# quiesce
switch(config-unit-if-Et3/3)# exit
switch(config-maintenance)# bgp 1.1.1.1
switch(config-maint-bgp-1.1.1.1)# quiesce
switch(config-maint-bgp-1.1.1.1)# exit
switch(config-maintenance)#
```

15.6.4.5 Exiting Maintenance Mode

Enter configuration commands **`unit`** and **no** **`quiesce`** using the **`maintenance profile bgp`** mode command for the switch to exit maintenance mode. The following code sequence causes unit foo, the ***interface 3/3***, and BGP ***1.1.1.1*** to exit maintenance mode.

Example

```
switch(config)# maintenance
switch(config-maintenance)# unit foo
switch(config-unit-foo)# no quiesce
switch(config-unit-foo)# exit
switch(config-maintenance)# interface ethernet 3/3
switch(config-maint-if-Et3/3)# quiesce
switch(config-unit-if-Et3/3)# exit
switch(config-maintenance)# bgp 1.1.1.1
switch(config-maint-bgp-1.1.1.1)# no quiesce
switch(config-maint-bgp-1.1.1.1)# exit
switch(config-maintenance)#
```

15.6.4.6 Configuring Event Handlers

Enter configuration options for the **`show maintenance`** command to fire at different stages while entering or exiting maintenance mode.

Example for Maintenance Mode Event Handler for all Stages

```
switch(config)# event-handler foo
switch(config-handler-foo)# trigger on-maintenance enter unit unit-foo all
switch(config-handler-foo)# action bash /mnt/flash/mm-event-handler-script
switch(config-handler-foo)# timeout 20
```

```
switch(config-handler-foo)# exit
switch(config)#
```

Note: The user is expected to configure the timeout value. This is time within which the script should complete execution and exit. If the script has not exited by the end of this period, then the following will occur:

1. Send the **SIGUSR1** signal to the script.
2. Wait for a **GRACE-PERIOD** of ***10*** seconds for the script to exit.
3. If the script does not exit even after that **GRACE-PERIOD**, then send a **SIGKILL** to the script.
4. The maintenance operation progresses to the next stage.
5. **GRACE-PERIOD** is not configurable.

```
switch(config)# event-handler bar
switch(config-handler-bar)# trigger on-maintenance exit unit unit-foo before
stage ratemon
switch(config-handler-bar)# action bash /mnt/flash/mm-event-handler-script
switch(config-handler-bar)# exit
switch(config)#
```

15.6.4.7 Configuring Groups

Enter the maintenance mode configuration options for groups with the **maintenance** and **group bgp** commands.

Example for Group Interface IG1

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# interface Ethernet1
switch(config-group-if-IG1)# interface Port-Channel1,20
switch(config-group-if-IG1)# interface Vlan1-20
switch(config-group-if-IG1)# exit
switch(config)#
```

Example for Group BGP BG1

```
switch(config)# group bgp BG1
switch(config-group-bgp-BG1)# neighbor 10.0.0.1
switch(config-group-bgp-BG1)# neighbor BGP_PG1
switch(config-group-bgp-BG1)# vrf vrf1
switch(config-group-bgp-BG1)# exit
switch(config)#
```

Note: BGP groups are specific to VRF.

15.6.4.8 Configuring Profiles

Enter the maintenance mode configuration options for profiles with the **profile interface**, **rate-monitoring threshold**, **profile bgp**, and **profile unit <profile_name>** commands.

These command examples assign a user configured profile as the **default** profile.

Example for Profile Interface IP1

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1
```

```
switch(config-profile-intf-IP1)# rate-monitoring load-interval 10
switch(config-profile-intf-IP1)# rate-monitoring threshold 100
switch(config-profile-intf-IP1)# shutdown max-delay 100
switch(config-profile-intf-IP1)# profile interface IP1 default
switch(config-profile-intf-IP1)# exit
switch(config-maintenance)#
```

Example for Profile BGP BP1

```
switch(config-maintenance)# profile bgp BP1
switch(config-profile-bgp-BP1)# initiator route-map rmap inout
switch(config-profile-bgp-BP1)# profile bgp BP1 default
switch(config-profile-bgp-BP1)# exit
switch(config-maintenance)#
```

Example for Profile Unit UP1

```
switch(config-maintenance)# profile unit UP1
switch(config-profile-unit-UP1)# on-boot duration 300
switch(config-profile-unit-UP1)# profile unit UP1 default
switch(config-profile-unit-UP1)# exit
switch(config-maintenance)#
```

15.6.4.9 Associating Profiles with Groups

Enter the maintenance mode configuration options for associating profiles with groups using the **maintenance** and **group bgp** command.

Example

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# maintenance profile bgp BP1
switch(config-group-if-IG1)# maintenance profile interface IP1
switch(config-group-if-IG1)#
```

Note: An interface/BGP profile can be associated with the interface group, and a BGP profile can be associated with the BGP group.

15.6.4.10 Configuring Units

Enter the maintenance mode configuration options for units using the **unit**, **group bgp**, and **maintenance** commands.

Example

```
switch(config)# maintenance
switch(config-maintenance)# unit foo
switch(config-unit-foo)# group bgp BG1
switch(config-unit-foo)# group interface IG1
switch(config-unit-foo)#profile unit UP1
```

15.6.5 Maintenance Mode Commands

Global Configuration Commands

- group bgp
- group interface
- maintenance

Group Configuration Commands

- interface
- maintenance profile bgp
- maintenance profile interface
- neighbor
- vrf

Maintenance Configuration Commands

- bgp <peer> [vrf <vrf_name>]
- interface intf-name
- profile bgp
- profile bgp <profile_name> default
- profile interface
- profile interface <profile_name> default
- profile unit
- profile unit <profile_name>
- profile unit <profile_name> default
- unit

Unit Configuration Commands

- group bgp <group_name>
- group interface <group_name>
- quiesce

Interface Profile Configuration Commands

- rate-monitoring load-interval
- rate-monitoring threshold
- shutdown max-delay

BGP Profile Configuration Commands

- initiator route-map <route-map-name> inout

Unit Profile Configuration Commands

- on-boot duration

EventMgr Configuration Commands

- trigger on-maintenance

Display Commands

- show interface
- show interface <intf_name> status
- show ip | ipv6 bgp
- show ip | ipv6 bgp summary [vrf <vrf_name>]
- show maintenance
- show maintenance bgp
- show maintenance bgp receiver route-map
- show maintenance debug
- show maintenance groups
- show maintenance interface
- show maintenance interface status
- show maintenance interface status quiesced
- show maintenance profiles
- show maintenance stages
- show maintenance summary
- show maintenance units

15.6.5.1 bgp <peer> [vrf <vrf_name>]

The **`bgp <peer> [vrf <vrf-name>]`** command places the switch in maintenance dynamic BGP unit configuration mode. If no VRF is specified, the BGP peer is considered to be in the DEFAULT VRF, otherwise, in the specified VRF.

The command creates the dynamic BGP unit if the specified dynamic BGP unit does not exist prior to issuing the command.

The **`no bgp <peer> [vrf <vrf_name>]`** and **`default bgp <peer> [vrf <vrf_name>]`** removes the dynamic BGP unit from ***running-config***.

Command Mode

Maintenance Configuration

Command Syntax

`bgp` ***ipv4_addr*** [vrf ***vrf_name***]

`bgp` ***ipv4_addr*** [vrf ***vrf_name***]

`bgp` ***ipv4_addr*** [vrf ***vrf_name***]

`no bgp` [***ipv4_addr*** | ***ipv6_addr*** | ***peer_group_name***][vrf ***vrf_name***]

`default bgp` [***ipv4_addr*** | ***ipv6_addr*** | ***peer_group_name***][vrf ***vrf_name***]

Parameters

- ***ipv4_addr*** BGP neighbor IPv4 address.
- ***ipv6_addr*** BGP neighbor IPv6 address.
- ***peer_group_name*** BGP peer group name.
- **vrf *vrf_name*** name of the VRF to which the BGP peer belongs.

Commands available in maintenance dynamic interface unit configuration mode:

`quiesce`

Example

This command creates dynamic BGP unit for IPv4 address ***1.0.1.1***, IPv6 addr ***1::1*** with quiesce and peer-group ***PG*** in ***VRF VRF1*** under maintenance configuration.

```
switch(config)# maintenance
switch(config-maintenance)# bgp 1.0.1.1
switch(config-maint-bgp-1.0.1.1)# exit
switch(config-maintenance)# bgp 1::1
switch(config-maint-bgp-1::1)# quiesce
switch(config-maint-bgp-1::1)# exit
switch(config-maintenance)# bgp PG vrf VRF1
switch(config-maint-bgp-PG)# exit
switch(config-maint-bgp-PG)# show active
maintenance
    bgp 1.0.1.1
    !
    bgp 1::1
        quiesce
    !
    bgp PG vrf VRF1
switch(config-maintenance)#
```

15.6.5.2 group bgp

The **`group bgp <group_name>`** command places the switch in group-BGP configuration mode for configuring the members of a BGP group in a particular VRF and associating a BGP maintenance profile for these members.

The command creates the group if the specified group does not exist prior to issuing the command.

The **`no group bgp <group_name>`** and **`default group bgp <group_name>`** removes the BGP group.

Command Mode

Global Configuration

Command Syntax

`group bgp` ***group_name***

`no group bgp` ***group_name***

`default group bgp` ***group_name***

Parameters

group_name name of the BGP group.

Commands available in group-BGP configuration mode:

- **`neighbor (ipv4 address | ipv6 address | peer-group)`**
- **`vrf (vrf-name)`**
- **`maintenance profile bgp`**



Note: Built-in BGP groups like ***AllBgpNeighborVrf-default*** and ***AllBgpNeighborVrf-<vrf_name>*** do not allow neighbor configuration. Only BGP maintenance profile can be associated to them.

Examples

- This command creates a BGP group ***BG1*** and enters into group ***BGP BG1*** configuration mode.

```
switch(config)# group bgp BG1
switch(config-group-bgp-BG1)# show active
group bgp BG1
exit
switch(config-group-bgp-BG1)#
```

- This command enters into ***BGP*** built-in configuration mode for ***AllBgpNeighborVrf-default***.

```
switch(config)# group bgp AllBgpNeighborVrf-default
switch(config-builtin-group-bgp-AllBgpNeighborVrf-default)#
group bgp AllBgpNeighborVrf-default
exit
switch(config-builtin-group-bgp-AllBgpNeighborVrf-default)# exit
switch(config)# show maintenance groups bgp AllBgpNeighborVrf-default
BGP Group: AllBgpNeighborVrf-default
Origin: Built-in
Neighbors:
Ipv4 Peers: 1.0.0.1, 1.0.1.2
Bgp Profile: Default
Vrf: default
Units: System
switch(config)#
```

15.6.5.3 group bgp <group_name>

The **`group bgp <group_name>`** command adds a BGP group to a unit.

The **`no group bgp <group_name>`** and **`default group bgp <group_name>`** removes the BGP group from a unit.

Command Mode

Maintenance Unit Configuration

Command Syntax

`group bgp` ***group_name***

`no group bgp` ***group_name***

`default group bgp` ***group_name***

Parameters

group_name name of the BGP group.

Example

This command adds a BGP group ***BG1*** to unit ***UNIT1***.

```
switch(config)# maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group bgp BG1
switch(config-unit-UNIT1)# show active
maintenance
    unit UNIT1
group bgp BG1
switch(config-unit-UNIT1)
```

15.6.5.4 group interface

The **`group interface`** command places the switch in group-intf configuration mode for configuring the members of interface group and associating a BGP/interface maintenance profile for these members.

The command creates the group if the specified group does not exist prior to issuing the command.

The **`no group interface <group_name>`** and **`default group interface <group_name>`** removes the interface group.

Command Mode

Global Configuration

Command Syntax

`group interface` ***group_name***

`no group interface` ***group_name***

`default group interface` ***group_name***

Parameters

group_name name of the interface group.

Commands available in group-BGP configuration mode:

- **`interface`**
- **`maintenance profile bgp`**
- **`maintenance profile interface`**

Note: Built-in Interface groups like ***AllEthernetInterface***, ***Linecard3***, ***Linecard4***, etc. do not allow interface configurations. Only BGP/interface maintenance profiles can be associated to them.

Examples

- This command creates an interface group ***IG1*** and enters into ***group interface IG1*** configuration mode.

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# show active
group interface IG1
exit
switch(config-group-if-IG1)#
```

- This command enters into built-in interface group ***AllEthernetInterface***.

```
switch(config)#group interface AllEthernetIntetrface
switch(config-builtin-group-if-AllEthernetInterface)# show active
group interface AllEthernetInterface
exit
switch(config-builtin-group-if-AllEthernetInterface)# exit
switch(config)# show maintenance groups interface AllEthernetInterface
Interface Group: AllEthernetInterface
Origin: Built-in
Interfaces:
Et1, Et2, Et3, Et4, Et5/1, â€¦ Et34, Et35, Et36
Profiles:
Interface Profile: Default
Bgp Profile: Default
Units: System#
```

15.6.5.5 group interface <group_name>

The **`group interface <group_name>`** command adds an interface to a unit.

The **`no group interface <group_name>`** and **`default group interface <group_name>`** removes the interface group from a unit.

Command Mode

Maintenance Unit Configuration

Command Syntax

`group interface` ***group_name***

`no group interface` ***group_name***

default group interface ***group_name***

Parameters

group_name name of the interface group.

Example

This command adds an ***group interface IG1*** to unit ***UNIT1***.

```
switch(config)# maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group interface IG1
switch(config-unit-UNIT1)# show active
maintenance
    unit UNIT1
group interface IG1
switch(config-unit-UNIT1)
```

15.6.5.6 initiator route-map <route-map-name> inout

The **initiator route-map <route-map-name> inout** command is a maintenance BGP profile configuration option for assigning the initiator route-map, which will be applied to inout (inbound and outbound).

The **no initiator route-map <route-map-name> inout** and **default initiator route-map <route-map-name> inout** removes this configuration from the BGP profile.

Command Mode

Maintenance-Profile-BGP Configuration

Command Syntax

initiator route-map ***route-map-name*** **inout**

no initiator route-map

default initiator route-map

Parameters

route-map-name initiator route-map name.

Example

This command configures initiator route-map ***RM1*** within a BGP profile ***BP1***.

```
switch(config)# maintenance
switch(config-maintenance)# profile bgp BP1
switch(config-profile-bgp-BP1)# initiator route-map RM1 inout
switch(config-profile-bgp-BP1)# show active
maintenance
    profile bgp BP1
        initiator route-map RM1 inout

switch(config-profile-bgp-BP1)#
```

15.6.5.7 interface

The **interface** command adds interfaces to interface group.

The **interface <intf-name>** and **default interface <intf-name>** removes the interface from the group.

Command Mode

Group-Interface Configuration

Command Syntax

`interface` ***interface-name***

`no interface` ***interface-name***

`default interface` ***interface-name***

Parameters

- ***interface-name*** name of the interface.
- ***ethernet e_range*** Ethernet interfaces specified by ***e_range***.
- port-channel ***p_range*** port channel interfaces specified by ***p_range***.
- vlan ***v_range*** vlans specified by ***v_range***.

Valid ***e_range***, ***p_range***, and ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges. Valid Ethernet numbers depend on the Ethernet interfaces available on the switch.

Example

- This command adds ***Ethernet8***, ***Ethernet9***, and ***port-channel10*** to the interface group ***IG1***.

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# interface Ethernet8-9
switch(config-group-if-IG1)# interface port-channel10
switch(config-group-if-IG1)# show active
group interface IG1
interface Et8-9
interface Po10
switch(config-group-if-IG1)# exit
switch(config)#
```

15.6.5.8 interface intf-name

The **`interface <intf-name>`** command places the switch in maintenance dynamic interface unit configuration mode.

The command creates the dynamic interface unit if the specified dynamic interface unit does not exist prior to issuing the command.

The **`no interface <intf-name>`** and **`default interface <intf-name>`** removes the dynamic interface unit from ***running-config***.

Command Mode

Maintenance Configuration

Command Syntax

`interface` ***interface-name***

`no interface` ***interface-name***

`default interface` ***interface-name***

Parameters

- ***interface-name*** name of the interface.
- **ethernet** ***e_range*** Ethernet interfaces specified by ***e_range***.
- **port-channel** ***p_range*** port channel interfaces specified by ***p_range***.
- **vlan** ***v_range*** vlans specified by ***v_range***.

Valid ***e_range***, ***p_range*** and ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Note: Different dynamic interface units are created for each interface in the range.

Commands available in maintenance dynamic interface unit configuration mode:

`quiesce`

Example

This command creates two dynamic interface units for interfaces ***Ethernet1-2*** under maintenance configuration.

```
switch(config)# maintenance
switch(config-maintenance)# interface Ethernet1-2
switch(config-maint-if-Et1-2)# exit
switch(config-maintenance)# show active
maintenance
    interface Ethernet1
    !
    interface Ethernet2
switch(config-maintenance)#
```

15.6.5.9 maintenance

The **`maintenance`** command allows you to enter maintenance configuration mode and specify maintenance configuration options.

The **`no maintenance`** and **`default maintenance`** command removes the maintenance configuration from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

`maintenance`

`no maintenance`

`default maintenance`

Commands available in maintenance configuration mode:

- **`unit`**
- **`bgp`**
- **`interface`**
- **`profile bgp`**
- **`profile interface`**
- **`profile unit`**
- **`profile interface <profile-name> default`**
- **`profile bgp <profile-name> default`**
- **`profile unit <profile-name> default`**

Example

This example shows the commands to enter maintenance configuration mode and configure maintenance related parameters.

```
switch(config)# maintenance
switch(config-maintenance)# profile unit foo
```

```
switch(config-profile-unit-foo)# on-boot duration 300
switch(config-profile-unit-foo)# exit
switch(config-maintenance)# unit U1
switch(config-unit-U1)# group interface IG1
switch(config-unit-U1)# group bgp BG1
switch(config-unit-U1)# profile unit foo
switch(config-unit-U1)# exit
switch(config-maintenance)# show active
maintenance
    profile unit foo
        on-boot duration 300
    unit U1
        group interface IG1
        group bgp BG1
        profile unit foo
switch(config-maintenance)#
```

15.6.5.10 maintenance profile bgp

The **maintenance profile bgp <profile-name>** command associates a BGP maintenance profile to an interface/BGP group. A BGP profile can be associated to both the interface and BGP group.

The **no maintenance profile bgp <profile-name>** and **default maintenance profile bgp <profile-name>** removes the profile from the interface/BGP group.

Command Mode

Group-Interface Configuration

Group-BGP Configuration

Built-in-Group-Interface Configuration

Built-in-Group-BGP Configuration

Command Syntax

maintenance profile bgp ***profile-name***

no maintenance profile bgp ***profile-name***

default maintenance profile bgp ***profile-name***

Parameters

profile name name of the BGP profile.

Examples

- This command adds ***BGP profile BP1*** to a BGP group ***BG1***.

```
switch(config)# group bgp BG1
switch(config-group-bgp-BG1)# neighbor 1.0.1.1
switch(config-group-bgp-BG1)# neighbor 1::1
switch(config-group-bgp-BG1)# neighbor PG
switch(config-group-bgp-BG1)# maintenance profile bgp BP1
switch(config-group-bgp-BG1)# show active
group bgp BG1
    neighbor 1.0.1.1
    neighbor 1::1
    neighbor PG
    maintenance profile bgp BP1
switch(config-group-bgp-BG1)# exit
switch(config)#
```

- This command adds BGP ***profile BP1*** to interface group ***IG1***.

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# interface Ethernet8-9
switch(config-group-if-IG1)# maintenance profile bgp BP1
switch(config-group-if-IG1)# show active
group interface IG1
    interface Et8-9
    maintenance profile bgp BP1
switch(config-group-if-IG1)# exit
switch(config)#
```

- This command adds BGP ***profile BP1*** to built-in interface group ***AllEthernetInterface***.

```
switch(config)# group interface AllEthernetInterface
switch(config-builtin-group-if-AllEtherentInterface)# maintenance profile
 bgp BP1
switch(config-builtin-group-if-AllEtherentInterface)# show active
group interface AllEthernetInterface
    maintenance profile bgp BP1

switch(config-builtin-group-if-AllEtherentInterface)#
```

15.6.5.11 maintenance profile interface

The **maintenance profile interface <profile-name>** command associates interface profile to interface group.

The **no maintenance profile interface <profile-name>** and **default maintenance profile interface <profile-name>** removes the interface profile from interface group.

Command Mode

Group-Interface Configuration

Built-in-Group-Interface Configuration

Command Syntax

maintenance profile interface ***profile-name***

no maintenance profile interface ***profile-name***

default maintenance profile interface ***profile-name***

Parameters

profile-name name of the interface profile.

Example

- This command adds ***profile interface IP1*** to interface group ***IG1***.

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# interface Ethernet8-9
switch(config-group-if-IG1)# maintenance profile interface IP1
switch(config-group-if-IG1)# show active
group interface IG1
    interface Et8-9
    maintenance profile interface IP1

switch(config-group-if-IG1)#
```

- This command adds ***profile interface IP1*** to built-in interface group ***AllEthernetInterface***.

```
switch(config)# group interface AllEthernetInterface
switch(config-builtin-group-if-AllEtherentInterface)# maintenance profile
interface IP1
switch(config-builtin-group-if-AllEtherentInterface)# show active
group interface AllEthernetInterface
    maintenance profile interface IP1

switch(config-builtin-group-if-AllEtherentInterface)#
```

15.6.5.12 neighbor

The **neighbor** command adds BGP peer(s) to a BGP group. The neighbors can be IPv4, IPv6, or a peer group. The **no neighbor <peer>** and **default neighbor <peer>** removes the BGP peer from the group.

Command Mode

Group-BGP Configuration

Command Syntax

neighbor ***ipv4_addr***

no neighbor ***ipv4_addr***

default neighbor ***ipv4_addr***

neighbor ***ipv6_addr***

no neighbor ***ipv6_addr***

default neighbor ***ipv6_addr***

neighbor ***peer group name***

no neighbor ***peer group name***

default neighbor ***peer group name***

Parameters

- ***ipv4_addr*** BGP neighbor ipv4 address.
- ***ipv6_addr*** BGP neighbor ipv6 address.
- ***peer group name*** BGP peer group name.

Example

- This command adds ipv4 peer ***1.0.1.1***, ipv6 peer ***1::1***, and peer group ***PG*** to the BGP group ***BG1***.

```
switch(config)# group bgp BG1
switch(config-group-bgp-BG1)# neighbor 1.0.1.1
switch(config-group-bgp-BG1)# neighbor 1::1
switch(config-group-bgp-BG1)# neighbor PG
switch(config-group-bgp-BG1)# group bgp BG1
switch(config-group-bgp-BG1)# neighbor 1.0.1.1
switch(config-group-bgp-BG1)# neighbor 1::1
switch(config-group-bgp-BG1)# neighbor PG
switch(config-group-bgp-BG1)# exit
switch(config)#
```

15.6.5.13 on-boot duration

The **on-boot duration** command is a maintenance unit profile configuration option for specifying the duration after which the associated unit will be brought out of maintenance after reboot. The on-boot property in the maintenance unit profile specifies that the unit will be placed into maintenance mode as part of boot-up, and remain so for the specified duration.

The **no on-boot** and **default on-boot** removes this configuration from the unit profile.

Command Mode

Maintenance-Profile-Unit Configuration

Command Syntax

on-boot duration ***duration***

no on-boot

default on-boot

Parameters

duration number of seconds for which unit will remain under maintenance after reboot (from ***300*** to ***3600*** seconds).

Example

This command configures on-boot duration of ***1000*** seconds in profile unit ***UP1***.

```
switch(config)# maintenance
switch(config-maintenance)# profile unit UP1
switch(config-profile-unit-UP1)# on-boot duration 1000
switch(config-profile-unit-UP1)# show active
maintenance
    profile unit UP1
        on-boot duration 1000

switch(config-profile-unit-UP1)#
```

15.6.5.14 profile bgp

The **profile bgp** command places the switch in maintenance profile BGP configuration mode for configuring initiator route-map.

The command creates the profile if the specified BGP profile does not exist prior to issuing the command.

The **no profile bgp <profile-name>** and **default profile bgp <profile-name>** removes the profile from ***running-config***.

Command Mode

Maintenance Configuration

Command Syntax

profile bgp ***profile-name***

no profile bgp ***profile-name***

default profile bgp ***profile-name***

Parameters

profile-name name of the BGP profile.

Commands available in maintenance profile BGP configuration mode:

initiator route-map (route-map name) inout

Example

This command creates BGP profile ***BP1***.

```
switch(config)# maintenance
switch(config-maintenance)# profile bgp BP1
switch(config-profile-bgp-BP1)# show active
maintenance
    profile bgp BP1

switch(config-profile-bgp-BP1)#
```

15.6.5.15 profile bgp <profile_name> default

The **profile bgp <profile_name> default** command configures a user-configured BGP profile as default BGP profile.

The **no profile bgp <profile_name> default** and **default profile bgp <profile_name> default** removes the user-configured BGP profile as default BGP profile.

Command Mode

Maintenance Configuration

Command Syntax

profile bgp ***profile_name*** **default**

no profile bgp ***profile_name*** **default**

default profile bgp ***profile_name*** **default**

Parameters

profile_name name of the BGP profile.

Example

This command configures user configured BGP profile BP1 as default BGP profile.

```
switch(config)# maintenance
switch(config-maintenance)# profile bgp BP1
switch(config-profile-bgp-BP1)# initiator route-map RM1 inout
switch(config-profile-bgp-BP1)# exit
switch(config-maintenance)#
switch(config-maintenance)# show maintenance profile bgp default
Bgp Profile: Default
  Initiator route-map: SystemGenerated
   route-map SystemGenerated permit 10
     Description:
       description System generated initiator route-map
     Match clauses:
     Set clauses:
       set community GSHUT additive
       set local-preference 0

switch(config-maintenance)# profile bgp BP1 default
switch(config-maintenance)# show maintenance profile bgp default
Bgp Profile: BP1
  Initiator route-map: RM1
switch(config-maintenance)#
switch(config-maintenance)# show active
maintenance
    profile bgp BP1
        initiator route-map RM1 inout
    profile bgp BP1 default

switch(config-maintenance)#
```

15.6.5.16 profile interface

The **profile interface** command places the switch in maintenance profile interface configuration mode for configuring rate-monitoring threshold, load-interval, and shutdown max-delay.

The command creates the profile if the specified interface profile does not exist prior to issuing the command.

The **no profile interface <profile-name>** and **default profile interface <profile-name>** removes the profile from ***running-config***.

Command Mode

Maintenance Configuration

Command Syntax

profile interface ***profile-name***

no profile interface ***profile-name***

default profile interface ***profile-name***

Parameters

profile-name name of the interface profile.

Commands available in maintenance profile interface configuration mode:

- **rate-monitoring load-interval**
- **rate-monitoring threshold**
- **shutdown max-delay**

Example

This command creates interface profile ***IP1***.

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1
switch(config-profile-intf-IP1)# show active
maintenance
    profile interface IP1

switch(config-profile-intf-IP1)#
```

15.6.5.17 profile interface <profile_name> default

The **profile interface <profile_name> default** command configures a user-configured interface profile as default interface profile.

The **no profile interface <profile_name> default** and **default profile interface <profile_name> default** removes the user-configured interface profile as default interface profile.

Command Mode

Maintenance Configuration

Command Syntax

profile interface ***profile_name*** **default**

no profile interface ***profile_name*** **default**

default profile interface ***profile_name*** **default**

Parameters

profile_name name of the interface profile.

Example

This command configures user configured interface profile ***IP1*** as default interface profile.

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1
switch(config-profile-intf-IP1)# rate-monitoring load-interval 100
switch(config-profile-intf-IP1)# rate-monitoring threshold 500
switch(config-profile-intf-IP1)# shutdown max-delay 100
switch(config-profile-intf-IP1)# exit
switch(config-maintenance)#
switch(config-maintenance)# show maintenance profile interface default
Interface Profile: Default
  Rate Monitoring:
    load-interval: 60 seconds
    threshold (in/out): 100 kbps
  shutdown:
    enabled: no
    max-delay: 300 seconds

switch(config-maintenance)#
switch(config-maintenance)# profile interface IP1 default
switch(config-maintenance)# show maintenance profile interface default
Interface Profile: IP1
  Rate Monitoring:
    load-interval: 100 seconds
    threshold (in/out): 500 kbps
  shutdown:
    enabled: yes
    max-delay: 100 seconds
switch(config-maintenance)#
switch(config-maintenance)# show active
maintenance
    profile interface IP1 default
    profile interface IP1
        rate-monitoring load-interval 100
        rate-monitoring threshold 500
        shutdown max-delay 100

switch(config-maintenance)#
```

15.6.5.18 profile unit

The **`profile unit`** command places the switch in maintenance profile unit configuration mode for configuring on-boot duration.

The command creates the profile if the specified BGP profile does not exist prior to issuing the command.

The **`no profile unit <profile-name>`** and **`default profile unit <profile-name>`** removes the profile from ***running-config***.

Command Mode

Maintenance Configuration

Command Syntax

`profile unit` ***profile-name***

`no profile unit` ***profile-name***

`default profile unit` ***profile-name***

Parameters

profile-name name of the unit profile.

Commands available in maintenance profile unit configuration mode:

`on-boot duration`

Example

This command creates unit profile ***UP1***.

```
switch(config)# maintenance
switch(config-maintenance)# profile unit UP1
switch(config-profile-unit-UP1)# show active
maintenance
    profile unit UP1

switch(config-profile-unit-UP1)#
```

15.6.5.19 profile unit <profile_name>

The **`profile unit <profile_name>`** command associates unit profile to a particular unit.

The **`no profile unit <profile_name>`** and **`default profile unit <profile_name>`** removes the unit profile from a unit.

Command Mode

Maintenance-Unit Configuration

Maintenance-Built-in-Unit Configuration

Command Syntax

`profile unit` ***profile-name***

`no profile unit` ***profile-name***

`default profile unit` ***profile-name***

Parameters

profile-name name of the unit profile.

Examples

- This command adds ***profile unit UP1*** to ***UNIT1***.

```
switch(config)# maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group interface IG1
switch(config-unit-UNIT1)# exit
switch(config-maintenance)# show maintenance units UNIT1
Unit Name: UNIT1
   Origin: User Configured
   Status: Not Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: never
   Interface Groups:
     IG1

switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)#profile unit UP1
switch(config-unit-UNIT1)# show maintenance units UNIT1
Unit Name: UNIT1
   Origin: User Configured
   Status: Not Under Maintenance
   Unit Profile: UP1
   Time Since Last State Change: never
   Interface Groups:
     IG1
switch(config-unit-UNIT1)# show active
maintenance
    unit UNIT1
        group interface IG1
        profile unit UP1

switch(config-unit-UNIT1)#
```

- This command adds ***profile unit UP2*** to built-in ***unit System***.

```
switch(config)# maintenance
switch(config-maintenance)#profile unit UP2
switch(config-profile-unit-UP2)# on-boot duration 600
switch(config-profile-unit-UP2)# exit
switch(config-maintenance)#
switch(config-maintenance)# unit System
switch(config-builtin-unit-System)# show active
maintenance
    unit System
switch(config-builtin-unit-System)# exit
switch(config-maintenance)# show maintenance units System
Unit Name: System
   Origin: Built-in
   Status: Not Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: never
   Interface Groups:
     AllEthernetInterface

switch(config-maintenance)#
switch(config-maintenance)# unit System
switch(config-builtin-unit-System)# profile unit UP2
switch(config-builtin-unit-System)# show active
maintenance
    unit System
        profile unit UP2
switch(config-builtin-unit-System)# exit
switch(config-maintenance)# show maintenance units System
Unit Name: System
   Origin: Built-in
   Status: Not Under Maintenance
   Unit Profile: UP2
   Time Since Last State Change: never
   Interface Groups:
     AllEthernetInterface

switch(config-maintenance)#
```

15.6.5.20 profile unit <profile_name> default

The **`profile unit <profile_name> default`** command configures a user-configured unit profile as default unit profile.

The **`no profile unit <profile_name> default`** and **`default profile unit <profile_name> default`** removes the user-configured unit profile as default unit profile.

Command Mode

Maintenance Configuration

Command Syntax

`profile unit` ***profile_name*** **`default`**

`no profile unit` ***profile_name*** **`default`**

`default profile unit` ***profile_name*** **`default`**

Parameters

profile_name name of the interface profile.

Example

This command configures user-configured unit profile ***UP1*** as the default unit profile.

```
switch(config)# maintenance
switch(config-maintenance)# profile unit UP1
switch(config-profile-unit-UP1)# on-boot duration 1000
switch(config-profile-unit-UP1)# exit
switch(config-maintenance)#
```

```
switch(config-maintenance)# show maintenance profiles unit default
Unit Profile: Default
    On-boot:
         enabled: no
         duration: 300 seconds

switch(config-maintenance)# profile unit UP1 default
switch(config-maintenance)# show maintenance profile unit default
Unit Profile: UP1
    On-boot:
         enabled: yes
         duration: 1000 seconds
switch(config-maintenance)#
switch(config-maintenance)# show active
maintenance
    profile unit UP1 default
    profile unit UP1
         on-boot duration 1000

switch(config-maintenance)#
```

15.6.5.21 quiesce

The **quiesce** command places a unit or dynamic interface/BGP unit into maintenance mode, gracefully transitioning traffic away from it.

The **no quiesce** and **default quiesce** exits the unit from maintenance.

Command Mode

Maintenance-Unit Configuration

Maintenance-Built-in-Unit Configuration

Maintenance Dynamic-Interface Unit Configuration

Maintenance Dynamic-Bgp Unit Configuration

Command Syntax

quiesce

no quiesce

default quiesce

Example

This command places unit ***UNIT1***, ***interface Et1***, BGP peer ***1.0.1.1*** in VRF default, and BGP peer ***1::1*** in **vrf** ***VRF1*** into maintenance.

```
switch(config)# group interface IG1
switch(config-group-if-IG1)# interface Ethernet3-6
switch(config-group-if-IG1)# maintenance profile interface IP1
switch(config-group-if-IG1)# exit
switch(config)# maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group interface IG1
switch(config-unit-UNIT1)# quiesce
switch(config-unit-UNIT1)# exit
switch(config-maintenance)# interface Ethernet1
switch(config-maint-if-Et1)# quiesce
switch(config-maint-if-Et1)# exit
switch(config-maintenance)# bgp 1.0.1.1
switch(config-maint-bgp-1.0.1.1)# quiesce
switch(config-maint-bgp-1.0.1.1)# exit
switch(config-maintenance)# bgp 1::1 vrf VRF1
switch(config-maint-bgp-1::1)# quiesce
switch(config-maint-bgp-1::1)# exit
switch(config-maintenance)# show active
maintenance
    bgp 1.0.1.1
        quiesce
    !
    bgp 1::1 vrf VRF1
        quiesce
    interface Et1
        quiesce
    unit UNIT1
        quiesce

switch(config-maintenance)# show maintenance
```

```
Flags:
o - On-boot maintenance
v - Violating traffic threshold

Unit Name              Status                  Time since last change    Flags
---------------------- ----------------------- ------------------------- -----
System                 Not Under Maintenance            never
UNIT1                  Under Maintenance             0:00:06 ago

Interface Name         Status                  Time since last change    Flags
---------------------- ----------------------- ------------------------- -----
Ethernet1              Entering Maintenance          0:00:06 ago

Bgp Neighbor(vrf: defa Status                  Time since last change    Flags
---------------------- ----------------------- ------------------------- -----
1.0.1.1                Under Maintenance             0:00:06 ago

Bgp Neighbor(vrf: VRF1 Status                  Time since last change    Flags
---------------------- ----------------------- ------------------------- -----
1::1                   Under Maintenance             0:00:06 ago

switch(config-maintenance)#
```

15.6.5.22 rate-monitoring load-interval

The **`rate-monitoring load-interval`** command is a maintenance interface profile configuration option for configuring the interfaces rate monitoring load interval with a load interval value between 5 and 600 seconds.

Command Mode

Maintenance-Profile-Interface Configuration

Command Syntax

`rate-monitoring load-interval` ***load_interval***

`no rate-monitoring load-interval` ***load_interval***

`default rate-monitoring load-interval` ***load_interval***

Parameters

load_interval load interval value between ***5*** and ***600*** seconds.

Example

This command configures the rate monitoring load interval for the profile interface ***IP1*** to a load interval of ***10*** seconds.

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1
switch(config-profile-intf-IP1)# rate-monitoring load-interval 10
switch(config-profile-intf-IP1)# show active
maintenance
    profile interface IP1
        rate-monitoring load-interval 10

switch(config-profile-intf-IP1)#
```

15.6.5.23 rate-monitoring threshold

The **`rate-monitoring threshold`** command is a maintenance interface profile configuration option for configuring the interfaces rate monitoring threshold with a threshold value between ***1*** and ***4294967295*** kilobytes.

The **`no rate-monitoring threshold`** and **`default rate-monitoring threshold`** removes this configuration from the interface profile.

Command Mode

Maintenance-Profile-Interface Configuration

Command Syntax

`rate-monitoring threshold` ***threshold_in_kbps***

`no rate-monitoring threshold` ***threshold_in_kbps***

`default rate-monitoring threshold` ***threshold_in_kbps***

Parameters

threshold_in_kbps threshold in kilobytes per second (kbps) between ***1*** and ***4294967295*** kilobytes.

Example

This command configures the rate monitoring threshold for the profile interface ***IP1*** to a threshold of ***1000*** kilobytes per second (kbps).

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1
switch(config-profile-intf-IP1)# rate-monitoring threshold 1000
switch(config-profile-intf-IP1)# show active
maintenance
    profile interface IP1
        rate-monitoring threshold 1000

switch(config-profile-intf-IP1)#
```

15.6.5.24 show interface

The **`show interface`** command displays detailed information about the interface.

It displays an extra line that reads: Under maintenance for time in hours and minutes.

Command Mode

EXEC

Command Syntax

`show interface` ***intf_name***

Parameters

intf_name name of the interface.

- **ethernet** ***e_range*** Ethernet interfaces specified by ***e_range***.
- **port-channel** ***p_range*** port channel interfaces specified by ***p_range***.
- **vlan** ***v_range*** vlans specified by ***v_range***.



Note: Valid ***e_range***, ***p_range***, and ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges. Valid Ethernet numbers depend on the Ethernet interfaces available on the switch.

Example

This command displays detailed information about ***interface ethernet 16/1***.

```
switch# show interface ethernet 16/1
Ethernet16/1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.7373.efc7
  Internet address is 1.0.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  IP MTU 1500 bytes, BW 40000000 kbit
  Full-duplex, 40Gb/s, auto negotiation: off, uni-link: n/a
  Up 4 hours, 44 minutes, 36 seconds
  Under maintenance for 4 hours, 22 minutes, 26 seconds
  Loopback Mode : None
  2 link status changes since last clear
```

```
  Last clearing of "show interface" counters 4:45:12 ago
  5 minutes input rate 20 bps (0.0% with framing overhead), 0 packets/sec
  5 minutes output rate 20 bps (0.0% with framing overhead), 0 packets/sec
     580 packets input, 46286 bytes
     Received 1 broadcasts, 0 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol, 0 input discards
     0 PAUSE input
     601 packets output, 48954 bytes
     Sent 7 broadcasts, 15 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred, 0 output discards
     0 PAUSE output switch#
```

15.6.5.25 show interface <intf_name> status

The **`show interface <intf_name> status`** command displays an m flag if the interface is undergoing maintenance operation.

Command Mode

EXEC

Command Syntax

`show interface` [***intf_name***] **`status`**

Parameters

intf_name name of the interface.

- **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
- **port-channel *p_range*** port channel interfaces specified by ***p_range***.
- **vlan *v_range*** vlans specified by ***v_range***.



Note: Valid ***e_range***, ***p_range***, and ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges. Valid Ethernet numbers depend on the Ethernet interfaces available on the switch.

Example

This command display tabular output and shows `m` flag for ***Ethernet16/1*** status.

```
switch# show interface Ethernet16/1 status
Port Name Status Vlan Duplex Speed Type Flags
Et1 disabled 1 auto auto 1000BASE-T
...
Et14/1 connected 2 full 40G 40GBASE-CR4
Et15/1 connected 2 full 40G 40GBASE-CR4
Et16/1 connected routed full 40G 40GBASE-CR4 m
Et17/1 notconnect 1 full 10G Not Present
...
switch#
```

15.6.5.26 show ip | ipv6 bgp

The **`show ip | ipv6 bgp`** command displays maintenance related information when relevant.

Command Mode

EXEC

Command Syntax

`show ip bgp neighbors` ***peer_addr*** [vrf ***vrf_name***]

`show ipv6 bgp peers` ***peer_addr*** [vrf ***vrf_name***]

Parameters

peer_addr name of the peer.

- ***ipv4_addr*** BGP neighbor IPv4 address.
- ***ipv6_addr*** BGP neighbor IPv6 address.
- ***peer-group-name*** BGP peer group name.
- ***vrf_name*** name of the VRF.

Example

This command displays the **m** flag in **show ip bgp summary** output for peer ***1.0.1.2*** which is in maintenance mode.

```
switch# show ip bgp neighbors 1.0.1.2
BGP neighbor is 1.0.1.2, remote AS 300, external link
  BGP version 4, remote router ID 0.0.2.1, VRF default
  Negotiated BGP version 4
  Last read 00:00:09, last write 00:00:11
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Connect timer is inactive
  Idle-restart timer is inactive
  Session is under maintenance
  BGP state is Established, up for 04:55:11
  Number of transitions to established: 1
  Last state was OpenConfirm
  Last event was RecvKeepAlive
  Neighbor Capabilities:
    Multiprotocol IPv4 Unicast: advertised and received and negotiated
    Four Octet ASN: advertised and received
    Route Refresh: advertised and received and negotiated
    Send End-of-RIB messages: advertised and received and negotiated
    Additional-paths Receive:
      IPv4 Unicast: advertised and received
  Restart timer is inactive
  End of rib timer is inactive
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent      Rcvd
    Opens:                  1         1
    Notifications:          0         0
    Updates:                6         2
    Keepalives:           297       297
    Route-Refresh:          0         0
    Total messages:       304       300
  Prefix statistics:
                         Sent      Rcvd
    IPv4 Unicast:           2         1
    IPv6 Unicast:           0         0
  Inbound updates dropped by reason:
    AS path loop detection: 0
    Enforced First AS: 0
    Malformed MPBGP routes: 0
    Originator ID matches local router ID: 0
    Nexthop matches local IP address: 0
    Unexpected IPv6 nexthop for IPv4 routes: 0
    Nexthop invalid for single hop eBGP: 0
  Inbound paths dropped by reason:
    IPv4 labeled-unicast NLRIs dropped due to excessive labels: 0
  Outbound paths dropped by reason:
    IPv4 local address not available: 0
    IPv6 local address not available: 0
  Maintenance-mode:
    Inbound and Outbound policy
      Route map is SystemGenerated
Local AS is 200, local router ID 0.0.1.1
TTL is 1
Local TCP address is 1.0.1.1, local port is 179
Remote TCP address is 1.0.1.2, remote port is 51936
Auto-Local-Addr is disabled
TCP Socket Information:
  TCP state is ESTABLISHED
  Recv-Q: 0/32768
  Send-Q: 0/32768
  Outgoing Maximum Segment Size (MSS): 1448
  Total Number of TCP retransmissions: 0
  Options:
    Timestamps enabled: yes
    Selective Acknowledgments enabled: yes
    Window Scale enabled: yes
    Explicit Congestion Notification (ECN) enabled: no
  Socket Statistics:
    Window Scale (wscale): 9,7
```

```
    Retransmission Timeout (rto): 204.0ms
    Round-trip Time (rtt/rtvar): 7.5ms/3.0ms
    Delayed Ack Timeout (ato): 40.0ms
    Congestion Window (cwnd): 10
    TCP Throughput: 15.45 Mbps
    Advertised Recv Window (rcv_space): 14480 switch#
```

15.6.5.27 show ip | ipv6 bgp summary [vrf <vrf_name>]

The **show ip | ipv6 bgp summary [vrf <vrf_name>]** command displays the m flag if the BGP IPv4 or IPv6 peer is undergoing maintenance operation.

Command Mode

EXEC

Command Syntax

show ip bgp summary [vrf ***vrf_name***]

show ipv6 bgp summary [vrf ***vrf_name***]

Parameter

vrf_name name of the VRF.

Example

This command displays the m flag in **show ip bgp summary** output for peer ***1.0.1.2*** which is in maintenance mode.

```
switch# show ip bgp summary
BGP summary information for VRF default
Router identifier 0.0.1.1, local AS number 200
Neighbor Status Codes: m - Under maintenance
  Neighbor V  AS MsgRcvd MsgSent InQ OutQ Up/Down State PfxRcd
PfxAcc
  1.0.0.1 4 100 292 296 0 0 04:47:44 Estab 1 1
m 1.0.1.2 4 300 292 296 0 0 04:47:44 Estab 1 1 switch#
```

15.6.5.28 show maintenance

The **show maintenance** command provides brief information about all units/dynamic interface unit/dynamic bgp unit and status.

o'- flag displays that unit is undergoing or has undergone a maintenance operation because of on-boot.

v - flag displays that one/some of the interfaces are violating traffic, i.e. traffic for those interfaces is above threshold.

Command Mode

EXEC

Command Syntax

show maintenance

Example

This command displays maintenance mode details.

```
switch# show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold
Unit Name  Status                  Time since last change Flags
---------- ----------------------- ---------------------- -----
System Not Under Maintenance       never
```

```
Foo        Under Maintenance       0:00:40 ago             o

Interface Name Status               Time since last change Flags
-------------- -------------------- ---------------------- -----
Ethernet16/1   Entering Maintenance 0:00:02 ago            v

Bgp Neighbor(vrf: defa Status          Time since last change Flags
---------------------- ---------------- ---------------------- -----
1.0.0.2        Not Under Maintenance   never

Bgp Neighbor(vrf: red) Status          Time since last change Flags
---------------------- ---------------- ---------------------- -----
2.0.1.2       Under Maintenance        0:00:16 ago switch#
```

15.6.5.29 show maintenance bgp

The **`show maintenance bgp`** command displays detailed maintenance information about BGP peers.

Command Mode

EXEC

Command Syntax

`show maintenance bgp` ***ipv4_addr*** [vrf ***vrf_name***] | ***ipv6_addr*** [vrf ***vrf_name***] | ***peer_group*** [vrf ***vrf_name***] | ip all [vrf ***vrf_name*** | vrf all] | ipv6 all [vrf ***vrf_name*** | vrf all]

Parameters

- ***ipv4_addr*** BGP neighbor ipv4 address.
- ***ipv6_addr*** BGP neighbor ipv6 address.
- ***peer_group*** BGP peer group name.
- ***vrf_name*** name of the VRF to which peer belongs.
- **`ip all vrf`** ***vrf_name*** all ipv4 peers in specified VRF.
- **`ipv6 all vrf`** ***vrf_name*** all ipv6 peers in specified VRF.
- **`ip all vrf all`** all ipv4 peers in all the VRFs.
- **`ipv6 all vrf all`** all ipv6 peers in all the VRFs.

Example

This command displays maintenance information about BGP peers ***1.0.0.1*** and ***1.0.1.1*** and maintenance route-map applied.

```
switch# show maintenance bgp ip all vrf all
BGP peer maintenance information for VRF default
Router identifier 0.0.1.1, local AS number 200
  Neighbor: 1.0.0.1
    Maintenance state: Under Maintenance
    Maintenance route-map: SystemGenerated
  Neighbor: 1.0.1.2
    Maintenance state: Under Maintenance
    Maintenance route-map: SystemGenerated switch#
```

15.6.5.30 show maintenance bgp receiver route-map

The **`show maintenance bgp receiver route-map`** command displays receiver route-map which is applied during maintenance operation.

Command Mode

EXEC

Command Syntax

`show maintenance bgp receiver route-map`

Example

This command displays receiver route-map contents.

```
switch# show maintenance bgp receiver route-map
route-map SystemGenerated permit 10
  Description:
    description System generated receiver route-map
  Match clauses:
    match community GSHUT-LIST
  SubRouteMap:
  Set clauses:
route-map SystemGenerated permit 50
  Description:
    description System generated receiver route-map
  Match clauses:
  SubRouteMap:
  Set clauses:
switch#
```

15.6.5.31 show maintenance interface status quiesced

This example of the **show maintenance interface status quisced** command displays maintenance mode interface status details for quiesced interfaces.

Example

```
switch(config)#show maintenance interface status quiesced
Flags:
v - Violating traffic threshold
s - Shutdown for maintenance
Rate (Mbps)
Interface Status            In  Out Flags
--------- ----------------- --- --- -----
Ethernet1 Under Maintenance 0.3 0.0 v
Ethernet2 Under Maintenance 0.0 0.0
Ethernet4 Under Maintenance 0.0 0.0

switch(config)#
```

15.6.5.32 show maintenance debug

The **show maintenance debug** command displays the history of various maintenance operations on a unit/interface/BGP peer.

Command Mode

EXEC

Command Syntax

show maintenance debug bgp [***peer_name***] | interface [***intf_name***] | units [***unit_name***]

Parameters

- **bgp** display history of all dynamic BGP units which have undergone maintenance operation.
- **interface** display history of all dynamic interface units which have undergone maintenance operation.
- **units** display history of all units which have undergone maintenance operation.
- ***peer_name*** name of the peer.
  - ***ipv4_addr*** BGP neighbor IPv4 address.

- ***ipv6_addr*** BGP neighbor IPv6 address.
- ***peer-group-name*** BGP peer group name.

- ***intf_name*** name of the interface.
  - **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
  - **port-channel *p_range*** port channel interfaces specified by ***p_range***.
  - **vlan *v_range*** vlans specified by ***v_range***.



Note: Valid ***e_range***, ***p_range***, and ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges. Valid Ethernet numbers depend on the Ethernet interfaces available on the switch.

- ***unit_name*** name of the unit.

Example

This command displays history of maintenance operation on ***Ethernet 16/1***.

```
switch# show maintenance debug interface Ethernet 16/1-4
Interface Ethernet16/1
History:
Maintenance Enter Stage Progression started 4:07:07 ago @ 2016-08-29 22:38:54
0.000000 maintEnter stages started
0.000091 stage begin started
0.000151 event begin:EventMgr started
0.004222 event begin:EventMgr completed
0.004256 stage begin is complete
0.004315 stage before_bgp started
0.004368 event before_bgp:EventMgr started
0.005820 event before_bgp:EventMgr completed
0.005843 stage before_bgp is complete
0.005904 stage bgp started
0.005947 event bgp:Rib started
0.013821 event bgp:Rib completed
0.013855 stage bgp is complete
0.013921 stage after_bgp started
0.013974 event after_bgp:EventMgr started
0.015848 event after_bgp:EventMgr completed
0.015878 stage after_bgp is complete
0.015935 stage before_ratemon started
0.015982 event before_ratemon:EventMgr started
0.017394 event before_ratemon:EventMgr completed
0.017423 stage before_ratemon is complete
0.017470 stage ratemon started
0.017506 event ratemon:MaintenanceMode started
5.021404 event ratemon:MaintenanceMode completed
5.021438 stage ratemon is complete
5.021500 stage after_ratemon started
5.021556 event after_ratemon:EventMgr started
5.023223 event after_ratemon:EventMgr completed
5.023247 stage after_ratemon is complete
5.023300 stage end started
5.023352 event end:EventMgr started
5.024683 event end:EventMgr completed
5.024705 stage end is complete
5.024762 maintEnter stages complete
```

15.6.5.33 show maintenance groups

The **`show maintenance groups`** command displays all the interface/BGP groups along with their members and associated profiles.

Command Mode

EXEC

Command Syntax

`show maintenance groups` interface | bgp ***group_name***

Parameters

- ***interface*** display only interface groups
- **bgp** display only BGP groups
- ***group_name*** name of the group

Example

This command displays group details for built-in interface group ***AllEthernetInterface*** and built-in BGP group ***AllBgpNeighborVrf-default*** and user-configured interface group ***IG1***.

```
switch# show maintenance groups
Interface Group: AllEthernetInterface
  Origin: Built-in
  Interfaces:
    Et1, Et2, Et3, Et4, Et5/1, Et5/2, Et5/3, Et5/4, Et6/1, Et6/2, Et6/3, Et6/4,
Et7/1, Et7/2, Et7/3, Et7/4, Et8/1, Et8/2, Et8/3, Et8/4, Et9/1, Et9/2, Et9/3,
Et9/4, Et10/1, Et10/2, Et10/3, Et10/4, Et11/1, Et11/2, Et11/3, Et11/4, Et12/1,
Et12/2, Et12/3, Et12/4, Et13/1, Et13/2, Et13/3, Et13/4, Et14/1, Et14/2, Et14/3,
Et14/4, Et15/1, Et15/2, Et15/3, Et15/4, Et16/1, Et16/2, Et16/3, Et16/4, Et17/1,
Et17/2, Et17/3, Et17/4, Et18/1, Et18/2, Et18/3, Et18/4, Et19/1, Et19/2, Et19/3,
Et19/4, Et20/1, Et20/2, Et20/3, Et20/4, Et21/1, Et21/2, Et21/3, Et21/4, Et22/1,
Et22/2, Et22/3, Et22/4, Et23/1, Et23/2, Et23/3, Et23/4, Et24/1, Et24/2, Et24/3,
Et24/4, Et25/1, Et25/2, Et25/3, Et25/4, Et26/1, Et26/2, Et26/3, Et26/4, Et27/1,
Et27/2, Et27/3, Et27/4, Et28/1, Et28/2, Et28/3, Et28/4, Et29, Et30, Et31, Et32,
Et33, Et34, Et35, Et36
    Profiles:
        Interface Profile: low-load-interval-profile
        Bgp Profile: Default
    Units: System
Interface Group: IG1
    Origin: User Configured
    Interfaces:
        Et1, Et2, Et3, Et4, Po10, Po11, Po12
    Profiles:
        Interface Profile: IP1
        Bgp Profile: BP1
    Units: UNIT1
Bgp Group: AllBgpNeighborVrf-default
    Origin: Built-in
    Neighbors:
        Ipv4 Peers: 1.0.0.1, 1.0.1.2
    Bgp Profile: Default
    Vrf: default
    Units: System switch#
```

15.6.5.34 show maintenance interface

The **`show maintenance interface`** command displays detailed information about interfaces and their maintenance status with traffic rates.

Command Mode

EXEC

Command Syntax

`show maintenance interface` [***intf_name*** [detail] | detail]

Parameters

- ***intf_name*** name of the interface or sub-interface. Options include:
  - **ethernet *e_range*** Ethernet interfaces specified by ***e_range***.
  - **port-channel *p_range*** port channel interfaces specified by ***p_range***.
  - **vlan *v_range*** vlans specified by v_range.
- **detail** provides the detailed rate-monitoring information

Guidelines

Valid ***e_range***, ***p_range***, and ***p_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Examples

- This command displays interface status and traffic rates.

```
switch# show maintenance interface
Flags:
v - Violating traffic threshold
s - Shutdown for maintenance
 Rate (Mbps)
Interface           Status                      In        Out        Flags
```

```
------------------- --------------------------- --------- ---------- ------
Ethernet1 Not Under Maintenance -  -
Ethernet2 Not Under Maintenance -  -
Ethernet3 Under Maintenance 0.0 0.0
Ethernet4 Not Under Maintenance -  -
...
Ethernet35 Entering Maintenance 8.7 2.9
Ethernet36 Not Under Maintenance - -
switch#
```

- This command displays detailed information about the ***interface Ethernet16/1***.

```
switch# show maintenance interface Ethernet16/1 detail
Ethernet16/1 is Under Maintenance
Groups: AllEthernetInterface
Selected profiles from Interface groups:
Interface Maintenance profile: low-load-interval-profile
Bgp Maintenance profile: Default
Bgp:
Maintenance State: Under Maintenance
Vrf: default
Neighbor: 1.0.1.2
Maintenance routemap: SystemGenerated
Rate Monitoring:
Passive monitoring since 0:42:25 ago
Total samples taken: 236
Before Maintenance:
Below threshold: 1
Above threshold: 0
After Maintenance:
Below threshold: 235
Above threshold: 0
Last sample information:
Sample taken 0:00:04 ago
In: 0.0 Mbps
Out: 0.0 Mbps switch#
```

15.6.5.35 show maintenance interface status

The **`show maintenance interface status`** command displays maintenance status and rates for interfaces.

Command Mode

EXEC

Command Syntax

`show maintenance interface status` active | **`entering`** | **`exiting`** | quiesced

Parameters

- **active** interfaces which are active.
- ***entering*** interface which are entering maintenance.
- ***exiting*** interface which are exiting maintenance.
- ***quiesced*** interface which are under maintenance.

Example

This command displays interface status and traffic rates of interfaces which are quiesced.

```
switch# show maintenance interface status quiesced
Flags:
v - Violating traffic threshold
s - Shutdown for maintenance
 Rate (Mbps)
Interface            Status                          In    Out    Flags
-------------------- ------------------------------ ----- ------ ------
Ethernet1 Not Under Maintenance -  -
Ethernet2 Not Under Maintenance -  -
Ethernet3 Not Under Maintenance -  -
Ethernet4 Not Under Maintenance -  -
Ethernet16/1 Under Maintenance 0.0 0.0
Port-Channel10 Under Maintenance 100.5 50.5v
```

```
Port-Channel11 Entering Maintenance 15.5 10.5
Port-Channel10 Under Maintenance -  -

switch#
```

15.6.5.36 show maintenance profiles

The **show maintenance profiles** command displays all the interface/BGP/unit profiles configuration.

Command Mode

EXEC

Command Syntax

show maintenance profiles interface | bgp | unit ***profile_name***

Parameters

- **interface** display only interface profiles.
- **bgp** display only BGP profiles.
- **unit** display only unit profiles.
- ***profile_name*** name of the profile.

Example

This command displays profile configuration details for interface profile ***IP1***, ***unit profile UP1*** and BGP profile ***BP1***.

```
switch# show maintenance profiles
Interface Profile: IP1
    Rate Monitoring:
        load-interval: 444 seconds
        threshold (in/out): 4000 Kbps
    shutdown:
        enabled: yes
        max-delay: 399 seconds
Bgp Profile: BP1
    Initiator route-map:
        name: RM1
Unit Profile: UP1
    On-boot:
        enabled: yes
        duration: 340 seconds

switch #
```

15.6.5.37 show maintenance stages

The **show maintenance stages** command displays stages of maintenance operation while entering/exiting maintenance.

Command Mode

EXEC

Command Syntax

show maintenance stages [enter | exit]

Parameters

- **enter** display maintenance stages during maintenance enter operation.
- **exit** display maintenance stages during maintenance exit operation.

Examples

- This command displays maintenance mode stages details.

```
switch# show maintenance stages
No.         Stage          Description
---------   -------------  -----------------
1 bgp       BGP            Maintenance processing
2 ratemon   Interface      Rate Monitoring
Maintenance Exit           Stage Sequence
No.         Stage          Description
---------   -------------  ------------------
1 ratemon Interface Rate Monitoring
2 bgp BGP Maintenance processing
switch #
```

- This command displays maintenance mode stage details during entry.

```
switch# show maintenance stages enter
No.       Stage         Description
--------- ------------- --------------------
1 bgp     BGP           Maintenance processing
2 ratemon Interface     Rate Monitoring switch#
```

15.6.5.38 show maintenance summary

The **show maintenance summary** command displays summarized information about the maintenance mode operations such as number of units configured, number of units Entering/Exiting maintenance etc.

Command Mode

EXEC

Command Syntax

show maintenance summary

Example

This command displays summary of maintenance mode operations.

```
switch# show maintenance summary
Number of Units Configured: 0
Number of Units Exiting Maintenance: 0
Number of Units Entering Maintenance: 0
Number of Units Not Under Maintenance: 1
Number of Units Under Maintenance: 0
Directly Put Under Maintenance:
  Number of interfaces Entering Maintenance: 0
  Number of interfaces Under Maintenance: 1
  Number of bgp peers Entering Maintenance: 0
  Number of bgp peers Under Maintenance: 1
Rate Monitoring:
  Number of interfaces Entering Maintenance: 0
  Number of interfaces Under Maintenance: 1
  Number of interfaces Under Maintenance with threshold violation: 0
  Number of interfaces shutdown for maintenance: 0 switch#
```

15.6.5.39 show maintenance units

The **show maintenance units** command displays detailed information about the particular unit.

Command Mode

EXEC

Command Syntax

show maintenance units [***unit_name***]

Parameters

unit_name name of unit.

Example

This command displays maintenance units details.

```
switch# show maintenance units
Unit Name: System
Origin: Built-in
Status: Not Under Maintenance
Unit Profile: Default
Time Since Last State Change: never
Bgp Groups:
AllBgpNeighborVrf-default
Interface Groups:
AllEthernetInterface

Unit Name: UNIT1
Origin: User Configured
Status: Under Maintenance
Unit Profile: UP1
Time Since Last State Change: 0:00:08 ago
Bgp Groups:
BG1
Interface Groups:
IG1
History:
2016-08-29 23:05:30 old state: 'maintenanceModeEnter' to new state:
'underMaintenance' 0:00:08 ago
2016-08-29 23:05:30 old state: 'active' to new state: 'maintenanceModeEnter'
0:00:08 ago switch#
```

15.6.5.40 shutdown max-delay

The **shutdown max-delay** command is a maintenance interface profile configuration option for configuring the maximum duration after which the interface is shutdown with a value between ***1*** and ***4294967295*** seconds.

The **no shutdown** and **default shutdown** removes this configuration from the interface profile.

Command Mode

Maintenance-Profile-Interface Configuration

Command Syntax

shutdown max-delay ***delay***

no shutdown max-delay ***delay***

default shutdown max-delay ***delay***

Parameters

delay maximum shutdown delay between ***1*** and ***4294967295*** seconds.

Example

This command configures the shutdown max-delay for the profile interface ***IP1*** to ***500*** seconds or ***1*** hour.

```
switch(config)# maintenance
switch(config-maintenance)# profile interface IP1
switch(config-profile-intf-IP1)# shutdown max-delay 500
switch(config-profile-intf-IP1)# show active
```

```
maintenance
    profile interface IP1
        shutdown max-delay 500

switch(config-profile-intf-IP1)#
```

15.6.5.41 trigger on-maintenance

The **`trigger on-maintenance`** command is an event handler configuration for triggering actions during the maintenance operation of a unit, interface and BGP peer at specified stages.

The event-handler configuration takes effect only after exiting the event-handler configuration mode.

Command Mode

Event-handler Configuration

Command Syntax

`trigger on-maintenance` [enter | exit][unit ***unit_name*** | bgp [***ipv4_addr*** | ***ipv6_addr*** | ***peer_group***][vrf ***vrf_name***] | [interface ***intf_name***] [begin | end | all] |[before | after][stage ***stage_name***]

Parameters

- **enter** trigger on-maintenance event-handler on maintenance enter operation.
- **exit** trigger on-maintenance event-handler on maintenance exit operation.
- **bgp** trigger event-handler on dynamic BGP unit maintenance operation.
  - ***pv4_addr*** BGP neighbor ipv4 address.
  - ***pv6_addr*** BGP neighbor ipv6 address.
  - ***peer_group*** BGP peer group name.
- ***vrf vrf_name*** name of the VRF to which BGP peer belongs.
- **interface** trigger event-handler on dynamic interface unit maintenance operation.
  - ***intf_name*** name of the interface.
    - **ethernet** trigger event-handler on specified Ethernet interface.
    - **port-channel** trigger event-handler on specified port channel interface.
    - **vlan** trigger event-handler on specified VLAN.

      Note: Comma-delimited list, ranges are not supported.
- **unit** trigger event-handler on maintenance operation of unit.
- **begin** action is triggered in the beginning of maintenance operation.
- **end** action is triggered at the end of maintenance operation.
- ***stage_name*** action is triggered at specified stage.
  - **bgp** and **ratemon** are the two stages.
- **all** action is triggered at all the stages
- **before** action is triggered before the specified stage
- **after** action is triggered after the specified stage

Examples

- This command configures event-handler *E1*, which triggers on maintenance an enter operation of unit ***UNIT1*** at all the stages.

```
switch(config)# event-handler E1
switch(config-handler-E1)# trigger on-maintenance enter unit UNIT1 all
switch(config-handler-E1)# action bash FastCli -c "show maintenance"
switch(config-handler-E1)# exit
switch(config)# show event-handler E1
```

```
Event-handler E1
Trigger: Asynchronous on-maintenance enter unit UNIT1 all delay 0 seconds
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: FastCli -c "show maintenance"
Action expected to finish in less than 10 seconds
Last Trigger Detection Time: Never
Total Trigger Detections: 0
Last Trigger Activation Time: Never
Total Trigger Activations: 0
Last Action Time: Never
Total Actions: 0

switch(config)#
```

- This command configures event-handler ***E2***, which triggers on maintenance an exit operation of dynamic interface unit ***Ethernet1*** before stage bgp.

```
switch(config)# event-handler E2
switch(config-handler-E2)# trigger on-maintenance exit interface Ethernet1
 before
stage bgp
switch(config-handler-E2)# action bash FastCli -c "show maintenance summary"
switch(config-handler-E2)# exit
switch(config)# show event-handler E2
Event-handler E2
Trigger: Asynchronous on-maintenance exit interface Ethernet1 before stage
 bgp
delay 0 seconds
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: FastCli -c "show maintenance summary"
Action expected to finish in less than 10 seconds
Last Trigger Detection Time: Never
Total Trigger Detections: 0
Last Trigger Activation Time: Never
Total Trigger Activations: 0
Last Action Time: Never
Total Actions: 0

switch(config)#
```

- This command configures event-handler ***E3***, which triggers on maintenance an enter operation of dynamic BGP unit ***1::1*** in VRF ***VRF1*** at the last stage end.

```
switch(config)# event-handler E3
switch(config-handler-E3)# trigger on-maintenance enter bgp 1::1 vrf VRF1
 end
switch(config-handler-E3)# action bash FastCli -c "show maintenance bgp ip
 all vrf all"
switch(config-handler-E3)# exit
switch(config)# show event-handler E3
Event-handler E3
Trigger: Asynchronous on-maintenance enter bgp 1::1 vrf VRF1 end delay 0
 seconds
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: FastCli -c "show maintenance bgp ip all vrf all"
Action expected to finish in less than 10 seconds
Last Trigger Detection Time: Never
Total Trigger Detections: 0
Last Trigger Activation Time: Never
Total Trigger Activations: 0
Last Action Time: Never
Total Actions: 0
```

```
switch(config)#
```

15.6.5.42 unit

The **unit <unit_name>** command places the switch in maintenance unit configuration mode for configuring BGP/interface groups in the unit.

The command creates the unit if the specified unit profile does not exist prior to issuing the command.

The **no unit <unit-name>** and **default unit <unit-name>** removes the unit from ***running-config***.

Command Mode

Maintenance Configuration

Command Syntax

unit linecard [***l_range*** | ***unit_name***]

no unit linecard [***l_range*** | ***unit_name***]

default unit linecard [***l_range*** | ***unit_name***]

Parameters

- ***l_range*** name of the Linecard built-in unit.
- ***0 l_range*** linecards available on the switch.
- ***unit_name*** name of the user-configured unit.

Commands available in maintenance unit configuration mode:

- **group interface**
- **group bgp**
- **profile unit**
- **quiesce**



Note: Built-in units like System, Linecard3, Linecard4, etc. do not allow group configuration but unit profile can be associated to these units.

Examples

- This command creates maintenance unit ***UNIT1***.

```
switch(config)# maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# show active
maintenance
unit UNIT1
switch(config-unit-UNIT1)#
```

- This command enters the built-in ***Linecard1*** unit configuration mode.

```
switch(config)# maintenance
switch(config-maintenance)# unit Linecard1
switch(config-builtin-unit-Linecard1)# show active
maintenance
unit Linecard1
switch(config-builtin-unit-Linecard1)#
```

15.6.5.43 vrf

The **vrf** command specifies the VRF for BGP group. All the neighbors configured in the BGP group are considered to be members of the BGP group in the particular VRF context.

The **no vrf <vrf-name>** and **default vrf <vrf-name>** removes the VRF configuration from the BGP group and sets the VRF context to default.

Command Mode

Group-BGP Configuration

Command Syntax

vrf ***vrf_name***

no vrf ***vrf_name***

default vrf ***vrf_name***

Parameters

vrf_name name of the VRF in a group belonging to neighbors in that group.

Example

This command specifies VRF ***VRF1*** for the neighbors in the BGP group ***BGP1***.

```
switch(config)# group bgp BG1
switch(config-group-bgp-BG1)# neighbor 1.0.1.1
switch(config-group-bgp-BG1)# neighbor 1::1
switch(config-group-bgp-BG1)# neighbor PG
switch(config-group-bgp-BG1)# vrf VRF1
switch(config-group-bgp-BG1)# show active
group bgp BG1
    neighbor 1.0.1.1
    neighbor 1::1
    neighbor PG
    vrf VRF1
switch(config-group-bgp-BG1)# exit
switch(config)#
```

15.7 Bidirectional Forwarding Detection

This section describes Bidirectional Forwarding Detection (BFD) and how it is configured in relation to various protocols. Topics in this section include:

- Introduction
- BFD Configuration
- Hardware Accelerated BFD Transmit
- BFD Commands

15.7.1 Introduction

In networks without data link signaling, connection failures are usually detected by the hello mechanisms of routing protocols. Detection can take over a second, and reducing detection time by increasing the rate at which hello packets are exchanged can create an excessive burden on the participating CPUs.

Bidirectional Forwarding Detection (BFD) is a low-overhead, protocol-independent mechanism which adjacent systems can use instead for faster detection of faults in the path between them. BFD is strictly a failure-detection mechanism, and does not discover neighbors or reroute traffic.

BFD is a simple mechanism which detects the liveness of a connection between adjacent systems, allowing it to quickly detect failure of any element in the connection. It does not operate independently, but only as an adjunct to routing protocols. The routing protocols are responsible for neighbor detection, and create BFD sessions with neighbors by requesting failure monitoring from BFD.

Once a BFD session is established with a neighbor, BFD exchanges control packets to verify connectivity and informs the requesting protocol of failure if a specified number of successive packets are not received. The requesting protocol is then responsible for responding to the loss of connectivity.

Routing protocols using BFD for failure detection continue to operate normally when BFD is enabled, including the exchange of hello packets.

The basic behavior of BFD is defined in ***RFC 5880***.

15.7.1.1 BFD Modes

BFD functions in asynchronous or demand mode, and also offers an echo function. EOS supports asynchronous mode and the echo function.

- Asynchronous Mode
- Demand Mode

15.7.1.1.1 Asynchronous Mode

In asynchronous mode, BFD control packets are exchanged by neighboring systems at regular intervals. If a specified number of sequential packets are not received, BFD declares the session to be down.

15.7.1.1.2 Demand Mode

In demand mode, once the BFD session is established, the participating systems can request that BFD packets not be sent, then request an exchange of packets only when needed to verify connectivity. EOS does not support demand mode.

15.7.1.2 Echo Function

When the echo function is in use, echo packets are looped back through the hardware forwarding path of the neighbor system without involving the CPU. Failure is detected by an interruption in the stream of echoed packets. The minimum reception rate for BFD control packets from the neighbor is also changed automatically when the echo function is operational, because liveness detection is supplied by the echo packets.

While BFD control messages are transmitted to port ***3784***, BFD echo messages use UDP port ***3785*** for both source and destination.

15.7.1.3 BFD on Port Channels

On port channels, the BFD per-link feature can be used to add resiliency to the port channel's BFD sessions. When BFD per-link is enabled, BFD considers the port channel "up" as long as any link in the port channel is functioning properly.

BFD per-link can be configured in full compliance with ***RFC 7130***, causing member ports to be removed from the port channel when their BFD micro sessions are down, or in legacy mode, which relies on the LAG itself to detect and remove unresponsive member ports. By default, BFD per-link operates in legacy mode, which allows the switch to inter-operate more effectively with older equipment, but which may drop traffic if downed links are not detected by other means. ***RFC7130*** mode allows for faster detection and removal of downed links within the port channel and can be used in situations where LACP is not supported. For the BFD session to come up, both peers must be configured in the same way.

15.7.2 BFD Configuration

To use BFD as the failure detection mechanism for a routing protocol, it must be enabled for each participating protocol.

These sections describe BFD configuration tasks:

- Configuring BFD on an Interface

- Configuring BFD on a Port Channel
- Configuring the Echo Function
- Configuring BFD for PIM
- Configuring BFD for BGP
- Configuring BFD for VRRP
- Configuring BFD for OSPF
- Configure BFD for IS-IS
- Configuring BFD Session Telemetry
- Displaying BFD Neighbor Information

15.7.2.1 Configuring BFD on an Interface

The transmission rate for BFD control packets, the minimum rate at which control packets are expected from the peer, and the multiplier (the number of packets that must be missed in succession before BFD declares the session to be down) can all be configured per interface. The values configured apply to all BFD sessions that pass through the interface.

The default values for these parameters are:

- **`transmission rate`** ***300*** milliseconds
- **`minimum receive rate`** ***300*** milliseconds
- **`multiplier`** ***3***

To configure different values for these parameters on an interface, use the **bfd interval** command.

For BFD to function as a failure detection mechanism, it must be enabled for each participating protocol.

Example

These commands set the transmit and receive intervals to ***200*** milliseconds and the multiplier to ***3*** for all BFD sessions passing through ***interface ethernet 3/20***.

```
switch(config)# interface ethernet 3/20
switch(config-if-Et3/20)# bfd interval 200 min-rx 200 multiplier 3
switch(config-if-Et3/20)#
```

15.7.2.2 Configuring BFD on a Port Channel

Basic BFD parameters are configured on a port channel as described in **Configuring BFD on an Interface** above.

Additionally, BFD can be configured in per-link mode on a port channel so that the port channel will be considered up as long as any link in the channel is up. BFD per-link can be configured in compliance with ***RFC 7130*** (causing member ports to be removed from the port channel when their BFD micro session is down), or in legacy mode for interoperability with older equipment. For the BFD session to come up, both peers must be configured in the same way (either ***RFC 7130*** or legacy mode).

Note: In ***RFC 7130*** mode, if multiple IP addresses are configured for a member of a port channel (e.g., one IPv4 address and one IPv6 address), the member will be removed from the port channel if the micro session associated with either IP address goes down.

15.7.2.2.1 Enabling BFD Per-link

To enable BFD per-link on a port channel, use the **bfd per-link** command.

Example

These commands enabled BFD per-link on port channel ***5***.

```
switch(config)# interface port-channel 5
```

```
switch(config-if-Po5)# bfd per-link
switch(config-if-Po5)#
```

15.7.2.2.2 Configuring BFD Per-link in RFC 7130 Mode

By default, BFD per-link operates in legacy mode. To enable ***RFC 7130*** mode (in which a member port is removed from the port channel when its BFD micro session is down), configure the switch as follows.

1. If you are configuring an L2 interface, specify a local L3 BFD address for the switch using the **bfd local-address** command. This is not necessary when configuring an L3 interface with an IP address configured on the port channel.
2. Enable BFD per-link on the port channel using the **bfd per-link** command.
3. Specify the L3 address of the port channel's BFD neighbor using the **bfd neighbor** command. For an L2 port channel, the address is the globally configured BFD local address on the peer switch. For an L3 port channel, the address is the IP address configured on the peer port channel.

Examples

- These commands configure BFD per-link in ***RFC 7130*** mode over an L2 port channel.

Switch 1 configuration:

```
switch1(config)# bfd local-address 10.0.0.5
switch1(config)# interface port-channel 5
switch1(config-if-Po5)# bfd per-link rfc-7130
switch1(config-if-Po5)# bfd neighbor 10.0.0.4
switch1(config-if-Po5)#
```

Switch 2 configuration:

```
switch2(config)# bfd local-address 10.0.0.4
switch2(config) #interface port-channel 5
switch2(config-if-Po5)# bfd per-link rfc-7130
switch2(config-if-Po5)# bfd neighbor 10.0.0.5
switch2(config-if-Po5)#
```

These commands configure BFD per-link in ***RFC 7130*** mode over an L3 port channel.

Switch 1 configuration:

```
switch1(config)# interface port-channel 5
switch1(config-if-Po5)# no switchport
switch1(config-if-Po5)# bfd per-link rfc-7130
switch1(config-if-Po5)# ip address 10.0.0.5/24
switch1(config-if-Po5)# bfd neighbor10.0.0.4
switch1(config-if-Po5)#
```

Switch 2 configuration:

```
switch2(config)# interface port-channel 5
switch2(config-if-Po5)# no switchport
switch2(config-if-Po5)# bfd per-link rfc-7130
switch2(config-if-Po5)# ip address 10.0.0.4/24
switch2(config-if-Po5)# bfd neighbor 10.0.0.5
```

15.7.2.3 Configuring the Echo Function

The echo function is disabled by default, and is enabled on an interface using the **bfd echo** command.

When the BFD echo function is enabled, a "slow-timer" value replaces the minimum receive interval value in BFD packets sent from the switch. The default value is 2000 milliseconds. To configure a different value for the slow-timer, use the **bfd slow-timer** command.

Examples

- These commands enable the BFD echo function on ***interface ethernet 5***. If a slow-timer value has been configured on the switch, the minimum receive rate expected from the BFD neighbor will be reset to that value; otherwise, the minimum receive rate will be set to ***2000*** milliseconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# bfd echo
switch(config-if-Et5)#
```

- This command configures BFD to expect control packets from the peer every ***10000*** milliseconds when the BFD echo function is enabled.

```
switch(config)# bfd slow-timer 10000
switch(config)#
```

15.7.2.4 Configuring BFD for PIM

The **bfd (Router-PIM Sparse-mode)** command enables or disables Bidirectional Forwarding Detection (BFD) globally for all Protocol-Independent Multicast (PIM) neighbors.

To enable or disable PIM BFD on a specific interface, use the **pim ipv4 bfd** command. The interface-level configuration supersedes the global setting.

Example

- These commands enable PIM BFD globally on the switch in the default VRF, enabling it on all PIM-SM interfaces where it is not explicitly disabled.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# bfd
switch(config-router-pim-sparse-ipv4)#
```

- These commands configure ***interface vlan 200*** to use BFD for PIM-SM connection failure detection regardless of the global PIM BFD configuration.

```
switch(config)# interface vlan 200
switch(config-if-VL200)# pim ipv4 bfd
switch(config-if-VL200)#
```

15.7.2.5 Configuring BFD for BGP

To enable or disable Bidirectional Forwarding Detection (BFD) for border gateway protocol (BGP) connections with a BGP neighbor or peer group, use the **neighbor bfd** command.

Example

These commands enable BFD failure detection for BGP connections with the neighbor at ***10.13.64.1***.

```
switch(config)# router bgp 300
switch(config-router-bgp)# neighbor 10.13.64.1 bfd
switch(config-router-bgp)#
```

15.7.2.6 Configuring BFD for VRRP

To enable or disable Bidirectional Forwarding Detection (BFD) for Virtual Router Redundancy Protocol (VRRP), use the **vrrp bfd ip** command.

When enabled, BFD provides failure detection for a 2-router VRRP system. When the master is configured with the physical IP address of the backup router, and the backup is configured with the address of the master, a BFD session is established between them. If the BFD session goes down, the backup router immediately assumes the master role.

VRRP master advertisement packets are still sent even when the BFD session is established to accommodate VRRP systems involving more than two routers.

Example

These commands enable BFD on ***interface ethernet 3/20*** for VRRP ID ***15*** with a connection to a router at IP address ***192.168.2.1***.

```
switch(config)# interface ethernet 3/20
switch(config-if-Et3/20)# vrrp 15 bfd ip 192.168.2.1
switch(config-if-Et3/20)#
```

15.7.2.7 Configuring BFD for OSPF

To enable or disable BFD globally for all OSPF neighbors, use the **bfd default (OSPF)** command in OSPF configuration mode.

To enable or disable BFD for OSPF on a specific interface, use the **ip ospf neighbor bfd** command. The interface-level configuration supersedes the global setting.

Examples

- These commands enable BFD in OSPF instance ***100*** for all OSPF neighbors on BFD-enabled interfaces except those connected to interfaces on which OSPF BFD has been explicitly disabled.

```
switch(config)# router ospf 100
switch(config-router-ospf)# bfd default
switch(config-router-ospf)#
```

- This command enables OSPF BFD on ***interface ethernet 3/21***.

```
switch(config)# interface ethernet 3/21
switch(config-if-Et3/21)# ip ospf neighbor bfd
switch(config-if-Et3/21)#
```

15.7.2.8 Configure BFD for IS-IS

The **isis bfd** and command configure Bidirectional Forwarding Detection (BFD), a low overhead protocol designed to provide rapid detection of failures at any protocol layer in the path between adjacent forwarding engines over any media. BFD is supported for IS-IS IPv4 routes.

Examples

- These commands enable BFD for all the interfaces on which IS-IS is enabled. By default BFD is disabled on all the interfaces.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv4
switch(config-router-af)# bfd default
switch(config-router-af)#
```

- These commands enable BFD on IS-IS interfaces.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# isis bfd
switch(config-if-Et5/6)#
```

15.7.2.9 Configuring BFD Session Telemetry

The BFD session telemetry automatically collects the per-session statistics and the rbfd kernel module statistics at a set interval and stores them in a shared memory where Cloud Vision Portal (CVP) or other applications may collect this information. Also, several new statistics have been added which are updated within the session-stats interval and provides a finer snapshot view of the session health. The BFD session telemetry supports both hardware-accelerated and software (kernel module) accelerated BFD sessions.

Use the **session stats snapshot interval** command to enable the BFD session telemetry. This command is configured under the ***router-bfd*** configuration mode. By default, this command is disabled and the telemetry interval is set to ***0*** seconds. A telemetry interval between ***10*** and ***3600*** seconds may be configured.

Use the **no** and **default** form of the command to disable the session stats snapshot interval command from the running configuration and sets the telemetry interval is set to .

Example

```
switch(config-router-bfd)# session stats snapshot interval 10
```

A telemetry interval may be configured to a value less than ***10*** seconds and as little as ***1*** second using an additional keyword **dangerous**, as follows:

```
switch(config-router-bfd)# session stats snapshot interval dangerous 1
```

However, note that, configurations including a telemetry interval of less than ***10*** seconds are not advised for systems with a large-scale BFD deployment as this may cause delays in the rbfd kernel module and result in BFD session instability.

15.7.2.10 Displaying BFD Neighbor Information

Use the **show bfd peers** command to display information about Bidirectional Forwarding Detection (BFD) neighbors.

Examples

- This command displays general information about BFD neighbors.

```
switch> show bfd peers
DstAddrMyDiscYoDiscIfLUpLDownLdiagState

10.168.1.561613et52_1(81)17151450 0NoDiagnosticUp

10.168.1.581714et52_2(65)17151883 0NoDiagnosticUp

10.168.1.241815et51_1(73)17152175 0NoDiagnosticUp
```

- This command displays detailed information about BFD neighbors.

```
switch> show bfd peers detail
Peer Addr 10.168.1.56, Intf Ethernet52/1, State Up
VRF default, LAddr 10.168.1.57, LD/RD 16/13
Last Up 17151450
Last Down 0
Last Diag: No Diagnostic
TxInt: 300, RxInt: 300, Multiplier: 3
Received RxInt: 300, Received Multiplier: 3
Rx Count: 433987, Tx Count: 433829
```

```
Detect Time: 900
Registered protocols: bgp

Peer Addr 10.168.1.58, Intf Ethernet52/2, State Up
VRF default, LAddr 10.168.1.59, LD/RD 17/14
Last Up 17151883
Last Down 0
Last Diag: No Diagnostic
TxInt: 300, RxInt: 300, Multiplier: 3
Received RxInt: 300, Received Multiplier: 3
Rx Count: 434235, Tx Count: 434050
Detect Time: 900
Registered protocols: bgp
```

15.7.3 Hardware Accelerated BFD Transmit

Hardware Accelerated BFD Transmit adds support for offloading BFD Transmit path to hardware (ASIC) for specific types of BFD sessions.

Hardware Accelerated BFD Transmit improves accuracy of transmit timer implementations for BFD (especially with fast timers like ***50*** ms) and relieves pressure on the main CPU in scenarios of scale. The RX packet processing for all BFD sessions is still handled by the BFD agent on the main CPU. The feature does not add any additional timer interval or multiplier configurations.

On supported platforms, hardware acceleration-capable BFD sessions is offloaded by default. Memory resources are required on ASIC to offload sessions (one unit per discriminator). Currently, the number of session discriminators that can be offloaded is restricted to ***200*** per ASIC.

Only single-hop BFD sessions on front panel ports are capable of being offloaded. In ***EOS Release 4.23.0F***, the following sessions types **cannot** be offloaded:

- BFD sessions over Port-Channel Sub-Interfaces
- BFD sessions over L3 Sub-Interfaces
- BFD sessions over an entire Port-Channel (not per member BFD)
- BFD sessions over a Switched Virtual Interface
- BFD sessions over a Loopback interface
- BFD sessions over Tunnel interfaces
- Multi-hop BFD sessions
- BFD sessions with Authentication configured

From ***EOS Release 4.23.1F*** onwards, the following additional session types **can** be offloaded:

- BFD sessions over L3 Sub-Interfaces

From ***EOS Release 4.24.0F*** onwards, hardware acceleration is only supported on certain SSO redundancy protocol configured modular systems.

Note: For the purposes of memory management in hardware, each discriminator occupies one unit of memory. When echo mode is enabled on a session, both asynchronous mode and the echo function have separate discriminators and each take up a unit each amongst the 200 available per ASIC.

15.7.3.1 Configuration

The Hardware Accelerated BFD Transmit feature is enabled by default on supported platforms. No explicit configuration is required.

The feature may be disabled using the following command:

```
switch(config-router-bfd)# hardware acceleration disabled
```

15.7.3.2 Show Commands

Hardware Acceleration Information

To view whether hardware acceleration is running use the **show bfd hardware acceleration** command. In this example, hardware acceleration is running.

Example

```
switch(config)# show bfd hardware acceleration
Hardware acceleration is running
```

The following example displays when hardware acceleration is not enabled and the reasons why.

Example

```
switch# show bfd hardware acceleration
Hardware acceleration is not running: user disabled, no eligible sessions, not
 supported with SSO
```

The reasons listed for when hardware acceleration could be a subset of the following:

- **User disabled**: The feature was explicitly disabled in the CLI configuration.
- **No eligible sessions**: There are no BFD sessions configured that could be hardware accelerated. For example, all current BFD sessions have authentication enabled.
- **Not supported with SSO**: On certain modular systems, the feature is not supported when the redundancy protocol is configured to SSO.

The following example output of the command is when the hardware acceleration feature is not supported on a product:

```
Hardware acceleration is not supported
```

Session Information

Use the **show bfd peers detail** command to display whether an individual session is hardware accelerated or not. An additional line has been added to the command to display hardware acceleration as shown in the following example.

```
switch# show bfd peers detail
VRF name: default
-----------------
Peer Addr 10.0.0.2, Intf Ethernet3/1/1, Type normal, State Down
VRF default, LAddr 0.0.0.0, LD/RD 1157402594/0
Session state is Down and not using echo function
Hardware Acceleration: Async On, Echo On
…
```

Async On denotes the Asynchronous Transmit component of the session has been offloaded.

Echo On denotes the Echo Transmit function of the session has been offloaded.

Note: There is no guarantee that both the Asynchronous transmit component and the echo function will be offloaded together to the hardware.

Hardware Acceleration Summary

Use the **show bfd hardware utilization** command to display a summary of the number of offloaded discriminators per ASIC.

Example

```
switch# show bfd hardware utilization
Chip Name  Number Of Sessions  Maximum Number Of Sessions
---------- ------------------- --------------------------
Jericho0                   20                        200
Jericho1                    0                        200
```

Detailed Hardware Acceleration Information

Use the show **bfd hardware utilization detail** command to display a detailed list of the BFD discriminators whose transmit path is offloaded can be viewed per ASIC using:

Example

```
switch# show bfd hardware utilization detail
sh bfd hardware utilization detail
Chip: Jericho0
Dst Addr  My Disc    Interface    VRF      Type
--------- ---------- ------------ -------- ------
1.1.6.2   3175653802 Ethernet1/1  default  normal
1.1.3.2   1151992021 Ethernet2/1  default  normal

Chip: Jericho1
Dst Addr   My Disc    Interface    VRF      Type
---------- ---------- ------------ -------- -----
```

15.7.3.3 Limitations

The following limitations are associated with the Hardware Accelerated BFD Transmit feature.

- Hardware acceleration is not supported on certain modular systems configured with SSO redundancy protocol.
- Before ***EOS Release 4.24.0F***, on 7500 series modular systems with both 7500E series and 7500R series line cards, hardware acceleration is not supported if the systems **Forwarding Mode** is ***Arad***, as shown by **show platform sand compatibility** command display output. From the ***EOS Release 4.24.0F*** onwards, this does not apply because the 7500E series line cards are deprecated.
- Hardware acceleration is not supported on Port-Channel Sub-Interfaces.
- In the ***EOS Release 4.23.0F***, hardware acceleration is not supported on L3 Sub-Interfaces.
- Hardware acceleration is not supported on Switched Virtual Interfaces.
- Hardware acceleration is not supported on Port-Channel interfaces (non-per-link BFD).
- Hardware acceleration is not supported on Loopback interfaces.
- Hardware acceleration is not supported on Tunnel interfaces.
- Hardware acceleration is not supported on front-panel or Port-Channel sub-interfaces.
- Hardware acceleration is not supported with Multi-hop BFD.
- Hardware acceleration is not supported when Authentication is enabled.
- Configuring authentication on an already offloaded session results in the session being migrated back to software. Similarly, deconfiguring authentication migrates the session to hardware if the session is hardware acceleration capable and if resources are available on the ASIC in question.
- Hardware acceleration limits the number of accelerated transmit discriminators to 200 per ASIC.
- Configuring more than 200 session discriminators on interfaces attached to an ASIC results in the additional session discriminators falling back to the default software transmit implementation, where the accuracy of transmit timers are dependent on the host CPU load.
- When enabling hardware acceleration with a high enough number of existing offload-capable software sessions, **a flap in some sessions may be observed once at the time of migration from software to hardware.**

- When disabling hardware acceleration with a high enough number of existing offloaded sessions, **a flap in some sessions may be observed once at the time of migration from hardware to software**.
- Sessions are offloaded to hardware in a first-come, first-served fashion. Currently, in an overflow scenario with more than 200 session discriminators per chip, there is no guarantee that shorter intervals are always offloaded.

15.7.4 BFD Commands

BFD Configuration Command

- bfd echo
- bfd interval
- bfd local-address
- bfd neighbor
- bfd per-link
- bfd slow-timer
- hardware acceleration disabled
- session stats snapshot interval (BFD)

BFD Display Commands

- show bfd hardware acceleration
- show bfd hardware utilization
- show bfd peers
- show bfd peers detail

PIM-BFD Configuration Commands

- bfd (Router-PIM Sparse-mode)
- pim ipv4 bfd

BGP-BFD Configuration Commands

- neighbor bfd

VRRP-BFD Configuration Commands

- vrrp bfd ip

OSPF-BFD Configuration Commands

- bfd default (OSPF)
- ip ospf neighbor bfd

ISIS-BFD Configuration Commands

- isis bfd
- bfd default (ISIS)

15.7.4.1 bfd (Router-PIM Sparse-mode)

The **`bfd (Router-PIM Sparse-mode)`** command enables Bidirectional Forwarding Detection (BFD) globally for use as a failure-detection mechanism for Protocol-Independent Multicast Sparse-Mode (PIM-SM) on the switch. To override the global configuration for a specific interface, use the **pim ipv4 bfd** command. All PIM-SM interfaces will use the global setting if they are not individually configured.

When PIM BFD is enabled, a BFD session is created for each PIM-SM neighbor and used to detect a loss of connectivity with the neighbor. PIM hello packets are still exchanged with PIM-SM neighbors when BFD is enabled.

The **no bfd** and **default bfd** commands disable PIM BFD globally by deleting the **bfd** statement from **running-config**. When this is done, only interfaces with PIM BFD explicitly enabled will use PIM BFD.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Command Syntax

bfd

no bfd

default bfd

Example

These commands enable PIM BFD globally on the switch in the default VRF, enabling it on all PIM-SM interfaces where it is not explicitly disabled.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# bfd
switch(config-router-pim-sparse-ipv4)#
```

15.7.4.2 bfd default (ISIS)

The **bfd default** command places the switch in address-family configuration mode.

The **bfd default** and **isis bfd** commands configure Bidirectional Forwarding Detection (BFD), a low overhead protocol designed to provide rapid detection of failures at any protocol layer in the path between adjacent forwarding engines over any media. BFD is supported for IS-IS IPv4 routes.

Command Mode

Router-Address-Family Configuration

Command Syntax

bfd default

Example

These commands enable BFD for all the interfaces on which IS-IS is enabled. By default BFD is disabled on all the interfaces.

```
switch(config)# router isis 1
switch(config-router-isis)# address-family ipv4
switch(config-router-af)# bfd default
switch(config-router-af)#
```

15.7.4.3 bfd default (OSPF)

The **bfd default** command globally configures OSPF to use Bidirectional Forwarding Detection (BFD). When this command is issued, BFD sessions will be established with all OSPF neighbors connected to BFD-enabled interfaces unless OSPF BFD has been disabled on a participating interface using the **ip ospf neighbor bfd** command. BFD is globally disabled in OSPF by default.

For OSPF BFD to function on an interface, BFD must also be enabled and configured on that interface using the **bfd interval** command.

The **no bfd default** and **default bfd default** commands disable OSPF BFD on all interfaces except those where it has been explicitly enabled using the **ip ospf neighbor bfd** command.

Command Mode

Router-OSPF Configuration

Command Syntax

bfd default

no bfd default

default bfd default

Examples

These commands enable BFD for OSPF instance ***100*** on all interfaces except those on which OSPF BFD has been explicitly disabled.

```
switch(config)# router ospf 100
switch(config-router-ospf)# bfd default
switch(config-router-ospf)#
```

15.7.4.4 bfd echo

The **bfd echo** command enables the BFD echo function on the configuration mode interface.

The **no bfd echo** and **default bfd echo** commands disable the BFD echo function by removing the corresponding **bfd echo** command from **running-config**.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

bfd echo

no bfd echo

default bfd echo

Example

These commands enable the BFD echo function on ***interface ethernet 5***. If a slow-timer value has been configured on the switch, the minimum receive rate expected from the BFD neighbor will be reset to that value; otherwise, the minimum receive rate will be set to ***2000*** milliseconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# bfd echo
switch(config-if-Et5)#
```

15.7.4.5 bfd interval

The **bfd interval** command configures the BFD control packet transmission rate, minimum control packet receive rate, and the number of missed packets that will signal that the session is down. These parameters can be configured globally for the switch or for the configuration mode interface. If a parameter is configured both globally and on the interface, the value configured on the interface takes precedence.

Note: For a BFD session to be established, BFD must be enabled for any routing protocol using BFD for failure detection.

The **no bfd interval** and **default bfd interval** commands return the BFD parameters on the configuration mode interface to default values by removing the corresponding **bfd interval** command from **running-config**.

Command Mode

Interface-Ethernet Configuration

Interface-Loopback Configuration

Interface-Management Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

bfd interval ***transmit_rate*** **min-rx** ***receive_minimum*** **multiplier** ***factor***

no bfd interval

default bfd interval

Parameters

- ***transmit_rate*** rate in milliseconds at which control packets will be sent. Values range from ***50*** to ***60000***; the default value is ***300***.
- ***receive_minimum*** rate in milliseconds at which control packets will be expected. Values range from ***50*** to ***60000***.
- ***factor*** number of consecutive missed BFD control packets after which BFD will declare the session as down. Values range from ***3*** to ***50***.

Example

These commands configure BFD on ***interface ethernet 5*** to expect packets from the peer every ***200*** milliseconds and declare the session down after failing to receive ***5*** consecutive packets. This configuration overrides any values configured globally.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# bfd interval 200 min-rx 200 multiplier 5
switch(config-if-Et5)#
```

15.7.4.6 bfd local-address

The **bfd local-address** command specifies the local L3 address for use in Bidirectional Forwarding Detection (BFD). When configuring an L2 interface, specification of a local L3 address is required in order to run BFD per-link in ***RFC 7130*** mode. (This is not necessary when configuring an L3 interface with an IP address configured on the port channel.)

The **no bfd local-address** and **default bfd local-address** commands remove the local L3 address by removing the corresponding **bfd local-address** command from **running-config**.

Command Mode

Global Configuration

Command Syntax

bfd local-address [***address***

no bfd local-address [***address***]

default bfd local-address [***address***]

Parameters

address local IPv4 or IPv6 address for BFD.

Example

This command specifies the local L3 address for BFD.

```
switch(config)# bfd local-address 10.0.0.4
switch(config#
```

15.7.4.7 bfd neighbor

The **bfd neighbor** command specifies the L3 address of the BFD neighbor of the port channel being configured. This is required to run BFD per-link in ***RFC 7130*** mode. For an L2 port channel, this address should be the BFD per-link "local address" globally configured on the peer switch. For an L3 port channel, this address should be the IP address configured on the peer port channel.

The **no bfd neighbor** and **default bfd neighbor** commands remove the BFD neighbor address by removing the corresponding **bfd neighbor** command from ***running-config***.

Command Mode

Interface-Port-channel Configuration

Command Syntax

bfd neighbor ***address***]

no bfd neighbor [***address***]

default bfd neighbor [***address***]

Parameters

address IPv4 or IPv6 address of the port channel's BFD neighbor.

Example

These commands specify the L3 address of the port channel's BFD neighbor.

```
switch(config)# interface port-channel 5
switch(config-if-Po5)# bfd neighbor 10.0.0.5
switch(config-if-Po5)#
```

15.7.4.8 bfd per-link

The **bfd per-link** command enables the BFD per-link function on the port channel being configured. When BFD per-link is enabled, BFD sub-sessions are run on each link of the port channel; BFD considers the port-channel to be up as long as any one of the links is live.

BFD per-link runs by default in legacy mode, which allows downed links to remain members of the port channel and relies on LACP or other means to prune the dead links. Legacy mode is provided for interoperability with older switches.

RFC 7130 mode runs BFD per-link in full compliance with ***RFC 7130***, and automatically removes links in down state from the port-channel, then adds them back again when they come up. Use the **rfc-7130** keyword to enable per-link in ***RFC 7130*** mode. You must also configure an L3 BFD neighbor address for each port-channel running ***RFC 7130*** per-link using the **bfd neighbor** command. When configuring an L2 interface, you must also globally configure a local L3 BFD address on the switch using the **bfd local-address** command.

For the BFD session to come up, both peers must be configured in the same way (either **rfc-7130** or legacy mode).

The **no bfd per-link** and **default bfd per-link** commands disable the BFD per-link function by removing the corresponding **bfd per-link** command from ***running-config***.

Command Mode

Interface-Port-channel Configuration

Command Syntax

bfd per-link [**rfc-7130**]

no bfd per-link [**rfc-7130**]

default bfd per-link [**rfc-7130**]

Examples

- These commands enable the BFD per-link function in legacy mode on ***port-channel 5***.

```
switch(config)# interface port-channel 5
switch(config-if-Po5)# bfd per-link
switch(config-if-Po5)#
```

- These commands globally specify a local L3 BFD address for the switch, enable the BFD per-link function in the ***rfc-7130*** mode on ***port-channel 5***, and specify the L3 address of the port channel's BFD neighbor.

```
switch(config)# bfd local-address 10.0.0.5
switch(config)# interface port-channel 5
switch(config-if-Po5)# bfd per-link rfc-7130
switch(config-if-Po5)# bfd neighbor 10.0.0.4
switch(config-if-Po5)#
```

15.7.4.9 bfd slow-timer

The **bfd slow-timer** command configures the minimum reception rate for BFD control packets which will be used if the BFD echo function is enabled. The default value is ***2000*** milliseconds.

Note: For a BFD session to be established, BFD must be enabled for any routing protocol using BFD for failure detection.

The **no bfd slow-timer** and **default bfd slow-timer** commands return the BFD slow-timer to the default value of ***2000*** milliseconds by removing the corresponding **bfd interval** command from **running-config**.

Command Mode

Global Configuration

Command Syntax

bfd slow-timer ***receive_minimum***

no bfd slow-timer

default bfd slow-timer

Parameters

receive_minimum rate in milliseconds at which control packets will be expected when the BFD echo function is enabled. Values range from ***2000*** to ***60000***; default value is ***2000***.

Examples

This command configures BFD to expect control packets from the peer every ***10000*** milliseconds when the BFD echo function is enabled.

```
switch(config)# bfd slow-timer 10000
switch(config)#
```

15.7.4.10 hardware acceleration disabled

Hardware acceleration is enabled by default on supported platforms and no explicit configuration is needed. Use the **`hardware acceleration disabled`** command to disable hardware acceleration.

Command Mode

BFD configuration mode

Command Syntax

`hardware acceleration disabled`

Example

```
switch(config-router-bfd)# hardware acceleration disabled
```

15.7.4.11 ip ospf neighbor bfd

The **`ip ospf neighbor bfd`** command enables Bidirectional Forwarding Detection (BFD) for the Open Shortest Path First protocol (OSPF) on the configuration mode interface regardless of the global settings for the OSPF instance. All OSPF neighbors associated with the interface become BFD peers, and OSPF uses BFD for failure detection.

For OSPF BFD to function on an interface, BFD must also be enabled and configured on that interface using the **bfd interval** command.

The **`no ip ospf neighbor bfd`** command disables OSPF BFD on the interface and terminates all BFD sessions with the interface OSPF peers. The **`default ip ospf neighbor bfd`** command causes the interface to follow global OSPF BFD settings configured by the **bfd default (OSPF)** command.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip ospf neighbor bfd`

`no ip ospf neighbor bfd`

`default ip ospf neighbor bfd`

Examples

- These commands enable BFD on ***interface ethernet 3/20***.

```
switch(config)# interface ethernet 3/20
switch(config-if-Et3/20)# ip ospf neighbor bfd
switch(config-if-Et3/20)#
```

- These commands cause ***interface ethernet 3/20*** to follow the global OSPF BFD configuration.

```
switch(config)# interface ethernet 3/20
switch(config-if-Et3/20)# default ip ospf neighbor bfd
switch(config-if-Et3/20)#
```

15.7.4.12 isis bfd

The **isis bfd** command activates the corresponding IS-IS routing instance on the configuration mode interface. By default, the IS-IS routing instance is not enabled on an interface.

The **no isis enable** and **default isis enable** commands disable IS-IS on the configuration mode interface by removing the corresponding **isis enable** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

isis bfd

no isis bfd

default isis bfd

Example

These commands enable BFD on IS-IS interfaces.

```
switch(config)# interface Ethernet 5/6
switch(config-if-Et5/6)# isis bfd
switch(config-if-Et5/6)#
```

15.7.4.13 neighbor bfd

The **neighbor bfd** command enables Bidirectional Forwarding Detection (BFD) for use as a failure detection mechanism for Border Gateway Protocol (BGP) connections to the specified BGP neighbor or peer group.

Once a BFD session is established with a BGP neighbor, if the BFD session goes down the status of the BGP session is changed to down as well.

The **no neighbor bfd** and **default neighbor bfd** commands disable BFD for BGP connections to the specified neighbor or peer group by removing the corresponding **neighbor bfd** command from **running-config**.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor ***NEIGHBOR_ID*** bfd

no neighbor ***NEIGHBOR_ID*** bfd

default neighbor ***NEIGHBOR_ID*** bfd

Parameters

NEIGHBOR_ID IP address or peer group name. Values include:

- ***ipv4_addr*** neighbor IPv4 address.
- ***ipv6_addr*** neighbor IPv6 address.

- ***group_name*** peer group name.

Example

These commands enable BFD failure detection for BGP connections with the neighbor at ***10.13.64.1***.

```
switch(config)# router bgp 300
switch(config-router-bgp)# neighbor 10.13.64.1 bfd
switch(config-router-bgp)#
```

15.7.4.14 pim ipv4 bfd

The **pim ipv4 bfd** command enables Bidirectional Forwarding Detection (BFD) on the configuration mode interface as a failure detection mechanism for Protocol-Independent Multicast Sparse-Mode (PIM-SM). To enable PIM BFD globally on the switch, use the **bfd (Router-PIM Sparse-mode)** command. Interface-level settings override the global setting.

When PIM BFD is enabled, a BFD session is created for each PIM-SM neighbor and used to detect a loss of connectivity with the neighbor. PIM-SM hello packets are still exchanged with PIM-SM neighbors when BFD is enabled.

The **no pim ipv4 bfd** disables PIM BFD on the configuration mode interface regardless of global settings. The **default pim ipv4 bfd** command causes the configuration mode interface to follow the global setting for PIM BFD by removing the corresponding **pim ipv4 bfd** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration I

nterface-VLAN Configuration

Command Syntax

pim ipv4 bfd

no pim ipv4 bfd

default pim ipv4 bfd

Example

These commands configure ***interface vlan 200*** to use BFD for PIM-SM connection failure detection regardless of the global PIM BFD configuration.

```
switch(config)# interface vlan 200
switch(config-if-VL200)# pim ipv4 bfd
switch(config-if-VL200)#
```

15.7.4.15 show bfd hardware acceleration

The **show bfd hardware acceleration** command displays the status of hardware acceleration on the switch for Bidirectional Forwarding Detection (BFD) session. The **show bfd peers detail** command displays the status of hardware acceleration.

Command Mode

EXEC

Command Syntax

show bfd hardware acceleration

Examples

- This command displays the status for hardware acceleration for BFD.

```
switch# show bfd hardware acceleration
Output when hardware acceleration is enabled(default) and running
Hardware acceleration is running
```

- Output when hardware acceleration is disabled with explanation.

```
Hardware acceleration is not running:
user disabled, no eligible sessions, not supported with SSO
```

- Output when hardware acceleration is not supported by switch.

```
Hardware acceleration is not supported
```

15.7.4.16 show bfd hardware utilization

The **show bfd hardware utilization** command displays the status of hardware acceleration on the switch for Bidirectional Forwarding Detection (BFD) session.

Command Mode

EXEC

Command Syntax

show bfd hardware utlization [**INFO_LEVEL**]

Parameters

INFO_LEVEL amount of information that is displayed. Options include:

- ***no parameter*** command displays a summary of offloaded discriminators per ASIC.
- **detail** command displays details of the BFD discriminators whose transmit path is offloaded per ASIC.

Examples

- This command displays a summary of offloaded discriminators per ASIC for BFD.

```
switch# show bfd hardware utilization
Chip Name          Number Of Sessions    Maximum Number Of Sessions
---------------- ----------------------- --------------------------
Jericho0                           20                          200
Jericho1                            0                          200
```

- This command displays a details list of offloaded discriminators per ASIC for BFD.

```
switch# show bfd hardware utilization detail
sh bfd hardware utilization detail
Chip: Jericho0
Dst Addr             My Disc    Interface           VRF           Type
-------------- ---------------- ------------------- ------------- ------
1.1.6.2           3175653802    Ethernet1/1         default       normal
1.1.3.2           1151992021    Ethernet2/1         default       normal

Chip: Jericho1
Dst Addr          My Disc    Interface       VRF       Type
-------------- ------------- --------------- --------- ----
```

15.7.4.17 show bfd peers

The **show bfd peers** command displays information about the neighbors with which the switch currently has a Bidirectional Forwarding Detection (BFD) session.

Command Mode

EXEC

Command Syntax

show bfd peers [**INFO_LEVEL**]

Parameters

INFO_LEVEL amount of information that is displayed. Options include:

- ***no parameter*** command displays data block for each specified interface.
- **detail** command displays table that summarizes interface data.
- **summary** displays the summary of the interface.

Display Values

- **DstAddr** IP address of the BFD neighbor.
- **MyDisc** Local discriminator value of the BFD session.
- **YoDisc** Neighbor's discriminator value for the BFD session.
- **If** Interface to which the neighbor is connected.
- **LUp** Last up.
- **LDown** Last down.
- **Ldiag** Diagnostic for the last change in session state.
- **State** State of the BFD session.
- **TxInt** Transmit interval of the local interface.
- **RxInt** Minimum receive interval set on the local interface.
- **Multiplier** Local multiplier (number of packets that must be missed to declare session down).
- **Received RxInt** Minimum receive interval set on the neighbor interface.
- **Received Multiplier** Neighbor's multiplier (number of packets that must be missed to declare session down).
- **Rx Count** BFD control packets transmitted.
- **Tx Count** BFD control packets received.
- **Detect** Time Total time in milliseconds it takes for BFD to detect connection failure.
- **Registered Protocols** Protocols using BFD with this neighbor.

Examples

- This command displays general information about BFD neighbors.

```
switch> show bfd peers
DstAddr MyDisc YoDisc If LUp LDown Ldiag S
tate
10.168.1.56 16 13 et52_1(81) 17151450 0 No
Diagnostic Up
10.168.1.58 17 14 et52_2(65) 17151883 0 No
Diagnostic Up
10.168.1.24 18 15 et51_1(73) 17152175 0 No
Diagnostic Up
10.168.254.6 19 12 vlan4094(26) 17152336 0 No
Diagnostic Up
10.168.1.26 20 16 et51_2(57) 17152523 0 No
Diagnostic Up
10.168.1.40 21 12 et50_1(77) 17152966 0 No
Diagnostic Up
10.168.1.42 22 13 et50_2(61) 17153488 0 No
Diagnostic Up
10.168.1.8 27 55 et49_1(69) 26710447 0 No
Diagnostic Up
10.168.1.10 28 56 et49_2(53) 26710847 0 No
```

```
Diagnostic Up
```

- This command displays detailed information about BFD neighbors.

```
switch> show bfd peers detail
Peer Addr 10.168.1.56, Intf Ethernet52/1, State Up
VRF default, LAddr 10.168.1.57, LD/RD 16/13
Last Up 17151450
Last Down 0
Last Diag: No Diagnostic
TxInt: 300, RxInt: 300, Multiplier: 3
Received RxInt: 300, Received Multiplier: 3
Rx Count: 433987, Tx Count: 433829
Detect Time: 900
Registered protocols: bgp

Peer Addr 10.168.1.58, Intf Ethernet52/2, State Up
VRF default, LAddr 10.168.1.59, LD/RD 17/14
Last Up 17151883
Last Down 0
Last Diag: No Diagnostic
TxInt: 300, RxInt: 300, Multiplier: 3
Received RxInt: 300, Received Multiplier: 3
Rx Count: 434235, Tx Count: 434050
Detect Time: 900
Registered protocols: bgp switch>
```

- This command displays the currently-configured BFD telemetry interval.

```
switch# show bfd peers summary
Global administrative shutdown:  No
Configured session stats snapshot interval 10s
BFD:
Configured global single hop interval 300ms min_rx 300ms multiplier 3
Configured global multiple hop interval 300ms min_rx 300ms multiplier 3
Slow timer: 2000ms
SBFD:
IPv4 operational state: globally disabled (Local interface is not configured)
Configured global initiator tx interval 300ms multiplier 3
Configured reflector rx interval 300ms


Legend:
*: pseudo LAG session (not counted in total sessions)
<N>[<M>]:  Number of sessions [ Number of sessions with echo enabled ]

Addressing    Type               Up     Init     Down   AdminDown
------------- ------------------ ------ -------- ------ ---------
All           All                1 [0]   0 [0]   0 [0]  0 [0]
IPv4          All                1 [0]   0 [0]   0 [0]  0 [0]
 single hop   All                1 [0]   0 [0]   0 [0]  0 [0]
              normal             1 [0]   0 [0]   0 [0]  0 [0]
              LAG RFC7130 *      0 [0]   0 [0]   0 [0]  0 [0]
              micro RFC7130      0 [0]   0 [0]   0 [0]  0 [0]
              LAG per-link *     0 [0]   0 [0]   0 [0]  0 [0]
              micro per-link     0 [0]   0 [0]   0 [0]  0 [0]
 multi-hop    multi-hop          0 [0]   0 [0]   0 [0]  0 [0]
IPv6          All                0 [0]   0 [0]   0 [0]  0 [0]
 single hop   All                0 [0]   0 [0]   0 [0]  0 [0]
              normal             0 [0]   0 [0]   0 [0]  0 [0]
              LAG RFC7130 *      0 [0]   0 [0]   0 [0]  0 [0]
              micro RFC7130      0 [0]   0 [0]   0 [0]  0 [0]
              LAG per-link *     0 [0]   0 [0]   0 [0]  0 [0]
              micro per-link     0 [0]   0 [0]   0 [0]  0 [0]
 multi-hop    multi-hop          0 [0]   0 [0]   0 [0]  0 [0]
Tunnel        VXLAN              0 [0]   0 [0]   0 [0]  0 [0]
L2            LAG RFC7130 *      0 [0]   0 [0]   0 [0]  0 [0]
              micro RFC7130      0 [0]   0 [0]   0 [0]  0 [0]
SR-TE Tunnel  All                   0       0       0      0
  IPv4        nitiator              0       0       0      0
              Reflector             0       0       0      0

  IPv6        Initiator             0       0       0      0
              Reflector             0       0       0      0
```

15.7.4.18 show bfd peers detail

Use the **`show bfd peers detail`** to display whether an individual session is hardware accelerated or not. An extra line has been added to to highlight hardware acceleration.

Command Mode

EXEC

Command Syntax

`show bfd peers` detail

Parameter

detail Displays a comprehansive view of the individual session.

Guidelines

- **Async On** Denotes the Asynchronous Transmit component of the session has been offloaded.
- **Echo On** Denotes the Echo Transmit function of the session has been offloaded.



Note: There is no guarantee that both the Asynchronous transmit component and the echo function are offloaded to hardware together.

Example

```
switch# show bfd peers detail
VRF name: default
-----------------
Peer Addr 10.0.0.2, Intf Ethernet3/1/1, Type normal, State Down
VRF default, LAddr 0.0.0.0, LD/RD 1157402594/0
Session state is Down and not using echo function
Hardware Acceleration: Async On, Echo On
…
```

15.7.4.19 session stats snapshot interval (BFD)

The **`session stats snapshot interval`** enables the BFD session telemetry on a switch. By default, this command is disabled and the telemetry interval is set to ***0*** seconds. A telemetry interval between ***10*** and ***3600*** seconds may be configured. A telemetry interval may be configured to a value less than ***10*** seconds and as small as ***1*** second using an additional keyword **dangerous**.



Note: Configurations including a telemetry interval of less than ***10*** seconds are not advised for systems with a large-scale BFD deployment as this may cause delays in the rbfd kernel module and result in BFD session instability.

The **`no session stats snapshot interval`** and **`default session stats snapshot interval`** commands disbales the BFD telemetery command from the **`running-config`** and sets the telemetry interval is set to ***0***.

Command Mode

Router BFD Configuration

Command Syntax

`session stats snapshot interval` ***timer_interval*** dangerous

`no session stats snapshot interval`

`default session stats snapshot interval`

Parameters

- ***timer_interval*** Session statistics timer interval in seconds between 10-3600.

- **dangerous** Set session statistics timer interval less than 10 seconds.

Examples

- The following commands places the switch in router BFD mode and enanles the BFD session telemetery. In this example a BFD telemetery session with a time interval of ***10*** seconds is configured.

```
switch(config)# router bfd
switch(config-router-bfd)# session stats snapshot interval 10
```

- In this example a BFD telemetery session with a time interval of ***1*** second is configured using a keyword **dangerous**.

```
switch(config-router-bfd)# session stats snapshot interval dangerous 1
```

15.7.4.20 vrrp bfd ip

The **vrrp bfd ip** command enables and configures Bidirectional Forwarding Detection (BFD) for Virtual Router Redundancy Protocol (VRRP) on the configuration mode interface.

When enabled, BFD provides failure detection for a 2-router VRRP system. When the master is configured with the physical IP address of the backup router, and the backup is configured with the address of the master, a BFD session is established between them. If the BFD session goes down, the backup router immediately assumes the master role.

VRRP master advertisement packets are still sent even when the BFD session is established to accommodate VRRP systems involving more than two routers.

The **no vrrp bfd ip** and **default vrrp bfd ip** commands disable BFD for VRRP on the configuration mode interface by removing the corresponding **vrrp bfd ip** statement from **running-config**. The **no vrrp** command also removes the **vrrp bfd ip** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

vrrp ***group*** bfd ip ***ipv4_address***

no vrrp ***group*** bfd ip

default vrrp ***group*** bfd ip

Parameters

- ***group*** virtual router identifier (VRID). Values range from ***1*** to ***255***.
- ***ipv4_address*** IPv4 address of the other VRRP router. On the master router, enter the physical IP address of the backup; on the backup, enter the physical IP address of the master.

Example

These commands enable BFD on ***interface ethernet 3/20*** for VRRP ID ***15*** with a connection to a router at IP address ***192.168.2.1***.

```
switch(config)# interface ethernet 3/20
switch(config-if-Et3/20)# vrrp 15 bfd ip 192.168.2.1
switch(config-if-Et3/20)#
```

Chapter 16

Multicast

IP multicast is the transmission of data packets to multiple hosts through a common IP address. Sections covered in this chapter include:

- Multicast Architecture
- IGMP and IGMP Snooping
- Protocol Independent Multicast
- Multicast Source Discovery Protocol (MSDP)
- Audio Video Bridging (AVB)

16.1 Multicast Architecture

IP multicast is the transmission of data packets to multiple hosts through a common IP address. Arista switches support multicast transmissions through IGMP, IGMP Snooping, and PIM-SM. These topics describe the Arista multicast architecture.

- Overview
- Multicast Architecture Description
- Multicast Listener Discovery (MLD)
- Multicast Route Counters
- Multicast (S,G) Counters
- Static IP Mroute
- Static Multicast
- Configuring Multicast
- Multicast Commands

16.1.1 Overview

Arista switches provide Layer 2 multicast filtering and Layer 3 routing features for applications requiring IP multicast services. The switches support over a thousand separate routed multicast sessions at wire speed without compromising other Layer 2/3 switching features. Arista switches support IGMP, IGMP snooping, PIM-SM, and MSDP to simplify and scale data center multicast deployments.

Supported Features

Feature support varies by platform; consult the release notes for multicast support information by platform.

Multicast and unicast use the same routing table. Unicast routes use TCAM resources, which may also impact the maximum number of multicast routes.

Features Not Supported

The multicast functions not supported by Arista switches include (*,*,G) forwarding or boundary routers, multicast MIBs, and router applications joining multicast groups.

16.1.2 Multicast Architecture Description

IP multicast is data transmission to a subset of all hosts through a single multicast group address. Multicast packets are delivered using best-effort reliability, similar to unicast packets. Senders use the multicast

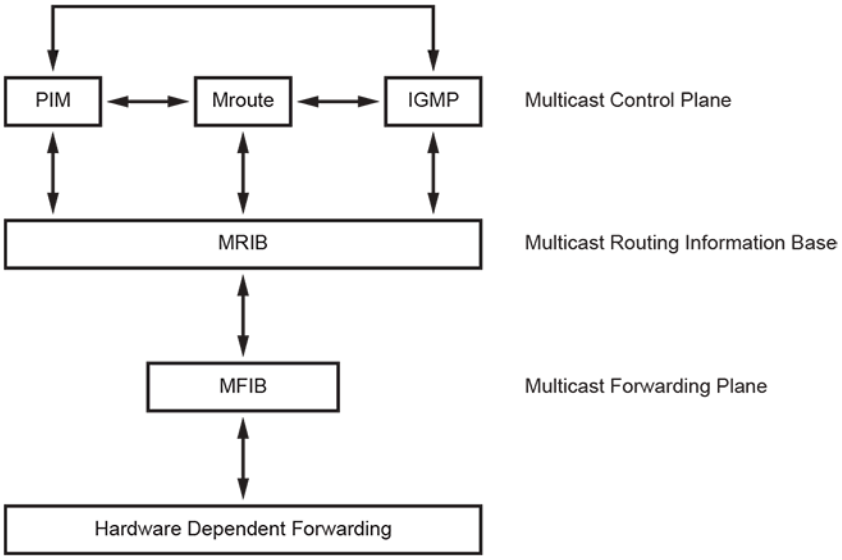
address as the destination address. Any host, regardless of group membership, can send to a group. However, only group members receive messages sent to a group address.

IP multicast addresses range from ***224.0.0.0*** to ***239.255.255.255***. Multicast routing protocol control traffic reserves the address range ***224.0.0.0*** to ***224.0.0.255***. The address ***224.0.0.0*** is never assigned to any group.

Multicast group membership is dynamic; a groups activity level and membership can vary over time. A host can also simultaneously belong to multiple multicast groups.

Multicast Architecture depicts the components that comprise the multicast architecture. The remainder of this section describes the multicast components depicted in the figure.

Figure 16-1: Multicast Architecture



16.1.2.1 Multicast Control Plane

The multicast control plane builds and maintains multicast distribution trees. It communicates changes in the multicast routing table to the MFIB for multicast forwarding.

- Protocol Independent Multicast (PIM) builds and maintains multicast routing trees using Reverse Path Forwarding (RPF) on a unicast routing table.
- Internet Group Management Protocol (IGMP) identifies multicast group members on subnets directly connected to the switch. Hosts manage multicast group membership with IGMP messages.
- The switch maintains an mroute (multicast routing) table when running PIM to provide forwarding tables used to deliver multicast packets.

The mroute table stores the states of inbound and outbound interfaces for each source/group pair (S,G). The switch discards and forwards packets on the basis of this state information. Each table entry, referred to as an mroute, corresponds to a unique (S,G) and contains:

- the multicast group address
- the multicast source address (or * for all sources)
- the inbound interface
- a list of outbound interfaces

16.1.2.2 Multicast Routing Information Base (MRIB)

The MRIB is the channel between multicast control plane clients and the multicast forwarding plane. The **`show ip mroute`** command displays MRIB entries as (*, G), (S, G), and (*, G/m) multicast entries.

MRIB entries are based on source, group, and group masks. The entries are associated with a list of interfaces whose forwarding state is described with flags. MRIB communication is based on the state change of entry and interface flags. Flags are significant to MRIB clients but are not interpreted by the MRIB.

16.1.2.3 Multicast Forwarding Plane

The multicast forwarding plane consists of the Multicast Forwarding Information Base (MFIB), a forwarding engine that is independent of multicast routing protocols.

MFIB formats PIM and IGMP multicast routes for protocol-independent hardware packet forwarding and adds them to the hardware Multicast Expansion Table (MET) and the hardware FIB.

MFIB uses a core forwarding engine for interrupt-level (fast switching) and process-level (process switching) forwarding. MFIB fast-switches inbound multicast packets that match an MFIB forwarding entry and process-switches packets requiring a forwarding entry if a matching entry does not exist.

16.1.2.4 Hardware Dependent Forwarding and Fast Drop

In IP multicast protocols, each (S,G) and (*,G) route corresponds to an inbound Reverse Path Forwarding (RPF) interface. Packets arriving on non-RPF interfaces may require PIM processing, as performed by the CPU subsystem software.

By default, hardware sends all packets arriving on non-RPF interfaces to the CPU subsystem software. However, the CPU can be overwhelmed by non-RPF packets that do not require software processing. The CPU subsystem software prevents CPU overload by creating a fast-drop entry in hardware for inbound non-RPF packets not requiring PIM processing. Packets matching a fast-drop entry are bridged in the ingress VLAN but not sent to the software, avoiding CPU subsystem software overload. Fast-drop entry usage is critical in topologies with persistent RPF failures.

Protocol events, such as links going down or unicast routing table changes, can change the set of packets that can be fast dropped. Packets that were correctly fast dropped before a topology change may require forwarding to the CPU subsystem software after the change. The CPU subsystem software handles fast-drop entries that respond to protocol events so that PIM can process all necessary non-RPF packets.

16.1.3 Multicast Listener Discovery (MLD)

Networks use Multicast Listener Discovery (MLD) to control the flow of layer 3 IPv6 multicast traffic. Hosts request and maintain multicast group membership through MLD messages. Multicast routers use MLD to maintain a membership list of active multicast groups for each attached network.

With respect to each of its attached networks, a multicast router is either a querier or non-querier. Each physical network contains only one querier. A network with more than one multicast router designates the router with the lowest IP address as its querier.

In an MLD Report or Done message, the multicast address field holds a specific IPv6 multicast address to which the message sender is listening or is ceasing to listen, respectively.

16.1.3.1 MLDv2 Snooping

MLDv2 Snooping optimizes the transmission of multicast packets in Layer 2 by using Layer 3 information contained in MLDv2 and PIM packets. MLDv2 is the protocol used to manage the membership of hosts in multicast groups for IPv6.

RFC 3810 talks about MLDv2 functionality. MLDv2 is the IPv6 counterpart of IGMPv3. Beginning with ***EOS Release 4.25.0F***, MLDv2 snooping is supported on MLAG deployments.

16.1.3.1.1 Limitations

- Extraneous “Switch” interface should be ignored in **`show mld snooping counters`** and **`show mld snooping counters errors`** command outputs.

- MLDv2 Snooping with EVPN is not supported.
- On CCS-720XP, CCS-750, DCS-7050CX3, DCS-7050SX3, DCS-7050TX3 and DCS-7300X3 for MLDv2 snooping to be enabled on a vlan, IGMP snooping needs to be enabled on it too for IPv6 unknown multicast traffic to be forwarded over IPv4 and IPv6 mrouter ports, else such traffic would be flooded to entire vlan. Forwarding IPv6 unknown multicast traffic over only IPv6 mrouter ports is not supported on these platforms.

16.1.3.2 Configuring Multicast Listener Discovery

16.1.3.2.1 Enabling MLD

Use **mld** command to enable MLD on an interface. When the switch fills the multicast routing table, it only adds interfaces when the interface receives join messages from downstream devices or when the interface is directly connected to a member of the MLD group. By default, MLD is disabled on an interface.

Examples

- This command enables MLD on the ***interface Ethernet 1***.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# mld
```

- This command disables MLD on the ***interface Ethernet 1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# no mld
```

16.1.3.2.2 Configuring MLD

An interface that runs MLD uses default protocol settings unless otherwise configured. The switch provides commands that alter startup query, last member query, and normal query settings.

MLD

The switch supports MLD versions 1 through 2. The **mld** command configures multicast routers on the configuration mode interface. Version 2 is the default MLD version.

Example

This command enables MLD on ***interface Ethernet 1***.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# mld
```

Startup Query

Membership queries are sent at an increased frequency immediately after an interface starts up to quickly establish the group state. Query count and query interval commands adjust the period between membership queries for a specified number of messages.

The **mld startup-query-interval** command specifies the interval between membership queries that an interface sends immediately after it starts up. The **mld startup-query-count** command specifies the number of queries that the switches sends from the interface at the startup interval rate.

Examples

- This command configures the startup query count of ***4*** on ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
```

```
switch(config-if-Et1)# mld startup-query-count 4
```

- This command configures the startup query interval of ***100*** seconds on ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld startup-query-interval 100
```

Membership Queries

The router with the lowest IP address on a subnet sends membership queries as the MLD querier. When a membership query is received from a source with a lower IP address, the router resets its query response timer. Upon timer expiry, the router begins sending membership queries. If the router subsequently receives a membership query originating from a lower IP address, it stops sending membership queries and resets the query response timer.

The **mld query-interval** command configures the frequency at which the active interface, as an MLD querier, sends membership query messages.

The **mld query-response-interval** command configures the time that a host has to respond to a membership query.

Examples

- This command configures the query interval of ***30*** seconds on ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld query-interval 30
```

- This command configures the query response interval of ***30*** seconds on ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld query-response-interval 30
```

Last Member Query

When the querier receives an MLD leave message, it verifies the group has no remaining hosts by sending a set of group-specific queries at a specified interval. If the querier does not receive a response to the queries, it removes the group state and discontinues multicast transmissions.

The **mld last-listener-query-count** command specifies the number of query messages the router sends in response to a group-specific or group-source-specific leave message.

The **mld last-listener-query-interval** command configures the transmission interval for sending group-specific or group-source-specific query messages to the active interface.

Examples

- This command configures the last listener query count to ***3*** on ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld last-listener-query-count 3
```

- This command configures the last listener query interval to ***2*** seconds on ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld last-listener-query-interval 2
```

Static Groups

The **mld static-group** command configures the configuration mode interface as a static member of the multicast group at the specified address. The router forwards multicast group packets through the interface

without otherwise appearing or acting as a group member. No interface is a static member of a multicast group by default.

Example

This command configures static groups on ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld static-group ff30::1 a::1
```

16.1.3.3 MLDv2 Snooping Configuration

16.1.3.3.1 Enabling or Disabling MLDv2 Snooping

You can configure MLDv2 snooping globally and per VLAN. A configuration mode is available for MLD related snooping commands in the ***global*** configuration mode.

Example

```
switch(config)# mld snooping
switch(config-mld-snooping)# disabled
switch(config-mld-snooping)# vlan 1-100
switch(config-mld-snooping)# vlan 101
switch(config-mld-snooping-vlan-101)# disabled
```

Items to consider:

- You must explicitly configure VLANs inside the ***mld-snooping*** configuration mode to enable snooping for those VLANs.
- Use the global **disabled** to disable the mld-snooping for all VLANs. **disabled** is *not* configured by default.
- Snooping can be explicitly **disabled** for a given VLAN. This is useful once there is configuration within the VLAN. **disabled** is not configured by default.
- Snooping cannot be explicitly enabled for a given VLAN if it is disabled globally.
- Beginning with the ***EOS Release 4.25.0F***, it allows the configuration of MLDvs snooping on PIM non DR (Designated Router) VLANs.

16.1.3.3.2 Static Groups and Multicast Router

To configure static groups, refer to the following commands:

```
switch(config)# mld snooping
switch(config-mld-snooping)# vlan 100
switch(config-mld-snooping-vlan-100)# member ipv6-group-addr
 interface intfs
```

To configure a static multicast router, refer to the following commands:

```
switch(config)# mld snooping
switch(config-mld-snooping)# vlan 100
```

```
switch(config-mld-snooping-vlan-100)# multicast-router interface intfs
```

In the examples above, ***intfs*** can be a group of similar interfaces (for example, ***Ethernet1-5,8*** or ***port-channel 1-5,8***).

16.1.3.4 Displaying MLDv2 Snooping Status

The following show commands display the MLD snooping status of the switch:

- ```
  switch# show mld snooping
     Global MLD Snooping configuration:
  -----------------------------------------
  MLD snooping                    : Enabled
  Robustness variable             : 2

  VLAN 1 :
  ----------
  MLD snooping                    : Disabled
  MLD max group limit             : No limit set
  Recent attempt to exceed limit : No
  MLD snooping pruning active     : False
  Flooding traffic to VLAN        : True
  VLAN 100 :
  ----------
  MLD snooping                    : Enabled
  MLD max group limit             : No limit set
  Recent attempt to exceed limit : No
  MLD snooping pruning active     : True
  Flooding traffic to VLAN        : False
  ```
- ```
  switch# show mld snooping
     Global MLD Snooping configuration:
  -----------------------------------------
  MLD snooping                    : Enabled
  Robustness variable             : 2

  VLAN 1 :
  ----------
  MLD snooping                    : Disabled
  MLD max group limit             : No limit set
  Recent attempt to exceed limit : No
  MLD snooping pruning active     : False
  Flooding traffic to VLAN        : True
  VLAN 100 :
  ----------
  MLD snooping                    : Enabled
  MLD max group limit             : No limit set
  Recent attempt to exceed limit : No
  MLD snooping pruning active     : True
  Flooding traffic to VLAN        : False
  ```
- Use the **`show mld snooping groups`** command to display MLDv2 snooping by group.

  ```
  switch# show mld snooping groups
  IGMP Snooping Group Membership
  EX : Filter mode Exclude
  IN : Filter mode Include
  IR : Ingress Replication
  VLAN  Group            Members
  ----  ---------------  ------------------------
  ```

```
100   ff05::2          Cpu, Et4
100   ff08::11         Et1
100   ff08::21         Et2
100   ff08::31         Et3
100   *                Et3, Et4
```

- Use the **show mld snooping groups detail** command to display detailed MLDv2 snooping information.

```
switch(config-mld-snooping-vlan-100)# show mld snooping groups detail
IGMP Snooping Group Membership
EX : Filter mode Exclude
IN : Filter mode Include
IR : Ingress Replication
VLAN  Group            Source     Mode  Uptime       Members
----  ---------------  ---------  ----  -----------  ----------
100   ff05::2          *           -     0d00h13m20   Cpu, Et4
100   ff08::11         *           -     0d00h13m17   Et1
100   ff08::21         1::1        IN    0d00h13m20   Et2
100   ff08::31         2::1        EX    0d00h01m31   Et3
100   *                *           -     -            Et3, Et4
```

The **show mld snooping groups** output can be further filtered to output just the

- **local** groups, that is groups learned locally using the **show mld snooping groups local** command.
- **user** groups, that is groups configured by user.
- **mlag** groups, that is groups learned from the MPLAG peer.

-
```
switch# show mld snooping mrouter
Vlan    Interface-ports
---------------------------------------
100     Et3(dynamic), Et4(static)
```

-
```
switch# show mld snooping mrouter detail
Vlan   Intf      Address           FirstHeard   LastHeard    Expires  Type
------------------------------------------------------------------------
--------
100    Et3       fe80::200:3ff:fe01:0 0d00h24m30  0d00h00m08  00h00m17
 querier
100    Et4       fe80::200:3ff:fe03:0 0d00h24m10  0d00h00m08  00h01m37 pim
100    Et2       0.0.0.0           -           -           -
 static
```

-
```
switch# show mld snooping mrouter vlan 100
Vlan    Interface-ports
---------------------------------------------------------
100     Et2(static), Et3(dynamic), Et4(dynamic)
```

-
```
switch# show mld snooping querier
Vlan  IP Address               Version  Port
-------------------------------------------
100   fe80::200:3ff:fe01:0     v2       Et3
```

-
```
switch# show mld snooping querier vlan 100
IP Address           : fe80::200:3ff:fe01:0
MLD Version          : v2
Port                 : Et3
Max response time    : 10.0
```

-
```
switch# show mld snooping counters
```

```
                 Input                    |    Output
Port   Queries Reports  Others  Errors|Queries Reports  Others
-------------------------------------------------------------
Cpu          1     153       0       0     154       6      51
Et1          0     153       0       0     156       0      51
Et2          0     153       0       0     156     110      51
Et3        154       1       0       0       1     610      51
Et4          0     152      51       0     155     453       0
Switch       0       0       0       0       0       0       0
```

-
```
switch# show mld snooping counters errors
          Packet       Packet  Bad IP    Unknown      Bad PIM   Bad ICMP  Bad MLD   Bad MLD
Port      Too Short    Not IP  Checksum  IP Protocol  Checksum  Checksum  Query     Report
-------------------------------------------------------------------------------------------
Cpu               0         0         0            0         0         0         0         0
Et1               0         0         0            0         0         0         0         0
Et2               0         0         0            0         0         0         0         0
Et3               0         0         0            0         0         0         0         0
Et4               0         0         0            0         0         0         0         0
Switch            0         0         0            0         0         0         0         0
```

16.1.4 Multicast Route Counters

Multicast Route Counters provides per multicast route ingress packet and byte counters for multicast routed packets.

16.1.4.1 Configuration

To enable multicast route ingress packet and byte counters use the following command:

```
switch(config)# hardware counter feature multicast ipv4
```

To disable multicast route ingress packet and byte counters use the following command:

```
switch(config)# no hardware counter feature multicast ipv4
```

On DCS-7020R, DCS-7280R/R2 and 7500R/R2 platforms, create and apply user defined TCAM profile with multicast counter feature db enabled. This example reates a profile copying it from the default, then modifying the profile.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile <profileName> copy default
switch(config-hw-tcam-profileName)# feature counter multicast ipv4
switch(config-hw-tcam-profileName-feature-counter-multicast-ipv4))# exit
switch(config-hw-tcam-profileName)# no feature mirror ip
switch(config-hw-tcam-profileName)# exit
switch(config-hw-tcam)# system profile <profileName>
```

16.1.4.2 Show Commands

Use the **show ip mfib counters** command to display per multicast route ingress packet and byte counters:

```
switch# show ip mfib 225.1.2.1 10.46.1.2 counters
Activity poll time: 60 seconds
   225.1.2.1 10.46.1.2
      Byte: 1200200
      Packet: 12002
      Ethernet46 (iif)
```

```
      Ethernet47
      Activity 0:02:52 ago switch#
```

```
switch# show ip mfib counters
Activity poll time: 60 seconds
   225.1.1.1 0.0.0.0/0
      Byte: 234100
      Packet: 2341
      1 (rpaIndex)
      Ethernet21
      Ethernet23
      Activity 1:00:52 ago
   225.1.2.1 10.46.1.2
      Byte: 1200200
      Packet: 12002
      Ethernet46 (iif)
      Ethernet47
      Activity 0:02:52 ago
   224.0.0.0/4 10.45.1.0/24
      Byte: N/A
      Packet: N/A
      Ethernet45 (iif)
      Cpu
      Activity 0:02:04 ago switch#
```

To clear all counters including per multicast route ingress counters, use the **clear counters** command.

```
switch# clear counters
```

To clear per multicast route ingress counters or clear counters for a particular multicast route, use the **clear ip multicast counters** command.

```
switch# clear ip multicast <vrf [vrf-name]> counters <group_addr>
 <source_addr>
```

16.1.4.3 Limitations

This feature contains the following limitations:

- Counters are not supported on fastdrop multicast routes.
- For Pim sparse mode, counters are supported only on (***Source_addr***, ***Group_addr***) multicast routes.
- Counting is supported for IPv4 multicast packets only.

16.1.5 Multicast (S,G) Counters

The Multicast Route Counters count the packets and bytes per group, source, and VRF. Every multicast route is counted when the Multicast (S,G) counters is enabled, and if there are sufficient hardware counter resources available. Since the number of hardware counter resources is limited, selected multicast routes can

be **prioritized** to provide them the needed hardware counter resources over the **non-priority** multicast routes in case of resource contention.

Priority multicast routes can be configured if:

- Hardware counter resources are available:
  - The priority multicast routes do not affect the existing non-priority multicast route counters.
- In case there are insufficient hardware counter resources:
  - If there are non-priority multicast routes, one non-priority multicast route counter resource will be freed, which results in losing the counters for the non-priority multicast route, and the priority multicast route will be programmed, meanwhile the non-priority multicast route will be pending to be programmed until there are hardware counter resources available.
  - If there are only priority multicast groups, the newly configured priority multicast groups will wait till the resources are freed by existing priority groups.

16.1.5.1 Configuring Multicast (S,G) Counters

- The multicast (S, G) counters feature is configured using the following command in the configuration mode:

  `hardware counter feature multicast` [ipv4 | ipv6]

  The **`hardware counter feature multicast`** command enables counting for all groups as non-priority groups unless the groups that require priority treatment have been configured under router multicast as described in the next paragraph. Use the **no** form of the command to disable the multicast (S, G) counters on a switch.
- Certain multicast routes can be configured as priority multicast routes for counting purposes, and they will be counted with higher priority. In case of running out of hardware counter resources, this could result in deleting the existing multicast (S, G) counters. The following configuration shows how the IPv6 version is configured, it's similar to IPv4. Use the **`no counters`** command to disable the priority routes.

  ```
  switch(config)# router multicast
  switch(config-router-multicast)# ipv6
  switch(config-router-multicast-ipv6)# counters
  switch(config-router-multicast-ipv6)# counters ff08::e101:101 2002::a01:101
  ```

  Of the above configuration steps, the following command is required before configuring the specific multicast routes:

  ```
  switch(config-router-multicast-ipv6)#
  ```

  Otherwise the system prompts the following error:

  ```
  ! 'counters' not configured, packet will not be counted
  ```

- Clear counters:

  `clear ipv6 multicast` [vrf [***vrf-name***]] counters ***group source***

  The above CLI command can either clear the counters for all multicast routes or the specific (vrf, group, source). The default VRF will be used if it is not specified. All the IPv6 counters are cleared if no group and source are specified. The IPv4 version of the command is:

  `clear ip multicast` [vrf [***vrf-name***]] counters ***group source***

  The below command clears the counters of all multicast routes along with all other counters.

  ```
  switch# clear counters
  ```

16.1.5.2 Multicast (S,G) Counters Show Commands

- The hardware counter resources are allocated if the Multicast (S,G) Counters is configured. The configuration state is shown as the following: **Mcast** is listed in the **Feature ID** field and the **Counters** column indicates the total number of hardware counter resources assigned to the Multicast (S,G) Counters:

```
switch# show platform trident flexcounters summary
Feature ID   Type        Request ID   Pool ID    Start Index   Counters
-------------------------------------------------------------------------
Mcast        ingress     0            0          1             8191
```

- The current counter resource allocation for each multicast route is shown using the following CLI command.The **Pool ID** column indicates which pool is used, and the **Base Counter Index** indicates the entry index in that pool.

```
switch# show platform trident flexcounters multicast summary
Group           Source         VRF  VLAN  Counter Index  Pool ID  Offset Mode Base Counter
 Index
--------------- -------------- ---- ----- -------------- -------- ----------- -------------
-----
ff08::e101:101  2002::a01:105  0    2570  1              0        0           1
225.1.1.1       10.1.1.5       0    2571  2              0        0           2
```

- The raw counter value of the each multicast route is shown using the following command. The **cntTbl** reflects the hardware table, the counters are copied from **ctnTbl** to **snapTbl** when the **clear counter** command is issued. The difference between **ctnTbl** and ***snapTbl*** is the current counter.

```
switch# show platform trident flexcounters multicast values
Group           Source       VRF VLAN Counter Index  Offset Bytes (cntTbl)  Bytes (snapTbl) Pkts (cntTbl)
 Pkts (snapTbl)
--------------- ------------- --- ---- -------------- ---------------------  --------------- -------------
 --------------
ff08::e101:101 2002::a01:105 0   2570 1              6600                    0               100           0
225.1.1.1      10.1.1.5      0   2571 2              8800                    0               100           0
```

- The multicast (S,G) counters is shown using the following CLI command. In the following example **Byte** and **Packet** are the counters:

```
switch# show multicast fib ipv6 <vrf [vrf-name]> <group> <source> counters
Activity poll time: 60 seconds
ff08::e101:101 2002::a01:101
    Byte: 66
    Packet: 1
    Vlan2780 (iif)
    Ethernet6/4
    Vlan2899

switch# show multicast fib ipv4 <vrf [vrf-name]> <group> <source> counters
Activity poll time: 60 seconds
255.1.1.1 10.1.1.5
    Byte: 66
    Packet: 1
    Vlan2781 (iif)
    Ethernet8/4
    Vlan2999
```

16.1.5.3 Limitations

The number of counters for which the hardware counters can be enabled simultaneously will be limited by the availability of counter hardware resources in the system.

When the configured hardware features exceed the available counter resources not all counters for all features will be available, the following CLI command shows the current allocation of the hardware resources:

```
switch# show platform trident flexcounters summary
Feature ID   Type       Request ID   Pool ID   Start Index   Counters
-------------------------------------------------------------------
Mcast        ingress    0            0         1             8191
```

16.1.6 Static IP Mroute

The Static IP Multicast route (or Static Mroute) interface overrides the interface that is ordinarily selected from the matching route in the unicast routing table, providing a means for breaking dependency on the unicast topology for the multicast topology. Let us assume that, PIM routers in a multicast network sends PIM joins towards a source to receive traffic from that source. The interface on which to send a PIM join is determined by looking up the unicast routing table for the source address. This interface is the upstream or RPF interface for that source. When traffic is received from that source, it is ensured it is received on the RPF interface for that source. This mechanism causes multicast traffic to take the same path through a network as unicast traffic would. In some cases, it is desirable to have the multicast traffic take a different path than the unicast traffic. For example, to avoid a slow firewall required for unicast traffic but not for multicast traffic or to receive multicast across a low latency, low bandwidth microwave link while unicast traverses a higher latency, higher bandwidth fiber path.

To overcome this situation, a static IP multicast route (or Static Mroute) command **`ip mroute`** command is introduced. The **`ip mroute`** command specifies a candidate for the RPF interface of any (S,G) multicast route where the source falls within the given source/mask. This interface potentially overrides the interface that would ordinarily have been selected from the matching route in the unicast routing table. This command, therefore, provides a means of breaking the dependence of the multicast topology on the unicast topology. The method of selecting the RPF interface for an (S,G) route is described next.

Example

This command configures a Static IP mroute for a source ***1.1.1.1/32*** with an administrative distance ***20*** on ***interface ethernet 2/1***.

```
switch(config)# ip mroute 1.1.1.1/32 ethernet 2/1 20
```

16.1.6.1 Selecting Static Mroute

The Static Mroute is selected based the following parameters:

- Longest Match
- Administrative Distance
- Interface Status

Longest Match

When a given source matches multiple static Mroutes in the MRIB, the longest match will be selected. The order in which the static Mroutes were configured will not be a factor.

Example

If the following static mroutes were configured in order:

```
ip mroute 0.0.0.0/0 Ethernet1
ip mroute 192.168.0.0/16 Ethernet2
ip mroute 192.168.1.0/24 Ethernet3
```

For an (S,G) route where S = ***192.168.1.1***, the third static mroute listed above would be selected since it is the most specific route to the source. The RPF interface would therefore be ***Ethernet3***. The table below shows the selected RPF interface for three different sources based on the configuration above:

Source	RPF Interface
192.168.1.1	Ethernet 3
192.168.1.2	Ethernet 3
192.168.2.1	Ethernet 2
10.0.0.1	Ethernet 1

Administrative Distance

User is allowed to specify an administrative distance with each static Mroute. While selecting a Static Mroute for a source, if multiple Static Mroute exist in the MRIB with the same source/mask, then, the one with the lowest Admin distance is selected. The default administrative distance for a Static Mroute is ***1***.

Interface Status

For a Static Mroute to be considered for selection, the specified interface must be UP and PIM must be enabled on it.

16.1.6.2 Selecting RPF interface

Static Mroutes are BGP IP Multicast (SAFI 2) learned routes. These routes are stored in the Multicast Routing Information Base (MRIB), a separate routing table. The RPF interface is selected for a source as follows:

Initially, a source route is looked up in the MRIB. If the MRIB lookup yields a route, that route is used for selecting the RPF interface. Therefore, any configured Static Mroutes matching the source wins the selection process over a 'Connected' route to the source. For a static mroute to be considered for selection, the specified interface must be up and PIM must be enabled on it. By default, Static Mroute have an Admin distance of ***1***. If multiple Static Mroutes exist with equal longest prefix match, the mroute with the lowest Admin distance will win. Admin distance is not be used to compare selection between unicast RIB and MRIB routes. Successful Static Mroute looked up in the MRIB are always chosen over unicast RIB lookups.

If MRIB lookup does not yield a route, then the unicast RIB is looked up for a route to select the RPF interface. If the selected route has ECMP, one of the corresponding paths is selected as RPF neighbor.



Note: The path to choose RPF neighbor is selected based on the hashing scheme; and protocols specified for valid paths, multi-path configuration, directly connected sources, and assert winners.

Example

Let us assume that the Static Mroute is configured, and for this example let us consider that the default Admin distance for connected routes is ***0***, ***1*** for static routes, and ***110*** for OSPF routes.

```
ip mroute 172.16.0.0/16 Ethernet1
ip mroute 192.168.0.0/16 Ethernet2
ip mroute 192.168.1.0/24 Ethernet3
ip mroute 192.168.1.0/24 Ethernet4 255
ip mroute 192.168.1.3/32 Ethernet5 255
ip mroute 200.10.0.0/16 Ethernet5
ip mroute 200.11.0.32/16 Ethernet5
```

So the MRIB table contains the following:

Prefix	Interface	Admin Distance
172.16.0.0/16	Ethernet 1	1
192.168.0.0/16	Ethernet 2	1
192.168.1.0/24	Ethernet 3	1
192.168.1.0/24	Ethernet 4	255
192.168.1.3/32	Ethernet 5	255
200.10.0.0/16	Ethernet 5	1
200.11.0.1/32	Ethernet 5	1

And let us assume the unicast RIB table contains the following:

Prefix	Interface	Protocol	Admin Distance
10.0.0.0/24	Ethernet 6	OSPF	110
172.16.1.0/24	Ethernet 7	OSPF	110
192.168.0.0/16	Ethernet 8	OSPF	110
192.168.1.0/24	Ethernet 9	Static	1
192.168.1.3/32	Ethernet 10	OSPF	110
200.10.0.0/16	Ethernet 11	Connected	0

The table below shows the RPF interface selections for a set of sources along with which Static Mroute was chosen, which unicast RIB route was chosen, which was the eventual winner, and the reasoning behind the selection.

Source	Static Mroute	Unicast Route	Winner	RPF Interface	Reasoning
10.0.0.1	-	1	Unicast Route	Ethernet 6	• Only the unicast Rib yields a route to the source so it wins
200.11.0.1	7	-	Static Mroute	Ethernet 5	• Only the MRIB yields a route to the source so it wins
192.168.1.3	5	5	Unicast Route	Ethernet 10	• In the MRIB (5) is the longest match • While comparing the static mroute and the unicast route, the unicast route is the winner because it has a lower Admin distance
192.168.2.1	2	3	Static Mroute	Ethernet 2	• In the MRIB (2) is the longest match • While comparing the static mroute and the unicast route, the static Mroute is the winner because it has a lower Admin distance
192.168.1.1	3	4	Static Mroute	Ethernet 3	• In the MRIB both (3) and (4) are the longest match but (3) has the lower Admin distance • While comparing the static mroute and the unicast route, the static mroute is the winner even though both have the same distance

16.1.7 Static Multicast

Static Multicast enables to configure multicast routes statically on any Arista switches and even at per VRF granularity. However, static multicast routes do not, at this point, perform any VRF validation for the interfaces involved in the route, which can cause some possibility of route leakage. The Static Multicast co-exists with PIM-SM and PIM-BIDIR protocols which are dynamic variants of programming multicast routes. However, choose the appropriate route selection method before the static routes are programmed to obtain the best results.

Static multicast routes compete with the routes provided by PIM-SM and PIM-BIDIR mainly because this static variant, allows the operator to configure a PIM-SM-like or PIM-BIDIR-like route among other routes. The route chosen depends upon setting up the priority on the route.

At present, PIM-SM/PIM-BIDIR installs routes with a priority value of 0 (zero) while static route installs routes with a priority of ***255***, by default.

The priorities of PIM-SM and PIM-BIDIR are subject to future change. Higher priority wins in case of conflicts while programming the hardware. Conflicts here imply the same route present as static multicast route and in PIM-SM/PIM-BIDIR. It concludes that, by default, static multicast routes will always be the winner.

16.1.8 Configuring Multicast

This section describes the following configuration tasks:

- Enabling IPv4 Multicast Routing
- Enabling IPv6 Multicast Routing
- Multicast-Routing Configuration Example
- Multicast Boundary Configuration
- Multicast multipath router-ID (IPv4)
- Configuring MFIB
- Configuring Static Multicast
- Displaying and Clearing the Mroute Table

16.1.8.1 Enabling IPv4 Multicast Routing

Enabling IPv4 multicast routing allows the switch to forward multicast packets. The **routing** command enables multicast routing. When multicast routing is enabled, ***running-config*** contains a **routing** statement.

Example

These commands enable IPv4 multicast routing on the switch.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# routing
```

16.1.8.2 Enabling IPv6 Multicast Routing

Enabling IPv6 Multicast routing allows the switch to distribute IPv6 datagrams to one or more recipients. IPv6 PIM builds and maintains Multicast routing using Reverse Path Forwarding (RPF) based on unicast routing table. IPv6 PIM is protocol-independent and can use routing tables consisting of OSPFv3, IPv6 BGP or static routes, for RPF lookup. MLD is used to discover Multicast hosts and maintain group membership on directly attached link. This feature is supported on 7280R and 7500R. Source-specific multicast (SSM) is currently supported on L3 routed port.

PIM Sparse Mode

In PIM-SM, each host (sender and/or receiver) is associated with a Designated Router (DR) which acts for all directly connected hosts in PIM-SM transactions. Upon receiving of MLD report from host or PIM join from downstream PIM neighbor, (S,G) route is created or programmed and router sends a PIM join to upstream PIM neighbor with shortest path to the source.

Configuring IPv6 Multicast Routing

The following steps are to configure IPv6 multicast routing on the switch.

1. Enabling IPv6 multicast-routing. By default, multicast-routing is disabled on switch. The following commands are used to enable IPv6 multicast-routing.

   Example

   ```
   switch(config)# router multicast
   switch(config-router-multicast)# ipv6
   switch(config-router-multicast-ipv6)# routing
   ```

2. Enabling IPv6 PIM Sparse Mode.

By default, IPv6 PIM is disabled on an interface. The **pim ipv6 sparse-mode** command enables an interface to participate in IPv6 multicast-routing domain.

Example

```
switch(config)# interface ethernet 15/1
switch(config-if-Et15/1)# pim ipv6 sparse-mode
```

Note: SVI is not supported.

3. Enabling MLD.

   By default, MLD is disabled on an interface. Enabling MLD is needed only on the interface that is connected to the MLD host which would like to receive IPv6 multicast traffic.

   Example

```
switch(config)# interface ethernet 15/1
switch(config-if-Et15/1)# mld
```

16.1.8.3 Multicast-Routing Configuration Example

```
router multicast
   ipv6
      Routing
interface Ethernet15/1
   no switchport
   ipv6 enable
   ipv6 address 40:1::3/64
   mld
   pim ipv6 sparse-mode
```

Displaying IPv6 Multicast Routing Information

- The **show mld membership** command displays the MLD group membership table as shown.

```
switch# show mld membership
Interface            Group               Source

Filter Mode
--------------------------------------------------
Ethernet2/1.1        ff33::1:0:0:1       101:1::2

include
Ethernet2/1.1        ff33::1:0:0:2       101:1::2

include
Ethernet2/1.1        ff33::1:0:0:3       101:1::2

include
Ethernet2/1.1        ff33::1:0:0:4       101:1::2

include
Ethernet2/1.1        ff33::1:0:0:5       101:1::2

include
Ethernet2/1.1        ff33::1:0:0:6       101:1::2

include
```

```
Ethernet2/1.1        ff33::1:0:0:7      101:1::2

include
Ethernet2/1.1        ff33::1:0:0:8      101:1::2

include
Ethernet2/1.1        ff33::1:0:0:9      101:1::2

include
Ethernet2/1.1        ff33::1:0:0:a      101:1::2

include
Ethernet2/1.1        ff33::1:0:0:b      101:1::2

include
```

- The **show pim ipv6 sparse-mode route** command displays the PIM Sparse Mode Multicast Routing table as shown.

```
switch# show pim ipv6 sparse-mode route
PIM Sparse Mode Multicast Routing Table
Flags: E - Entry forwarding on the RPT, J - Joining to the SPT
    R - RPT bit is set, S - SPT bit is set, L - Source is attached
    W - Wildcard entry, X - External component interest
    I - SG Include Join alert rcvd, P - (*,G) Programmed in hardware
    H - Joining SPT due to policy, D - Joining SPT due to protocol
    Z - Entry marked for deletion, C - Learned from a DR via a register
    A - Learned via Anycast RP Router, M - Learned via MSDP
    N - May notify MSDP, K - Keepalive timer not running
    T - Switching Incoming Interface, B - Learned via Border Router
RPF route: U - From unicast routing table
           M - From multicast routing table
ff33::1:0:0:1
  101:1::2, 2:03:00, flags: S
    Incoming interface: Ethernet11/1
    RPF route: [U] 101:1::/64 [110/1] via fe80::464c:a8ff:feb7:39e9
    Outgoing interface list:
      Ethernet6/1.1
      Ethernet4/1.1
      Ethernet7/1.1
      Ethernet9/1.1
      Ethernet8/1.1
      Ethernet2/1.1
      Ethernet5/1.1
      Ethernet3/1.1
```

- The **show multicast fib ipv6** command displays the Multicast Forwarding Information Base (MFIB) table.

```
switch# show mfib ipv6
Activity poll time: 60 seconds
  ff33::1:0:0:1 101:1::2
    Ethernet11/1 (iif)
    Ethernet9/1.1
    Ethernet2/1.1
    Ethernet3/1.1
    Ethernet6/1.1
    Ethernet5/1.1
    Ethernet8/1.1
    Ethernet7/1.1
    Ethernet4/1.1
      Activity 0:00:35 ago
```

- The **show platform fap mroute ipv6** command displays the Platform Hardware Forwarding table.

```
switch# show platform fap mroute ipv6
Jericho0 Multicast Routes:
--------------------------
Location    GroupId    Group                            Source

IIF         McId       OIF
FLP/TT      FLP/TT     TT                               FLP

FLP         FLP        FLP
-------------------------------------------------------------------
4096/2048   1/1       ff33::1:0:0:23/128                101:1::2/128

Vlan1357 21504     Vlan1044(Et7/1) Vlan1123(Et9/1)


Vlan1200(Et8/1) Vlan1223(Et2/1)


Vlan1226(Et5/1) Vlan1232(Et3/1)


Vlan1307(Et6/1) Vlan1337(Et4/1)
```

16.1.8.4 Multicast Boundary Configuration

The multicast boundary specifies subnets where source traffic entering an interface is filtered to prevent the creation of mroute states on the interface. The interface is not included in the Outgoing Interface List (OIL). Multicast PIM, IGMP and other multicast data cannot cross the boundary, facilitating the use of a multicast group address in multiple administrative domains.

In addition, an interface with a boundary ACL will filter any joins (RX or TX) for groups that are not allowed by the ACL. This also applies to multicast boundary ACLs with the **s** option.

The **ip multicast boundary** command configures the multicast boundary. The multicast boundary can be specified through multiple IPv4 subnets or one standard IPv4 ACL.

In an ACL method, the multicast subnets are allowed only from the permit entries of the ACL and rest is either denied or filtered. Whereas, in a non-ACL method the statements configure subnets that are only denied or filtered.

Examples

- These commands configure the multicast address of ***229.43.23.0/24*** as a multicast boundary where source traffic is restricted from ***interface vlan 300***.

```
switch(config)# interface vlan 300
switch(config-if-vl300)# ip multicast boundary 229.43.23.0/24
switch(config-if-vl300)#
```

- These commands create a standard ACL, then implement the ACL in an **ip multicast boundary** command to allow multicast for subnet (***224.0.0.0/4***) and create a multicast boundary for all remaining subnets by denying them.

```
switch(config)# ip access-list standard mbac1
switch(config-std-acl-mbac1)# 10 deny 225.123.0.0/16
switch(config-std-acl-mbac1)# 20 deny 239.120.10.0/24
switch(config-std-acl-mbac1)# 30 permit 224.0.0.0/4
```

```
switch(config-std-acl-mbac1)# exit
switch(config)# interface vlan 200
switch(config-if-Vl200)# ip multicast boundary mbac1
switch(config-if-Vl200)# exit
switch(config)#
```

16.1.8.5 Multicast multipath router-ID (IPv4)

The **`multipath deterministic`** command with the **router-ID** allows downstream PIM neighbors to direct traffic for a group or source-group to be routed through only the specified upstream PIM neighbor for all downstream PIM neighbors with the same ECMP routes. Traffic is load balanced by group or source-group across upstream PIM neighbors as in the default multipath mode.

Multipath router-ID uses the Interface Identifier (from the Hello Option for PIM) to give PIM-enabled devices a globally unique router ID for each of their PIM neighbors.

The following configures the switch to use the router_ID.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# multipath deterministic router-id
```

The following displays the switch before the configuration.

```
sSwitch(config-router-multicast-ipv4)# show active all
router multicast
   ipv4
      counters bytes
      activity polling-interval 60
      no routing
      multipath deterministic
      max-fastdrops 1024
      unresolved cache-entries max 4000
      unresolved packet-buffers max 3
      software-forwarding kernel
```

The following displays the switch after the configuration.

```
Switch(config-router-multicast-ipv4)# show active all
router multicast
   ipv4
      counters bytes
      activity polling-interval 60
      no routing
      multipath deterministic router-id
      max-fastdrops 1024
      unresolved cache-entries max 4000
      unresolved packet-buffers max 3
      software-forwarding kernel
```

16.1.8.6 Configuring MFIB

MFIB formats PIM and IGMP multicast routes for protocol-independent hardware packet forwarding and adds them to the hardware Multicast Expansion Table (MET) and the hardware FIB.

MFIB Polling Interval

The switch records activity levels for multicast routes in the MFIB after polling the corresponding hardware activity bits. The **`activity polling-interval`** command specifies the frequency at which the switch polls the hardware activity bits for the multicast routes.

Example

These commands set the MFIB activity polling period to *15* seconds.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# activity polling-interval 15
switch(config-router-multicast-ipv4)#
```

MFIB Fast Drops

In IP multicast protocols, every (S,G) or (*,G) route is associated with an inbound Reverse Path Forwarding (RPF) interface. Packets arriving on an interface not associated with the route may need CPU-dependent PIM processing, so packets received by non-RPF interfaces are sent to the CPU by default, causing heavy CPU processing loads.

Multicast routing protocols often do not require non-RPF packets; these packets do not require software processing. The CPU therefore updates the hardware MFIB with a fast-drop entry when it receives a non-RPF interface packet that PIM does not require. Additional packets that match the fast-drop entry are not sent to the system software.

Fast drop is enabled on all interfaces by default. The **no ip mfib fastdrop** command disables MFIB fast drop for the configuration mode interface.

Example

This command disables MFIB fast drop for the ***interface vlan 120***.

```
switch(config)# interface vlan 120
switch(config-if-Vl120)# no ip mfib fastdrop
switch(config-if-Vl120)#
```

The **ip mfib max-fastdrops** command limits the number of fast-drop routes that the switchs MFIB table can contain. The default fast-drop route limit is ***1024***.

Example

This command sets the maximum number of fast-drop routes to ***2000***.

```
switch(config)# ip mfib max-fastdrops 2000
switch(config)#
```

The **clear ip mfib fastdrop** command, in global configuration mode, removes all MFIB fast-drop entries on all interfaces.

Example

This command removes all fast-drop entries from the MFIB table.

```
switch# clear ip mfib fastdrop
switch#
```

The **show multicast fib ipv4** command displays information about the routes and interfaces in the IPv4 MFIB.

- **show multicast fib ipv4** displays MFIB information for hardware-forwarded routes.
- **show multicast fib ipv4 software** displays MFIB information for software-forwarded routes.

Example

This command displays MFIB information for hardware-forwarded routes.

```
switch# show multicast fib ipv4
```

```
Activity poll time: 60 seconds
239.255.255.250 172.17.26.25
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
239.255.255.250 172.17.26.156
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
239.255.255.250 172.17.26.178
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:03:37 ago switch#
```

MFIB Unresolved Cache-entries Max

The **unresolved cache-entries max** command configures the maximum number of unresolved (S,G) routes that the switch can cache packets. All packets belonging to (S,G) routes exceeding the limit are dropped. The default buffer size is ***4000*** routes. See **ip multicast boundary** to limit the number of cached packets per S,G.

Example

This command sets the maximum MFIB unresolved cache-entry buffer size to ***6000*** routes in the default VRF.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# unresolved cache-entries max 6000
switch(config-router-multicast-ipv4)#
```

MFIB Unresolved Packet-buffers Max

The **ip multicast boundary** command specifies the number of packets per unresolved route that are queued while the route is being resolved by the switch. The limit for **ip multicast boundary** is for an individual route, packets that exceed this limit are dropped. By default, the switch processes three unresolved packets for an individual route. See unresolved cache-entries max to limit the number of unresolved routes that are cached.

Example

This command configures the switch in the default VRF to cache up to ***30*** multicast packets from any route before that route is resolved.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# unresolved packet-buffers max 30
switch(config-router-multicast-ipv4)#
```

16.1.8.7 Configuring Static Multicast

Use the **route** command to configure the Static Multicast in the Router Multicast mode.

Example

```
switch(config)# router multicast
switch(config-router-multicast)# route
```

Configuring Static Multicast routes for a VRF

The Static Multicast route can be configured on a VRF in the Router Multicast VRF IPv4 Configuration mode. Note, static multicast commands belong in the router multicast mode. To maintain backward compatibility for default VRF, static multicast commands for default VRF can be entered without changing to router multicast mode.

Example

```
switch(config)# router multicast
switch(config-router-multicast)# route
```

Source address can be optionally entered. When no source address is entered, it is assumed to be ***0.0.0.0***.

```
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1 iif
```

Note, the IIF must belong to the same VRF for which the command is being executed for the configuration to take effect.

```
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1 iif ethernet
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1 iif ethernet 30
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1 iif ethernet 30 oif
```

Multiple interfaces can be entered following the OIF. However, OIF is also optional. When there are no OIFs, then traffic for the S,G on the incoming interface is dropped.

```
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1 iif ethernet 30 oif
 ethernet
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1 iif ethernet 30 oif
 ethernet 30 ethernet 32
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1 iif ethernet 30 oif
 ethernet 30 ethernet 32 cpu priority
```

The CPU option can be used to have CPU in the OIF set, it is optional. The **priority** is an optional and the value ranges from ***1*** to ***255***. By default, the value is ***255***.

```
switch(config-router-multicast)# route 1.1.1.1 10.1.1.1 iif ethernet 30 oif
 ethernet 30 ethernet 32 cpu priority 40
```

Displaying Static Multicast Information

Use the **show multicast fib ipv4** command verify the status of the Static Multicast configured interfaces.

```
switch# show multicast fib ipv4
Activity poll time: 60 seconds
  225.1.1.1 10.1.1.1
    Ethernet1 (iif)
    Ethernet2
    Cpu
```

16.1.8.8 Displaying and Clearing the Mroute Table

The mroute table stores the states of inbound and outbound interfaces for each source/group pair (S,G). The switch discards and forwards packets on the basis of this state information. Each table entry, referred to as an mroute, corresponds to a unique (S,G) and contains:

- the multicast group address

- the multicast source address (or * for all sources)
- the inbound interface
- a list of outbound interfaces

Clearing mroute Entries

The **clear ip mroute** command removes route entries from the mroute table:

- **clear ip mroute *** all entries from the mroute table.
- **clear ip mroute *gp_ipv4*** all entries for the specified multicast group.
- **clear ip mroute *gp_ipv4 src_ipv4*** all entries for the specified source sending to a specified group.

Examples

- This command removes all route entries from the mroute table.

```
switch# clear ip mroute *
switch#
```

- This command removes entries for source ***228.3.10.1*** sending to multicast group ***224.2.205.42***.

```
switch# clear ip mroute 224.2.205.42 228.3.10.1
switch#
```

Displaying the mroute Table

The **show ip mroute count** command displays IP multicast routing table statistics.

Example

This command displays IP multicast routing table statistics.

```
switch# show ip mroute count
IP Multicast Statistics
1 groups and 1 sources
Multicast routes: 1 (*,G), 1 (S,G)
Average of 1.00 sources per group
Maximum of 1 sources per group:
228.24.12.1 switch>
```

The **show ip mroute** command displays information from the IP multicast routing table.

- **show ip mroute** displays information for all routes in the table.
- **show ip mroute *gp_addr*** displays information for the specified multicast group.

Example

This command displays the IP multicast routing table for the multicast group ***225.1.1.1***.

```
switch# show ip mroute 225.1.1.1
PIM Sparse Mode Multicast Routing Table
Flags: E - Entry forwarding on the RPT, J - Joining to the SPT
R - RPT bit is set, S - SPT bit is set
W - Wildcard entry, X - External component interest
I - SG Include Join alert rcvd, P - Ex-Prune alert rcvd
H - Joining SPT due to policy, D - Joining SPT due to protocol
Z - Entry marked for deletion
A - Learned via Anycast RP Router
225.1.1.1
172.28.1.100, 5d04h, flags: S
Incoming interface: Vlan281
Outgoing interface list:
```

```
Port-Channel999 switch>
```

16.1.9 Multicast Commands

Multicast Configuration Commands (Global)

- activity polling-interval
- hardware counter feature multicast ipv4
- ip mfib max-fastdrops
- ip mroute
- ip multicast static
- mld last-listener-query-count
- mld last-listener-query-interval
- mld query-interval
- mld query-response-interval
- mld robustness
- mld snooping
- mld startup-query-count
- mld startup-query-interval
- mld static-group
- mld
- multipath deterministic
- multipath none
- route
- router multicast
- routing
- rpf route
- unresolved cache-entries max
- unresolved packet-buffers max

Multicast Configuration Commands (Interface)

- ip mfib fastdrop
- ip multicast boundary

Multicast Clear Commands

- clear ip mfib fastdrop
- clear ip multicast counters
- clear ip mroute

Multicast Display Commands

- show ip mroute
- show ip mfib counters
- show ip mroute count
- show ip multicast boundary
- show mld membership
- show mld querier
- show mld snooping
- show mld statistics
- show mld summary

- show multicast fib ipv4
- show multicast fib ipv4 software
- show multicast fib ipv6
- show pim ipv6 sparse-mode route
- show platform fap mroute ipv6

16.1.9.1 activity polling-interval

The switch records activity levels for multicast routes in the mfib after polling the corresponding hardware activity bits. The **`activity polling-interval`** command specifies the frequency at which the switch polls the hardware activity bits for the multicast routes.

The **`no activity polling-interval`** and **`default activity polling-interval`** commands restore the default interval of ***60*** seconds by removing the **`activity polling-interval`** command from ***running-config***.

Command Mode

Router Multicast IPv4 Configuration

Command Syntax

`activity polling-interval` ***period***

`no activity polling-interval`

`default activity polling-interval`

Parameter

period interval (seconds) between polls. Values range from ***1*** to ***60***. Default is ***60***.

Example

These commands set the MFIB activity polling period to ***15*** seconds.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# activity polling-interval 15
switch(config-router-multicast-ipv4)#
```

16.1.9.2 clear ip mfib fastdrop

The **`clear ip mfib fastdrop`** command removes all fast-drop entries from the MFIB table.

Command Mode

Privileged EXEC

Command Syntax

`clear ip mfib fastdrop`

Example

This command removes all fast-drop entries from the MFIB table.

```
switch# clear ip mfib fastdrop
switch#
```

16.1.9.3 clear ip mroute

The **clear ip mroute** command removes route entries from the mroute table, as follows:

- **clear ip mroute *** removes all entries from the mroute table.
- **clear ip mroute *gp_ipv4*** removes all entries for the specified multicast group.
- **clear ip mroute *gp_ipv4src_ipv4*** removes all entries for the specified source sending to the specified group.

Command Mode

Privileged EXEC

Command Syntax

`clear ip mroute` **ENTRY_LIST**

Parameters

ENTRY_LIST entries that the command removes from the mroute table. Options include:

- * all route entries.
- ***gp_ipv4*** all entries for multicast group ***gp_ipv4*** (dotted decimal notation)
- ***gp_ipv4 src_ipv4*** all entries for source (***src_ipv4***) sending to group (***gp_ipv4***)

Examples

- This command removes all route entries from the mroute table.

```
switch# clear ip mroute *
switch#
```

- This command removes entries for the source ***228.3.10.1*** sending to multicast group ***224.2.205.42***.

```
switch# clear ip mroute 224.2.205.42 228.3.10.1
switch#
```

16.1.9.4 clear ip multicast counters

To clear per multicast route ingress counters or clear counters for a particular multicast route, use the **`clear ip multicast counters`** command.

Command Mode

EXEC

Command Syntax

`clear ip multicast` [vrf [***vrf-name***]] counters ***group_addr source_addr***

Parameters

- **vrf *vrf-name*** VRF name.
- **counters** Counter for bytes/packets.
- ***group_addr*** Group address.
- ***source_addr*** Source address.

16.1.9.5 hardware counter feature multicast ipv4

Use the **`hardware counter feature multicast ipv4`** command to enable per multicast router ingress packets and byte counters. The **no** and **default** forms of the command disables the feature. The feature is disabled by default.

Command Mode

Configuration mode

Command Syntax

hardware counter feature multicast ipv4

no hardware counter feature multicast ipv4

default hardware counter feature multicast ipv4

Example

On DCS-7020R, DCS-7280R/R2 and 7500R/R2 platforms, create and apply user defined TCAM profile with multicast counter feature db enabled. This example creates a profile copying it from the default, then modifying the profile.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile <profileName> copy default
switch(config-hw-tcam-profileName)# feature counter multicast ipv4
switch(config-hw-tcam-profileName-feature-counter-multicast-ipv4))# exit
switch(config-hw-tcam-profileName)# no feature mirror ip
switch(config-hw-tcam-profileName)# exit
switch(config-hw-tcam)# system profile <profileName>
```

16.1.9.6 ip mfib fastdrop

In IP multicast protocols, every (S,G) or (*,G) route is associated with an inbound Reverse Path Forwarding (RPF) interface. Packets arriving on an interface not associated with the route may need CPU-dependent PIM processing, so packets received by non-RPF interfaces are sent to the CPU by default, causing heavy CPU processing loads.

Multicast routing protocols often do not require non-RPF packets; these packets do not require software processing. The CPU therefore updates the hardware MFIB with a fast-drop entry when it receives a non-RPF interface packet that PIM does not require. Additional packets that match the fast-drop entry are not sent to the system software.

Fast drop is enabled on all interfaces by default. The **no ip mfib fastdrop** command disables MFIB fast drop for the configuration mode interface.

The **ip mfib fastdrop** and **default ip mfib fastdrop** commands enable MFIB fast drop for the configuration mode interface by removing the corresponding **no ip mfib fastdrop** command from ***running-config***.

The **clear ip mfib fastdrop** command, in the ***global*** configuration mode, removes all MFIB fast-drop entries on all interfaces.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ip mfib fastdrop

no ip mfib fastdrop

default ip mfib fastdrop

Example

This command disbles MFIB fast drop for ***interface vlan 120***.

```
switch(config)# interface vlan 120
switch(config-if-Vl120)# no ip mfib fastdrop
switch(config-if-Vl120)#
```

16.1.9.7 ip mfib max-fastdrops

The **ip mfib max-fastdrops** command limits the number of fast-drop routes that the switchs MFIB table can contain.

The **no ip mfib max-fastdrops** and **default ip mfib max-fastdrops** commands restore the default fast-drop route limit of ***1024*** by removing the **ip mfib max-fastdrops** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip mfib max-fastdrops ***quantity***

no ip mfib max-fastdrops

default ip mfib max-fastdrops

Parameters

quantity maximum number of fast-drop routes. Value ranges from ***0*** to ***1000000*** (one million). Default is ***1024***.

Example

This command sets the maximum number of fast-drop routes to ***2000***.

```
switch(config)# ip mfib max-fastdrops 2000
switch(config)#
```

16.1.9.8 ip mroute

The **ip mroute** command configures the Static Mroute on the switch.

The **no ip mroute** and **default ip mroute** commands remove the specified static mroute from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip mroute [***source-prefix*** | ***source-address mask***] [***rpf-interface*** | ***rpf-neighbor***] [admin ***distance***]

Example

This command configures a Static IP mroute for a source ***1.1.1.1/32*** with an administrative distance ***20*** on ***interface ethernet 2/1***.

```
switch(config)# ip mroute 1.1.1.1/32 ethernet 2/1 20
```

16.1.9.9 ip multicast boundary

The **ip multicast boundary** command specifies subnets where source traffic entering the configuration mode interface is dropped, preventing the creation of mroute states on the interface. The interface is not included in the Outgoing Interface List (OIL). The multicast boundary can be specified through multiple IPv4 subnets or one standard IPv4 ACL.

In an ACL method, the multicast subnets are allowed only from the permit entries of the ACL and rest is either denied or filtered. Whereas, in a non-ACL method the statements configure subnets that are only denied or filtered.

Multicast PIM, IGMP and other multicast data cannot cross the boundary, facilitating the use of a multicast group address in multiple administrative domains.

The **no ip multicast boundary** and **default ip multicast boundary** commands delete the specified subnet restriction by removing the corresponding **ip multicast boundary** command from ***running-config***. When these commands do not specify a subnet address, all **ip multicast boundary** statements for the configuration mode interface are removed.

Command Mode

Interface-Ethernet Configuration

Interface-Port-channel Configuration

Interface-VLAN Configuration

Command Syntax

ip multicast boundary **SUBNET** [**TCAM**]

no ip multicast boundary [**SUBNET**]

default ip multicast boundary [**SUBNET**]

Parameters

- **SUBNET** the subnet address configured as the multicast boundary. Options include:
  - ***net_addr*** multicast subnet address (CIDR or address mask).
  - ***acl_name*** standard Access Control List (ACL) that specifies the multicast group addresses.
- **TCAM** specifies address inclusion in the routing table. Options include:
  - ***no parameter*** boundaries ((S,G) entries) are added to routing table.
  - **out** boundaries are not added to routing table.

Guidelines

When **out** is selected, the first inbound data packet corresponding to the **SUBNET** may be sent to the CPU. In response, the packet is dropped and the boundary prefix is added to the hardware table. In this scenario, the mroute entry is added only when data traffic is received.

Restrictions

Only one command that specifies an ACL can be assigned to an interface. Commands that specify an ACL and a subnet cannot be simultaneously assigned to an interface.

Examples

- This command configures the multicast address of ***229.43.23.0/24*** as a multicast boundary where source traffic is restricted from ***interface vlan 300***.

```
switch(config)# interface vlan 300
switch(config-if-vl300)# ip multicast boundary 229.43.23.0/24
switch(config-if-vl300)#
```

- These commands create a standard ACL, then implement the ACL in an ip multicast boundary command to allow multicast for subnet (***224.0.0.0/4***) and create a multicast boundary for all remaining subnets by denying them.

```
switch(config)# ip access-list standard mbac1
switch(config-std-acl-mbac1)# 10 deny 225.123.0.0/16
switch(config-std-acl-mbac1)# 20 deny 239.120.10.0/24
switch(config-std-acl-mbac1)# 30 permit 224.0.0.0/4
switch(config-std-acl-mbac1)# exit
switch(config)# interface vlan 200
switch(config-if-Vl200)# ip multicast boundary mbac1
```

```
switch(config-if-Vl200)# exit
switch(config)#
```

16.1.9.10 ip multicast static

The **ip multicast static** command enables static multicast routing on the switch.

The **exit** command returns the switch to global configuration mode.

Command Mode

Interface Ethernet Configuration

Command Syntax

ip multicast static

Example

The following commands configure the static multicast routing on the switch.

```
switch(config)# interface ethernet 1/2
switch(config-if-Et1/2)# no switchport
switch(config-if-Et1/2)# ip address 1.1.1.1/24
switch(config-if-Et1/2)# ip multicast static
```

16.1.9.11 mld last-listener-query-count

The **mld last-listener-query-count** command specifies the number of query messages the switch sends in response to a group-specific or group-source-specific leave message.

After receiving a message from a host leaving a group, the switch sends query messages at intervals specified by **mld last-listener-query-interval**. If the switch does not receive a response to the queries after sending the number of messages specified by this parameter, it stops forwarding messages to the host.

The **no mld last-listener-query-count** and **default mld last-listener-query-count** commands reset the last-listener-query-count to the default value by removing the corresponding **mld last-listener-query-count** command from the ***running-config***. Default value is ***2***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mld last-listener-query-count *number*

no mld last-listener-query-count

default mld last-listener-query-count

Parameter

number the last listener query count. Values range from ***0*** to ***100***. Default value is ***2***.

Example

This command configures the last listener query count to ***3*** on an ***interface Ethernet 1***.

```
switch(config)# interface Ethernet 1
switch(config-if-Et1)# mld last-listener-query-count 3
```

16.1.9.12 mld last-listener-query-interval

The **mld last-listener-query-interval** command configures the switchs transmission interval for sending group-specific or group-source-specific query messages from the configuration mode interface.

When a switch receives a message from a host that is leaving a group, it sends query messages at intervals set by this command. The **mld last-listener-query-count** specifies the number of messages that are sent before the switch stops forwarding packets to the host.

If the switch does not receive a response after this period, it stops forwarding traffic to the host on behalf of the group, source, or channel.

The **no mld last-listener-query-interval** and **default mld last-listener-query-interval** commands reset the last-listener-query-interval to the default value by removing the corresponding **mld last-listener-query-interval** command from the ***running-config***. Default value is ***1*** second.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mld last-listener-query-interval ***period***

no mld last-listener-query-interval

default mld last-listener-query-interval

Parameter

period the last listener query interval in seconds. Values range from ***1*** to ***3175***.

Example

This command configures the last listener query interval to ***2*** seconds on an ***interface Ethernet1***.

```
switch(config) #interface Ethernet1
switch(config-if-Et1)# mld last-listener-query-interval 2
```

16.1.9.13 mld query-interval

The **mld query-interval** command configures the frequency at which the configuration mode interface, as an MLD querier, sends host-query messages.

An MLD querier sends host-query messages to discover the multicast groups that have members on networks attached to the interface. The switch implements a default query interval of ***125*** seconds.

The **no mld query-interval** and **default mld query-interval** commands reset the query interval to the default value by removing the corresponding **mld query-interval** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mld query-interval ***period***

no mld query-interval

default mld query-interval

Parameters

period the interval between query messages in seconds. Values range from ***1*** to ***3175***.

Example

This command configures the query interval of ***30*** seconds on an ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld query-interval 30
```

16.1.9.14 mld query-response-interval

The **mld query-response-interval** command configures the maximum response time that the recipient can wait before responding with a membership report on receipt of a general query.

The **no mld query-response-interval** and **default mld query-response-interval** commands reset the query interval to the default value by removing the corresponding **mld query-response-interval** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mld query-interval ***period***

no mld query-interval

default mld query-interval

Parameter

period the query response interval in seconds. Values range from ***1*** to ***3175***.

Example

This command configures the query response interval of ***30*** seconds on an ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld query-response-interval 30
```

16.1.9.15 mld robustness

The **mld robustness** command configures the number of general queries to be sent before the router assumes there are no more listeners.

The **no mld robustness** and **default mld robustness** commands reset the robustness to the default value by removing the corresponding **mld robustness** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mld robustness ***robust_value***

no mld robustness

default mld robustness

Parameter

robust_value the robustness count. Values range from ***1*** to ***100***.

Example

This command configures the robustness value to ***2*** on an ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld robustness 2
```

16.1.9.16 mld snooping

Multicast Listener Discovery (MLD) snooping constrains the flooding of IPv6 multicast traffic on VLANs and IGMP snooping for IPv4 environments. When MLD snooping is enabled on a VLAN, the device examines MLD messages between the hosts and multicast routers and learns which hosts are interested in receiving traffic for a multicast group. You can use the **`mld snooping`** command to configure MLDv2 snooping globally and per VLAN.

The **`no`** and the **`default`** forms of the command removes the mld snooping configuration.

Command Mode

Global configuration mode

Command Syntax

`mld snooping`

`no mld snooping`

`default mld snooping`

Example

The following example configures MLDv2 snooping globally and per VLAN.

```
switch(config)# mld snooping
switch(config-mld-snooping)# disabled
switch(config-mld-snooping)# vlan 1-100
switch(config-mld-snooping)# vlan 101
switch(config-mld-snooping-vlan-101)# disabled
```

16.1.9.17 mld startup-query-count

The **`mld startup-query-count`** command specifies the number of query messages that an interface sends during the startup query interval.

When an interface starts running MLD, it can establish the group state more quickly by sending query messages at a higher frequency. The **`mld startup-query-interval`** and **`mld startup-query-count`** commands define the startup period and the query message transmission frequency during that period.

The **`no mld startup-query-count`** and **`default mld startup-query-count`** commands restore the default startup-query-count value by removing the corresponding mld startup-query-count command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`mld startup-query-count` ***number***

`no mld startup-query-count`

`default mld startup-query-count`

Parameters

number the startup query count. Values range from ***1*** to ***100***.

Example

This command configures the startup query count of ***4*** on an ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld startup-query-count 4
```

16.1.9.18 mld startup-query-interval

The **mld startup-query-interval** command specifies the interval between the general queries sent by a querier on startup.

When an interface starts running MLD, it can establish the group state quicker by sending query messages at a higher frequency. The **mld startup-query-interval** and **mld startup-query-count** commands define the startup period and the query message transmission frequency during that period.

The **no mld startup-query-count** and **default mld startup-query-interval** commands restore the default startup-query-interval value by removing the corresponding **mld startup-query-interval** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mld startup-query-interval ***period***

no mld startup-query-interval

default mld startup-query-interval

Parameters

period the startup query interval in seconds. Values range from ***1*** to ***3175***.

Example

This command configures the startup query interval of ***100*** seconds on an ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld startup-query-interval 100
```

16.1.9.19 mld static-group

The **mld static-group** command configures the configuration mode interface as a static member of a specified multicast group. This allows the router to forward multicast group packets through the interface without otherwise appearing or acting as a group member. By default, static group memberships are not configured on any interfaces.

If the command includes a source address, only multicast group messages received from the specified host address are fast-switched. Otherwise, all multicast traffic of the specified group is fast-switched.

The **no mld static-group** and **default mld static-group** commands remove the configuration mode interfaces group membership by removing the corresponding **mld startup-group** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mld static-group ***source_address*** [***group_address*** | access-list ***acl_name***]

no mld static-group ***source_address*** [***group_address*** | access-list ***acl_name***]

default mld static-group ***source_address*** [***group_address*** | access-list ***acl_name***]

Parameters

- ***source_address*** IP address of the host that originates multicast data packets.
- ***group_address*** IPv6 address of a multicast group.
- ***access-list*** IPv6 access list to use as a static group list.
- ***acl_name*** specifies access-list name

Examples

- This command configures static groups on an ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld static-group ff30::1 a::1
```

- This command configures multiple static groups using an access list on an ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld static-group access-list testAccessList
```

16.1.9.20 mld

The **mld** command enables multicast listener discovery on an interface which controls the flow of layer 3 IPv6 multicast traffic. Hosts request and maintain multicast group membership through MLD messages. Multicast routers use MLD to maintain a membership list of active multicast groups for each attached network.

The **no mld** and **default mld** commands restore the default behavior by removing the corresponding **mld** command from the ***running-config***.

Note: It is possible to change the values of the querier parameters used by MLD.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mld [***last-listener-query-count*** | ***last-listener-query-interval*** | ***query-interval*** | ***query-response-interval*** | ***robustness*** | ***startup-query-count*** | ***startup-query-interval*** | ***static-group***]

no mld

default mld

Parameters

- ***last-listener-query-count*** the number of group-specific or group-source-specific queries to send before the router assumes there are no more listeners.
- ***last-listener-query-interval*** the interval between the last listener queries.
- ***query-interval*** the interval between the general queries regularly sent by a querier.
- ***query-response-interval*** the interval that the host has to respond to a general query.
- ***robustness*** the number of general queries to send before the router assumes there are no more listeners.
- ***startup-query-count*** the number of queries a router sends at startup.
- ***startup-query-interval*** the interval between the general queries sent by a querier at startup.
- ***static-group*** the number of static groups or sources of MLD messages.

Example

This command enables MLD on the ***interface Ethernet1***.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# mld
```

16.1.9.21 multipath deterministic

By default, multicast traffic is load balanced by distributing packets over all ECMP links. The **no multipath deterministic** command routes multicast ECMP traffic to the neighbor with the highest IPv4 address.

The **multipath deterministic** and **default multipath deterministic** commands restore the default behavior of randomly distributing multicast traffic over all ECMP links.

Command Mode

Router Multicast IPv4 Configuration

Command Syntax

multipath deterministic

no multipath deterministic

default multipath deterministic

Related Commands

The **multipath none** command performs the same function as **no multipath deterministic**.

Examples

- These commands configure the switch to route multicast traffic through the ECMP link to the neighbor with the highest IP address.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# no multipath deterministic
switch(config-router-multicast-ipv4)#
```

- These commands configure the switch to load balance multicast traffic by distributing packets over all ECMP links.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# multipath deterministic
switch(config-router-multicast-ipv4)#
```

- These commands configure the switch to load balance multicast traffic by enabling the behavior of RPF selection.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# multipath deterministic router-id
switch(config-router-multicast-ipv4)#
```

16.1.9.22 multipath none

By default, multicast traffic is load balanced by distributing packets over all ECMP links. The **multipath none** command routes multicast ECMP traffic to the neighbor with the highest IPv4 address.

The **no multipath none** and **default multipath none** commands restore the default behavior of randomly distributing multicast traffic over all ECMP links by removing the **multipath none** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

multipath none

no multipath none

default multipath none

Related Commands

The **multipath deterministic** commsnd performs the same function as **no multipath none**

Examples

- These commands configure the switch to route multicast traffic through the ECMP link to the neighbor with the highest IP address.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# multipath none
switch(config-router-multicast-ipv4)#
```

- These commands configure the switch to load balance multicast traffic by distributing packets over all ECMP links.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# no multipath none
switch(config-router-multicast-ipv4)#
```

16.1.9.23 route

The **route** command configures a static multicast route for the specified source, destination group, and incoming interface on the router.

The **no route** and **default route** commands remove the specified static multicast route by removing the corresponding **route** command from ***running-config***.

Command Mode

Router Multicast IPv4 Configuration

Router Multicast VRF IPv4 Configuration

Command Syntax

route ***group_address*** [***source_address***] iif ***interface*** [oif ***interface***] [cpu] [iifFrr ***interface***] [priority ***priority_num***]

no route ***group_address***

default route ***group_address***

Parameters

- ***group_address*** the multicast group address.
- ***source_address*** the optional source address for the multicast route.
- **iif *interface*** specifies an incoming interface for the static route.
- **cpu** optionally mirrors multicast packets to the CPU.
- **oif *interface*** specifies an optional outgoing interface to be included among those on which the multicast traffic is forwarded.
- **iifFrr *interface*** specifies an optional interface for multicast-only fast reroute.

- ***interface*** options include:
  - **Ethernet *ethernet_port*** Ethernet interface.
  - **Null0** drops all traffic.
  - **Port-Channel *lag_no*** port-channel interface or sub-interface; values range from ***1***-***2000*** or ***1***-***2000***.***1***-***4094***.
  - **Register0** drops all incoming traffic.
  - **Vlan *vlan_no*** VLAN interface.
- **priority *priority_num*** specifies an optional priority for the multicast route. If the same route is present in several multicast routing tables, the priority number is used to select the best available route. Values range from ***0*** to ***255***; PIM routes by default have a priority of ***0***, while static multicast routes by default have a priority of ***255***.

Example

These commands create a static multicast route in the default VRF. The static route has a group address of ***225.3.3.3*** and source address of ***1.1.1.1***. It uses ***Vlan100*** as its incoming interface, VLANs ***200*** and ***300*** as its outgoing interfaces, and ***interface Ethernet2*** as its multicast-only fast reroute interface.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# route 225.3.3.3 1.1.1.1 iif Vlan100 oif
Vlan200 Vlan300 iifFrr Ethernet2
switch(config-router-multicast-ipv4)#
```

16.1.9.24 router multicast

The **`router multicast`** command places the switch in router-multicast configuration mode to configure IPv4 and IPv6 router multicast traffic.

Command Mode

Global Configuration

Command Syntax

`router multicast`

Example

The following command places the switch in ***router-multicast*** configuration mode.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv6
switch(config-router-multicast-ipv6)#routing
```

16.1.9.25 routing

The **`routing`** command allows the switch to forward multicast packets. Multicast routing is disabled by default.

The **`no routing`** and **`default routing`** commands disable multicast routing by removing the **`routing`** command from ***running-config***.

Command Mode

Router Multicast IPv4 Configuration

Router Multicast VRF IPv4 Configuration

Command Syntax

`routing`

no routing

default routing

Example

These commands enable multicast routing on the switch.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# routing
switch(config-router-multicast-ipv4)#
```

16.1.9.26 rpf route

The **rpf route** command specifies a candidate for the multicast Reverse Path Forwarding (RPF) interface of any (S,G) multicast route (mroute), where the source falls within the given network prefix. Static mroutes are stored in a separate routing table, the Multicast Routing Information Base (MRIB).

Command Mode

Router Multicast IPv4 Configuration

Router Multicast VRF IPv4 Configuration

Command Syntax

rpf route {***source_prefix*** | ***source_address*** | ***mask***}{***rpf_interface*** | ***rpf_neighbor***>} [***admin_distance***]

no rpf route {***source_prefix*** | ***source_address*** | ***mask***}{***rpf_interface*** | ***rpf_neighbor***}

default rpf route {***source_prefix*** | ***source_address mask***}{***rpf_interface*** | ***rpf_neighbor***}

Parameters

- ***source_prefix*** specifies the source prefix.
- ***source_address*** specifies the source address.
- *mask* specifies the address mask.
- ***rpf_interface*** specifies the multicast RPF interface.
- ***rpf_neighbor*** specifies the multicast RPF neighbor.
- ***admin_distance*** specifies the administrative distance (optional). Values range from ***1*** to ***255***.

Examples

- These commands select the longest match when a source matches multiple static mroutes in the MRIB.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# rpf route 10.0.0.0/16 Ethernet 4
switch(config-router-multicast-ipv4)# rpf route 11.10.1.0/24 Ethernet 5
switch(config-router-multicast-ipv4)# rpf route 11.10.1.2/32 Ethernet 6
switch(config-router-multicast-ipv4)#
```

- These commands include an administrative distance of ***255*** on ***interface Ethernet 5*** with static mroute.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# rpf route 10.0.0.0/16 Ethernet 4
switch(config-router-multicast-ipv4)# rpf route 11.10.1.0/24 Ethernet 5 255
switch(config-router-multicast-ipv4)# rpf route 11.10.1.2/32 Ethernet 6
switch(config-router-multicast-ipv4)#
```

16.1.9.27 show ip mfib counters

Use the **show ip mfib counters** command to display per multicast route ingress packet and byte counters.

Command Mode

EXEC

Command Syntax

show ip mfib vrf [***vrf-name***] ***group_addr source_addr*** counters

Parameters

- **vrf** [***vrf-name***] VRF name.
- ***group_addr*** Group address.
- ***source_addr*** Source address.
- **counters** Counter for bytes/packets.

Example

```
switch# show ip mfib 225.1.2.1 10.46.1.2 counters
Activity poll time: 60 seconds
   225.1.2.1 10.46.1.2
      Byte: 1200200
      Packet: 12002
      Ethernet46 (iif)
      Ethernet47
      Activity 0:02:52 ago switch#
```

16.1.9.28 show ip mroute

The **show ip mroute** command displays information from the IP multicast routing table.

- **show ip mroute** displays information for all routes in the table.
- **show ip mroute** ***gp_addr*** displays information for the specified multicast group.

Command Mode

EXEC

Command Syntax

show ip mroute

show ip mroute ***gp_addr***

Parameters

gp_addr group IP address (dotted decimal notation).

Example

This command displays the IP multicast routing table entry for the multicast group ***225.1.1.11***.

```
switch# show ip mroute 225.1.1.1
PIM Sparse Mode Multicast Routing Table
Flags: E - Entry forwarding on the RPT, J - Joining to the SPT
R - RPT bit is set, S - SPT bit is set
W - Wildcard entry, X - External component interest
I - SG Include Join alert rcvd, P - Ex-Prune alert rcvd
H - Joining SPT due to policy, D - Joining SPT due to protocol
```

```
Z - Entry marked for deletion
A - Learned via Anycast RP Router
225.1.1.1
172.28.1.100, 5d04h, flags: S
Incoming interface: Vlan281
Outgoing interface list:
Port-Channel999 switch#
```

16.1.9.29 show ip mroute count

The **show ip mroute count** command displays IP multicast routing table statistics.

The **show ip mroute** command displays information from the IP multicast routing table.

Command Mode

EXEC

Command Syntax

show ip mroute count

Example

This command displays IP multicast routing table statistics.

```
switch# show ip mroute count
IP Multicast Statistics
1 groups and 1 sources
Multicast routes: 1 (*,G), 1 (S,G)
Average of 1.00 sources per group
Maximum of 1 sources per group:
228.24.12.1 switch#
```

16.1.9.30 show ip multicast boundary

The **show ip multicast boundary** command displays the summary of all IP multicast boundaries across all interfaces.

Command Mode

EXEC

Command Syntax

show ip multicast boundary [***group_prefix*** | ***group_prefix/length*** [out] | interface {ethernet ***e_num*** | loopback ***l_num*** | management ***m_num*** | port-channel ***p_num*** | vlan ***v_num***} | out]

Parameters

- ***no parameters*** displays the summary of all IP multicast boundaries across all interfaces.
- ***group_prefix*** displays the list of IP multicast boundaries matching the specified group address with subnet mask.
- ***group_prefix/length*** displays the list of IP multicast boundaries matching the specified group address with CIDR notation. Option includes:
  - **out** displays the specified group addresss IP multicast boundaries whose control plane filtering is enabled.
- **interface** displays IP multicast boundary of the specified interface. Options include:
  - **ethernet *e_num*** displays IP multicast boundaries of the specified Ethernet interface.
  - **loopback *l_num*** displays IP multicast boundaries of the specified Loopback interface.
  - **management *m_num*** displays IP multicast boundaries of the specified management interface.

  - **port-channel *p_num*** displays IP multicast boundaries of the specified port channel interface.
  - **vlan *v_num*** displays IP multicast boundaries of the specified VLAN interface.
- **out** displays all IP multicast boundaries whose only control plane filtering is enabled.

Examples

- This command displays the summary of all IP multicast boundaries across all interfaces.

```
switch(config-if-Et24)# show ip multicast boundary
Interface Denied Prefix Data Plane Filtered
Ethernet1 224.5.5.0/24 Yes
Ethernet1 224.6.6.0/24 Yes
Ethernet2 224.4.4.0/24 Yes
Ethernet3 224.5.5.0/24 No
```

- This command displays all IP multicast boundaries matching ***224.5.5.0 255.255.255.255***.

```
switch(config-if-Et24)# show ip multicast boundary 224.5.5.0 255.255.255.255
Interface Denied Prefix Data Plane Filtered
Ethernet1 224.5.5.0 255.255.255.255
Ethernet3 224.5.5.0 255.255.255.255 No
```

- This command displays all IP multicast boundaries matching ***224.5.5.0/24***.

```
switch(config-if-Et24)# show ip multicast boundary 224.5.5.0/24
Interface Denied Prefix Data Plane Filtered
Ethernet1 224.5.5.0/24
Ethernet3 224.5.5.0/24 No
```

- This command displays all IP multicast boundaries of ***interface Ethernet1***.

```
switch(config-if-Et24)# show ip multicast boundary interface Ethernet1
Interface Denied Prefix Data Plane Filtered
Ethernet1 224.5.5.0/24
Ethernet1 224.6.6.0/24 No
```

- This command displays the list of IP multicast boundaries whose only control plane filtering is enabled.

```
switch(config-if-Et24)# show ip multicast boundary out
Interface Denied Prefix Data Plane Filtered
Ethernet1 224.5.5.0/24 No
Ethernet3 224.5.5.0/24 No
```

16.1.9.31 show mld membership

The **show mld membership** command displays MLD group and source membership information on a specific interface.

Command Mode

EXEC

Command Syntax

show mld membership [dynamic | group | interface | static]

Parameters

- **dynamic** displays MLD information for a dynamic group.
- **group** displays MLD information for a specified multicast group address.
- **interface** displays MLD information for the specified interface.
- **static** displays MLD information for statically configured group.

Example

This command displays MLD group and source information on the Ethernet interfaces ***3*** and ***6***.

```
switch# show mld membership
Interface      Group         Source       Filter Mode
---------      -----         ------       -----------
Ethernet3      ff30::1       a::2         include
Ethernet3      ff30::1       a::1         include
Ethernet6      ff30::2       a::2         include
```

16.1.9.32 show mld querier

The **show mld querier** command displays information about the MLD querier and querier parameters.

Command Mode

EXEC

Command Syntax

show mld querier [interface | parameters]

Parameters

- **interface** displays MLD querier information.
- **parameters** displays MLD querier parameters.

Example

- This command displays MLD querier on the Ethernet interface ***Et3*** and ***Et6***.

```
switch# show mld querier
Interface    Querier                General    Other      Version
                                    Query      Querier
                                    Expiry     Expiry
---------    -----------------      ---------- ---------  -------
Et3          fe80::1:ff:fe01:0      0:01:14    N/A        2
Et6          fe80::1:ff:fe01:0      0:01:14    N/A        2
```

- This command displays MLD querier parameters on the Ethernet interface ***Et3*** and ***Et6***.

```
switch# show mld querier parameters
Interface     Robustness Query      Query      Startup    Startup    Last       Last
                         Interval   Response   Query      Query      Listener   Listener
                                    Interval   Interval   Count      Query      Query
                                                                     Interval   Count
---------     ---------- --------   ---------  ---------  -------    --------   --------
Et3           2          125        10         31.25      2          1          2
Et6           2          125        10         31.25      2          1          2
```

16.1.9.33 show mld snooping

Use the **show mld snooping** command to display the snooping status of the switch.

Command Mode

EXEC

Command Syntax

show mld snooping [counters [errors] | groups [***A:B:C:D:E:F:G:H*** | count | detail | local | mlag | user | vlan] | mrouter [detail | vlan] | querier [vlan] | vlan [*num*]]

Parameters

- **counters** MLD counter information.
  - **errors** Error counters.

- **groups** MLD group information.
  - ***A:B:C:D:E:F:G:H*** IPv6 address.
  - **count** Displays membership count.
  - **detail** Displays a comprehensive output.
  - **local** Displays groups learned locally via MLD.
  - **mlag** Displays groups learned via MLAG peer.
  - **user** Displays groups configured by the user.
  - **vlan** Specifies VLAN.
- **mrouter** MLD multicast router information.
  - **detail** Displays a comprehensive output.
  - **vlan** Specifies VLAN.
- **querier** MLD querier information.
  - **vlan** Specifies VLAN.
- **vlan** Specifies VLAN.
  - ***num 1-4094*** Identifier for Virtual LAN.

Examples

- Use the **`show mld snooping`** command to display the MLD snooping status of the switch.

```
switch# show mld snooping
   Global MLD Snooping configuration:
-------------------------------------------
MLD snooping                    : Enabled
Robustness variable             : 2

VLAN 1 :
----------
MLD snooping                    : Disabled
MLD max group limit             : No limit set
Recent attempt to exceed limit : No
MLD snooping pruning active     : False
Flooding traffic to VLAN        : True
VLAN 100 :
----------
MLD snooping                    : Enabled
MLD max group limit             : No limit set
Recent attempt to exceed limit : No
MLD snooping pruning active     : True
Flooding traffic to VLAN        : False


switch##show mld snooping vlan 100
   Global MLD Snooping configuration:
-------------------------------------------
MLD snooping                    : Enabled
Robustness variable             : 2

VLAN 100 :
----------
MLD snooping                    : Enabled
MLD max group limit             : No limit set
Recent attempt to exceed limit : No
MLD snooping pruning active     : True
Flooding traffic to VLAN        : False
```

- Use the **`show mld snooping groups`** command to display VLANs by group.

```
switch# show mld snooping groups
IGMP Snooping Group Membership
EX : Filter mode Exclude
IN : Filter mode Include
IR : Ingress Replication
VLAN  Group           Members
----  --------------  -------------------
100   ff05::2         Cpu, Et4
100   ff08::11        Et1
100   ff08::21        Et2
```

```
100   ff08::31         Et3
100   *                Et3, Et4
```

- Use the **show mld snooping groups detail** command to display detailed VLANs information by group.

```
switch(config-mld-snooping-vlan-100)# show mld snooping groups detail
IGMP Snooping Group Membership
EX : Filter mode Exclude
IN : Filter mode Include
IR : Ingress Replication
VLAN  Group            Source           Mode  Uptime       Members
----  ---------------  ---------------  ----  -----------  ------------
100   ff05::2          *                -     0d00h13m20   Cpu, Et4
100   ff08::11         *                -     0d00h13m17   Et1
100   ff08::21         1::1             IN    0d00h13m20   Et2
100   ff08::31         2::1             EX    0d00h01m31   Et3
100   *                *                -     -            Et3, Et4
```

The **show mld snooping groups** command output can be further filtered to output just the:

- **local** groups, That is groups learned locally using **show mld snooping groups local** command.
- **user** groups, that is groups configured by user using the **show mld snooping groups user** command.
- **mlag** groups, that is groups learned from the MLAG peer using the **show mld snooping groups mlag** command.
-

- Use the **show mld snooping mrouter** to display MLD multicat router information.

```
switch# show mld snooping mrouter
Vlan    Interface-ports
---------------------------------------------------------
100     Et3(dynamic), Et4(static)

switch#show mld snooping mrouter detail
Vlan   Intf      Address           FirstHeard   LastHeard    Expires  Type
-------------------------------------------------------------------------------
100    Et3       fe80::200:3ff:fe01:0 0d00h24m30  0d00h00m08  00h00m17 querier
100    Et4       fe80::200:3ff:fe03:0 0d00h24m10  0d00h00m08  00h01m37 pim
100    Et2       0.0.0.0           -           -           -           static

switch##show mld snooping mrouter vlan 100
Vlan    Interface-ports
---------------------------------------------------------
100     Et2(static), Et3(dynamic), Et4(dynamic)
switch#show mld  snooping querier
Vlan  IP Address               Version  Port
-------------------------------------------
100   fe80::200:3ff:fe01:0     v2       Et3
```

- Use the **show mld snooping querier** command to display querier information. In the example, the querie information requested for display if for ***vlan 100***.

```
switch# show mld snooping querier vlan 100
IP Address           : fe80::200:3ff:fe01:0
MLD Version          : v2
Port                 : Et3
Max response time    : 10.0

switch#show mld snooping counters
               Input                |    Output
Port   Queries Reports  Others  Errors|Queries Reports  Others
-------------------------------------------------------------------
Cpu          1     153       0       0     154       6      51
Et1          0     153       0       0     156       0      51
Et2          0     153       0       0     156     110      51
Et3        154       1       0       0       1     610      51
Et4          0     152      51       0     155     453       0
Switch       0       0       0       0       0       0       0

switch#show mld snooping counters errors
           Packet       Packet  Bad IP    Unknown      Bad PIM   Bad ICMP  Bad MLD   Bad MLD
Port       Too Short    Not IP  Checksum  IP Protocol  Checksum  Checksum  Query     Report
```

```
-------------------------------------------------------------------------------------
Cpu                0         0         0         0         0         0         0         0
Et1                0         0         0         0         0         0         0         0
Et2                0         0         0         0         0         0         0         0
Et3                0         0         0         0         0         0         0         0
Et4                0         0         0         0         0         0         0         0
Switch             0         0         0         0         0         0         0         0
```

The **Switch** interface in the above output can be ignored.

16.1.9.34 show mld statistics

The **show mld statistics** command displays total statistics information of incoming and outgoing MLD messages on a specific interface.

Command Mode

EXEC

Command Syntax

show mld statistics version ***value***

Parameters

- **version** specifies MLD version.
- ***value*** specifies the MLD version number. Accepted version values are 1 and 2.

Example

This command displays total MLD statistics on the Ethernet interface ***Et3*** and ***Et6***.

```
switch# show mld statistics
             MLD Total (Version1 + Version2) Statistics
           Received                                    | Sent
Interface  Queries  Reports  Dones   Others   Errors | Queries
-------------------------------------------------------------------
Et3        0        12       0       0        0        12
Et6        0        11       0       0        0        12
```

16.1.9.35 show mld summary

The **show mld summary** command displays MLD summary information.

Command Mode

EXEC

Command Syntax

show mld summary

Parameters

interface displays MLD summary on a specified interface.

Example

This command displays MLD summary on the interface ***Ethernet3*** and ***Ethernet6***.

```
switch# show mld summary
Interface     IPv6 link-local address      Group Count     Querier State
---------     -----------------------      -----------     -------------
Ethernet3     fe80::1:ff:fe01:0            2               querier
Ethernet6     fe80::1:ff:fe01:0            2               querier

Number of MLD interfaces: 2
```

```
Number of total groups joined across all MLD interfaces: 4
```

16.1.9.36 show multicast fib ipv4 software

The **show multicast fib ipv4 software** command displays information about the interfaces and the software-forwarded routes included in the IPv4 multicast forwarding information base (MFIB). Use the **show multicast fib ipv4** command for hardware-forwarded routes.

Parameter options are available to filter output by group address or group and source address.

Command Mode

EXEC

Command Syntax

show multicast fib ipv4 software [**INFO_LEVEL**] [**ROUTE**]

Parameters

- **INFO_LEVEL** specifies the type of information displayed. Options include
  - ***no parameter*** displays packet reception counters.
  - **detail** displays packet reception counters and packet queued/dropped counters.

  :
- **ROUTE** routes displayed, filtered by multicast group and source IP addresses:
  - ***no parameter*** shows information for all software-forwarded routes in the MFIB.
  - ***group_addr*** shows information only for the specified multicast group.
  - ***group_addr source address*** shows information only for the specified group and source.

Examples

- This command displays MFIB information for all software-forwarded routes in the MFIB.

```
switch# show multicast fib ipv4 software
239.255.255.250 172.17.41.150
        Vlan3040 (iif)
        Packets Received: 18
        Bytes Received  : 9147
        RPF Failures    : 0
239.255.255.250 172.17.41.120
        Vlan3040 (iif)
        Packets Received: 6
        Bytes Received  : 966
        RPF Failures    : 0 switch#
```

- This command displays detailed MFIB information for all software-forwarded routes in the MFIB.

```
switch# show multicast fib ipv4 software detail
239.255.255.250 172.17.41.150
Vlan3040 (iif)
Packets Received: 18
Bytes Received: 9147
RPF Failures: 0
Packets Queued/Dropped : 0 / 0
239.255.255.250 172.17.41.120
Vlan3040 (iif)
Packets Received: 6
Bytes Received: 966
RPF Failures: 0
Packets Queued/Dropped : 0 / 0 switch#
```

16.1.9.37 show multicast fib ipv4

The **show multicast fib ipv4** command displays information about interfaces and the hardware-forwarded routes included in the IPv4 Multicast Forwarding Information Base (MFIB).

Command Mode

EXEC

Command Syntax

show multicast fib ipv4 [***group_address*** [***source_address***] | bidirectional | count | counter | df | rpa | software | sparse-mode | static | summary | vrf]

Parameters

- no parameters displays information for all hardware-forwarded routes in the MFIB.
- ***group_address*** displays the information of the specified multicast group address. Options include:
  - ***source_address*** displays the information of the specified multicast group and source addresses.
  - **count** displays the multicast routes count of the specified group address.
  - **counters** displays the multicast route traffic count of the specified group address.
- **bidirectional** displays the information of bidirectional routes.
- **count** displays the count of multicast routes.
- **counter** displays the count of multicast route traffic in either bytes or packets.
- **df** displays the bidirectional Protocol Independent Multicast (PIM) Designated Forwarder (DF) bitmap.
- **rpa** displays the bidirectional PIM Rendezvous Point Address (RPA) index.
- **software** displays the software multicast FIB.
- **sparse-mode** displays the sparse-mode information.
- **static** displays the static multicast information.
- **summary** displays the multicast FIB summary.
- **vrf *vrf_name*** displays information of the corresponding VRF.

Guidelines

The counter is not available (N/A) if a multicast route does not have an associated counter. If the counter value for any source in a group address is N/A, then the sum of counters for the group address is N/A. However, the counter values for other sources are still displayed.

Examples

- This command displays the bidirectional PIM RPA index.

```
switch# show multicast fib ipv4 rpa
Prefix                                  Rpa Index
225.0.0.0/8                             1
226.0.0.0/8                             1
```

- This command displays the static multicast route information.

```
switch# show multicast fib ipv4 static count
(S,G) routes: 34
(*,G) routes: 31
Fastdrop routes:  0
Prefix routes: 12
```

- This command displays the multicast routes count of the specified group and source addresses.

```
switch# show multicast fib ipv4 229.0.0.0 10.1.5.101 count
Activity poll time: 60 seconds
(S,G) routes:  1
Fastdrop routes:  0
```

- This command displays the multicast route traffic count of the specified group and source addresses.

```
switch# show multicast fib ipv4 229.0.0.0 10.1.5.101 counters
Activity poll time: 60 seconds
 229.0.0.0 10.1.5.101
   Byte: 46128
   Packet: 93
   Port-Channel100 (iif)
     Activity 0:53:52 ago
```

- This command displays the multicast FIB summary.

```
switch# show multicast fib ipv4 summary
Number of multicast routes: 12
Number of fastdrop routes : 45
```

16.1.9.38 show multicast fib ipv6

The **show multicast fib ipv6** command displays the Multicast Forwarding Information Base (MFIB) table.

Command Mode

EXEC

Command Syntax

show multicast fib ipv6

Example

The command output displays the Multicast Forwarding Information Base (MFIB) table as shown.

```
switch# show multicast fib ipv6
Activity poll time: 60 seconds
  ff33::1:0:0:1 101:1::2
    Ethernet11/1 (iif)
    Ethernet9/1.1
    Ethernet2/1.1
    Ethernet3/1.1
    Ethernet6/1.1
    Ethernet5/1.1
    Ethernet8/1.1
    Ethernet7/1.1
    Ethernet4/1.1
Activity 0:00:35 ago
```

16.1.9.39 show pim ipv6 sparse-mode route

The **show pim ipv6 sparse-mode route** command displays the PIM Sparse Mode Multicast Routing table.

Command Mode

EXEC

Command Syntax

show pim ipv6 sparse-mode route

Example

The command output displays the PIM Sparse Mode Multicast Routing table as shown.

```
switch# show pim ipv6 sparse-mode route
PIM Sparse Mode Multicast Routing Table
Flags: E - Entry forwarding on the RPT, J - Joining to the SPT
    R - RPT bit is set, S - SPT bit is set, L - Source is attached
    W - Wildcard entry, X - External component interest
    I - SG Include Join alert rcvd, P - (*,G) Programmed in hardware
    H - Joining SPT due to policy, D - Joining SPT due to protocol
    Z - Entry marked for deletion, C - Learned from a DR via a register
    A - Learned via Anycast RP Router, M - Learned via MSDP
    N - May notify MSDP, K - Keepalive timer not running
    T - Switching Incoming Interface, B - Learned via Border Router
RPF route: U - From unicast routing table
           M - From multicast routing table
ff33::1:0:0:1
  101:1::2, 2:03:00, flags: S
    Incoming interface: Ethernet11/1
    RPF route: [U] 101:1::/64 [110/1] via fe80::464c:a8ff:feb7:39e9
    Outgoing interface list:
      Ethernet6/1.1
      Ethernet4/1.1
      Ethernet7/1.1
      Ethernet9/1.1
      Ethernet8/1.1
      Ethernet2/1.1
      Ethernet5/1.1
      Ethernet3/1.1
```

16.1.9.40 show platform fap mroute ipv6

The **`show platform fap mroute ipv6`** command displays the Platform Hardware Forwarding table.

Command Mode

EXEC

Command Syntax

`show platform fap mroute ipv6`

Example

The command output displays the Platform Hardware Forwarding table as shown

```
switch# show platform fap mroute ipv6
Jericho0 Multicast Routes:
--------------------------
Location    GroupId    Group              Source
IIF           McId        OIF
  FLP/TT               FLP/TT         TT
FLP                  FLP           FLP                FLP
------------------------------------------------------------------------
4096/2048   1/1      ff33::1:0:0:23/128  101:1::2/128
Vlan1357 21504     Vlan1044(Et7/1) Vlan1123(Et9/1)


Vlan1200(Et8/1) Vlan1223(Et2/1)


Vlan1226(Et5/1) Vlan1232(Et3/1)
```

16.1.9.41 unresolved cache-entries max

The **`unresolved cache-entries max`** command configures the maximum number of unresolved (S,G) routes that the switch can cache packets. The default buffer size is ***4000*** (S,G) routes.

The **no unresolved cache-entries max** and **default unresolved cache-entries max** commands restore the default unresolved cache-entries buffer size of ***4000*** (S,G) routes by removing the **unresolved cache-entries max** command from ***running-config***. See **ip multicast boundary** to limit the number of cached packets per S,G.

Command Mode

Router Multicast IPv4 Configuration

Router Multicast VRF IPv4 Configuration

Command Syntax

unresolved cache-entries max ***quantity_entries***

no unresolved cache-entries max

default unresolved cache-entries max

Parameter

quantity_entries maximum buffer size (routes). Value ranges from ***10*** to ***10000000***. Default is ***4000***.

Example

This command sets the maximum MFIB unresolved cache-entry buffer size to ***6000*** routes in the default VRF.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# unresolved cache-entries max 6000
switch(config-router-multicast-ipv4)#
```

16.1.9.42 unresolved packet-buffers max

The **unresolved packet-buffers max** command specifies the number of (S,G) multicast packets for an individual route that the switch can process before the (S,G) entry is entered into cache. Packets that are received in excess of this limit before the route is programmed into the cache are dropped. By default, the switch processes 3 unresolved packets for an individual route.

The **no unresolved packet-buffers max** and **default unresolved packet-buffers max** commands restore the number of unresolved packets that the switch processes to the default value of ***3*** packets by removing the **unresolved packet-buffers max** command from ***running-config***. See **unresolved cache-entries max** to limit the number of unresolved routes that are cached.

Command Mode

Router Multicast IPv4 Configuration

Router Multicast VRF IPv4 Configuration

Command Syntax

unresolved packet-buffers max ***quantity_packets***

no unresolved packet-buffers max

default unresolved packet-buffers max

Parameters

quantity_packets packets per unresolved route that the switch processes. Values range from ***3*** to ***10000000***. Default is ***3***.

Example

This command programs the switch in the default VRF to process ***30*** multicast packets from any route regardless of its entrys presence in the multicast routing cache.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)# unresolved packet-buffers max 30
switch(config-router-multicast-ipv4)#
```

16.2 IGMP and IGMP Snooping

- IGMP Snooping
- IGMP Host Proxy Description
- Supported Features
- IGMP Protocols
- Configuring IGMP
- Configuring IGMP Snooping
- IGMP Host Proxy
- IGMP and IGMP Snooping Commands

16.2.1 IGMP Snooping

IGMP snooping is a Layer 2 switch process that extracts lists of hosts receiving multicast group traffic by monitoring IGMP network packets. The switch uses these lists to avoid flooding hosts with extraneous multicast traffic by sending group packets only to group members. Besides preventing local hosts from receiving traffic for groups they did not join, snooping prunes multicast traffic from links that do not contain IGMP clients.

When snooping is enabled, a switch examines IGMP packets sent between hosts connected to network switches and multicast routers (mrouters). When a switch finds an IGMP report from a multicast group recipient, it adds the recipient's port to the group multicast list. When the switch receives an IGMP leave, it removes the recipient's port from the list. Groups are removed upon the group timer expiry. When the switch finds an IGMP query packet or PIM hello packet from a multicast router, it adds the router's port to the port list for all multicast groups.

Snooping Querier

Snooping requires an IGMP querier in the network to create multicast group tables. An IGMP snooping querier performs the multicast router (mrouter) role when the network does not have a router. When the snooping querier is enabled on a VLAN, the switch periodically broadcasts IGMP queries and listens for IGMP Reports that indicate host group memberships.

Networks that contain multiple snooping queriers elect one as the querier, based on IP address. When IGMP snooping querier is enabled on a VLAN, the switch performs as a querier only when it is elected or it is the only snooping querier on the network.

L2 Report Flooding

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. Relying on a single switch to maintain and send report messages can degrade performance. L2 report flooding addresses this by facilitating report message forwarding through any network port. This allows switches to bypass the querier when forwarding multicast traffic to its interested ports.

IGMP Snooping Proxy

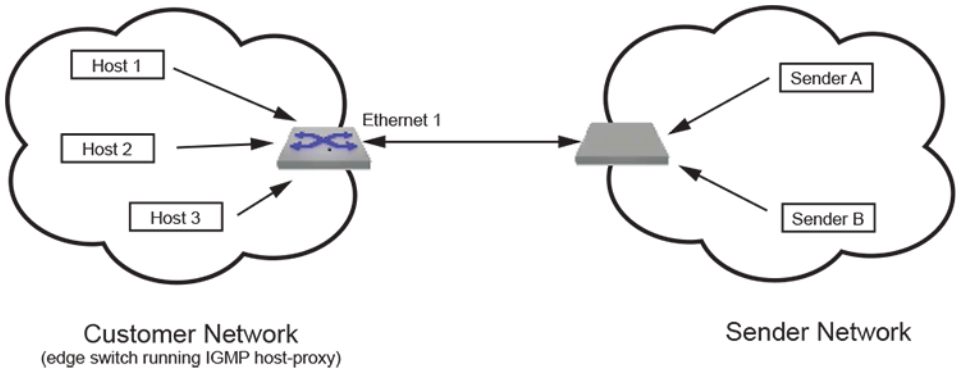
IGMP snooping proxy is an enhancement over IGMP snooping. When snooping proxy is enabled, the switch starts sending proxy queries periodically to the downstream hosts and collects the IGMP reports and updates the local state. Later, when the switch receives an IGMP query from an upstream router, the switch immediately responds with a report based on its local state.

When IGMP snooping proxy is disabled, the IGMP queries in VLAN, and the reports from hosts are flooded. Enabling IGMP snooping proxy prevents a sudden burst in IGMP report traffic in response to every query. It also reduces the number of reports that the IGMP Querier needs to process in the VLAN. However, it introduces a latency in the propagation of the IGMP state through the VLAN.

16.2.2 IGMP Host Proxy Description

The figure displays a typical IGMP host-proxy implementation. The customer network connects to the sender network through the edge switch's ***Ethernet 1*** interface, which is configured as an IGMP host proxy. PIM is enabled within the sender and customer networks but not on the connection between the networks.

Figure 16-2: IP IGMP Host Proxy Implementation



The IGMP proxy agent sends unsolicited IGMP joins when a (S,G) or (*,G) entry arrives in the multicast routing table (mroute table). Subsequently, IGMP reports are sent when queries or group-specific queries arrive on the host proxy interface. When the customer network is void of active listeners, the connection eventually expires and the senders stop transmitting to the network.

IGMP host proxy requires the following:

- PIM Multicast Border Router (MBR) must be enabled on the interface.
- IP IGMP and IP multicast must be enabled.
- The switch must be an RP or in each host's RP path.
- Fast-drop entries are required when there are no interested listeners for the group.

IGMP host proxy is configurable to filter for specific multicast groups and sources.

16.2.3 Supported Features

For a list of the IGMP features that each Arista switch platform supports, refer to the supported features table here: https://www.arista.com/en/support/product-documentation/supported-features.

16.2.4 IGMP Protocols

- IGMP
- IGMP Snooping

16.2.4.1 IGMP

Networks use Internet Group Management Protocol (IGMP) to control the flow of layer 3 multicast traffic. Hosts request and maintain multicast group membership through IGMP messages. Multicast routers use IGMP to maintain a membership list of active multicast groups for each attached network.

- IGMP version 1 is defined in ***RFC 1112***. Hosts can join multicast groups without a method to leave a group. Routers use a timeout-based process to determine when hosts lose interest in a group.
- IGMP version 2 is defined in ***RFC 2236***. Version 2 adds leave messages that hosts use to terminate group membership.
- IGMP version 3 is defined in ***RFC 4604***. Version 3 allows hosts to specify IP addresses within a group from where they receive traffic. Traffic from all other group addresses is blocked from the host.

With respect to each of its attached networks, a multicast router is either a querier or non-querier. Each physical network contains only one querier. A network with more than one multicast router designates the router with the lowest IP address as its querier.

Queriers solicit group membership information by periodically sending General Query messages. Queriers also receive unsolicited messages from hosts joining or leaving a multicast group. When a querier receives a message from a host, it updates its membership list for the group referenced in the message and the network where the message originated.

Queriers forward multicasts from remote sources only to networks as specified by its membership list. If a querier does not receive a report from a network host for a specific group, it removes the corresponding entry from the table and discontinues forwarding multicasts for that group on the network. Queriers also send group-specific queries after receiving a leave request from a host to determine if the network still contains active multicast group members. If it does not receive a membership report during the period defined by the **last member query response interval**, the querier removes the group-network entry from the membership list.

When a host receives a General Query, it responds with Membership Report messages for each of its multicast groups within the interval specified by the Max Response Time field in the query. IGMP suppresses multiple messages from different hosts on a network for the same group. Hosts send unsolicited Membership reports to join a multicast group and send leave messages to exit a group.

16.2.5 Configuring IGMP

This section describes the following configuration tasks:

- Enabling IGMP
- Configuring IGMP Settings

16.2.5.1 Enabling IGMP

Enabling PIM also enables IGMP on that interface. When the switch fills the multicast routing table, it only adds interfaces when the interface receives join messages from downstream devices or when the interface is directly connected to a member of the IGMP group.

By default, PIM and IGMP are disabled on an interface. Use the **pim ipv4 sparse-mode** or **pim ipv4 bidirectional** command to enable PIM and IGMP on the configuration mode interface.

Example

This command enables PIM and IGMP on ***interface vlan 8***.

```
switch(config)# interface vlan 8
switch(config-if-Vl8)# pim ipv4 sparse-mode
switch(config-if-Vl8)#
```

In the unlikely event that the IGMP agent needs to run on an interface without PIM being enabled, use the **ip igmp** command.

Example

This command enables IGMP on ***interface vlan 8*** without enabling PIM.

```
switch(config)# interface vlan 8
switch(config-if-Vl8)# ip igmp
switch(config-if-Vl8)#
```

16.2.5.2 Configuring IGMP Settings

An interface that runs IGMP uses default protocol settings unless otherwise configured. The switch provides commands that alter startup query, last member query, and normal query settings.

IGMP Version

The switch supports IGMP versions 1 through 3. The **ip igmp version** command configures the IGMP version on the configuration mode interface. Version 3 is the default IGMP version.

Example

This command configures ***IGMP version 3*** on ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp version 3
switch(config-if-Vl4)#
```

Startup Query

Membership queries are sent at an increased frequency immediately after an interface starts up to quickly establish the group state. Query count and query interval commands adjust the period between membership queries for a specified number of messages.

The **ip igmp startup-query-interval** command specifies the interval between membership queries that an interface sends immediately after it starts up. The **ip igmp startup-query-count** command specifies the number of queries that the switches sends from the interface at the startup interval rate.

- **Example**

  These commands define a startup interval of ***15*** seconds for the first ***10*** membership queries sent from ***interface vlan 12***.

  ```
  switch(config)# interface vlan 12
  switch(config-if-Vl12)# ip igmp startup-query-interval 150
  switch(config-if-Vl12)# ip igmp startup-query-count 10
  switch(config-if-Vl12)#
  ```

Membership Queries

The router with the lowest IP address on a subnet sends membership queries as the IGMP querier. When a membership query is received from a source with a lower IP address, the router resets its query response timer. Upon timer expiry, the router begins sending membership queries. If the router subsequently receives a membership query originating from a lower IP address, it stops sending membership queries and resets the query response timer.

The **ip igmp query-interval** command configures the frequency at which the active interface, as an IGMP querier, sends membership query messages.

The **igmp query-max-response-time** command configures the time that a host has to respond to a membership query.

Example

These commands define a membership query interval of ***75*** seconds and a query response timer reset value of ***45*** seconds for queries sent from ***interface vlan 15***.

```
switch(config)# interface vlan 15
switch(config-if-Vl15)# ip igmp query-interval 75
switch(config-if-Vl15)# igmp query-max-response-time 450
switch(config-if-Vl15)#
```

Last Member Query

When the querier receives an IGMP leave message, it verifies the group has no remaining hosts by sending a set of group-specific queries at a specified interval. If the querier does not receive a response to the queries, it removes the group state and discontinues multicast transmissions.

The **ip igmp last-member-query-count** (LMQC) command specifies the number of query messages the router sends in response to a group-specific or group-source-specific leave message.

The **ip igmp last-member-query-interval** command configures the transmission interval for sending group-specific or group-source-specific query messages to the active interface.

Example

These commands program the switch to send ***3*** query messages, one every ***25*** seconds, when ***interface vlan 15*** receives an IGMP leave message.

```
switch(config)# interface vlan 15
switch(config-if-Vl15)# ip igmp last-member-query-interval 250
switch(config-if-Vl15)# ip igmp last-member-query-count 3
switch(config-if-Vl15)#
```

Static Groups

The **ip igmp static-group** command configures the configuration mode interface as a static member of the multicast group at the specified address. The router forwards multicast group packets through the interface without otherwise appearing or acting as a group member. No interface is a static member of a multicast group by default.

Note: To become a static member of a multicast group, the switch must be the PIM designated router (DR) for the network. If it is not, you can use the **pim ipv4 dr-priority** command to make it the DR by configuring its PIM DR value to be the highest on the network.

Example

These commands configure ***interface vlan 15*** as the PIM designated router, then configure it as a static member of the multicast group at address ***231.1.1.15*** for multicast data packets that originate at ***10.1.1.1***.

```
switch(config)# interface vlan 15
switch(config-if-Vl15)# pim ipv4 dr-priority 5000
switch(config-if-Vl15)# ip igmp static-group 231.1.1.45 10.1.1.1
switch(config-if-Vl15)#
```

16.2.6 Configuring IGMP Snooping

This section describes the following configuration tasks:

- Enabling Snooping
- Configuring Snooping Parameters
- Snooping Querier
- IGMP Snooping L2 Report Flooding
- IGMP Snooping Filters

- Configuring IGMP Snooping Proxy

16.2.6.1 Enabling Snooping

The switch provides two control settings for snooping IGMP packets:

- Global settings control the availability of IGMP snooping on the switch. Snooping is globally enabled by default.
- Per-VLAN settings control IGMP on individual VLANs. If snooping is enabled on the VLAN, it follows the global snooping state.

The **ip igmp snooping** command controls the global snooping setting. The **ip igmp snooping vlan** command configures snooping on individual VLANs.

Examples

- This command globally enables snooping on the switch.

```
switch(config)# ip igmp snooping
switch(config)#
```

- This command disables snooping on VLANs ***2***, ***3***, and ***4***.

```
switch(config)# no ip igmp snooping vlan 2-4
switch(config)#
```

16.2.6.2 Configuring Snooping Parameters

Specifying a Static Multicast Router Connection

The ip igmp snooping vlan multicast-router command statically configures a port that connects to a multicast router to join all multicast groups. The port to the router must be in the specified VLAN range.

Snooping may not always be able to locate the IGMP querier. This command is for IGMP queriers that are known to connect through the network to a port on the switch.

Example

This command configures the static connection to a multicast router through Ethernet port 3.

```
switch(config)# ip igmp snooping vlan 2 mrouter interface ethernet 3
switch(config)#
```

Adding a Port to a Multicast Group

The **ip igmp snooping vlan member** command adds an a port to a multicast group. The IP address must be an unreserved IPv4 multicast address. The interface to the port must be in the specified VLAN range.

Example

This command configures the static connection to a multicast group at ***237.2.1.4*** through ***interface ethernet 3***.

```
switch(config)# ip igmp snooping vlan 7 member 237.2.1.4 interface ethernet 3
switch(config)#
```

Robustness Variable

The robustness variable specifies the number of unacknowledged snooping queries that a switch sends before removing the recipient from the group list.

The **ip igmp snooping robustness-variable** command configures the robustness variable for all snooping packets sent from the switch. The default value is ***2***.

Example

This command sets the robustness-variable value to ***3***.

```
switch(config)# ip igmp snooping robustness-variable 3
switch(config)#
```

Configuring Interface Startup Initial Query Times

The **ip igmp snooping interface-restart-query** command configures the interface startup initial query times in milliseconds. If nothing is configured, a default value of ***2000*** milliseconds is used. Issuing the command replaces any values already configured. Multiple values may be input in a single command; this makes the mechanism more resilient in the case of dropped packets.

Examples

- This command configures interfaces to send IGMP queries at ***1000***, ***2000***, and ***4000*** milliseconds (i.e., ***1*** second, ***2*** seconds, and ***4*** seconds) after an interface restart or spanning tree change.

```
switch(config)# ip igmp snooping interface-restart-query 1000 2000 4000
switch(config)#
```

- This command configures interfaces to send a single IGMP query of ***5000*** milliseconds (***5*** seconds) after an interface restart or spanning tree change.

```
switch(config)# ip igmp snooping interface-restart-query 5000
switch(config)#
```

16.2.6.3 Snooping Querier

The IGMP snooping querier supports snooping by sending Layer 2 membership queries to hosts attached to the switch. Note that if IGMP snooping is enabled, QoS will not apply to IGMP packets.

16.2.6.3.1 Enabling the Snooping Querier

Enabling the snooping querier on an interface requires the explicit configuration of a global querier address or a local querier address for the interface. See **Configuring Snooping Querier Parameters**.

The switch provides two control settings for controlling the snooping querier:

- The global setting controls the querier on VLANs for which there is no snooping querier command.
- VLAN querier settings take precedence over the global querier setting.

The **ip igmp snooping querier** command controls the global querier setting. When enabled globally, the querier is controlled on individual VLANs through the **ip igmp snooping vlan querier** command.

The **ip igmp snooping vlan querier** command controls the querier for the specified VLANs. VLANs follow the global querier setting unless overridden by one of these commands:

- **ip igmp snooping vlan querier** enables the querier on specified VLANs.
- **no ip igmp snooping vlan querier** disables the querier on specified VLANs.

Example

- These commands globally enables the snooping querier on the switch, explicitly disables snooping on VLANs ***1-4***, and explicitly enables snooping on VLANs ***5-8***.

```
switch(config)# ip igmp snooping querier
```

```
switch(config)# no ip igmp snooping vlan 1-4 querier
switch(config)# ip igmp snooping vlan 5-8 querier
switch(config)#
```

- This command removes the querier setting for VLANs ***3-6***:

```
switch(config)# default ip igmp snooping vlan 3-6 querier
switch(config)#
```

Globally Set the Snooping Querier Version

The **ip igmp snooping querier version** command configures the IGMP snooping querier version. ***Version 2*** is the default IGMP snooping version.

Example

This command globally configures IGMP snooping querier ***version 2***.

```
switch(config)# ip igmp snooping querier version 2
switch(config)#
```

The **ip igmp snooping vlan querier version** command configures IGMP globally on the VLAN. Version 2 is the default IGMP snooping version.

Example

This command configures IGMP snooping vlan querier version ***vlan 5***.

```
switch(config)# ip igmp snooping vlan 5 querier version 2
switch(config)#
```

16.2.6.3.2 Configuring Snooping Querier Parameters

Querier Address

The switch provides two IP addresses for setting the querier source:

- The global address is used by VLANs for which there is no querier address command.
- VLAN querier address settings take precedence over the global querier address.

The snooping querier address specifies the source IP address for IGMP snooping query packets that the switch transmits. The source address is also used to elect a snooping querier when the subnet contains multiple snooping queriers.

The default global querier address is not defined. When the configuration includes a snooping querier, a querier address must be defined globally or for each interface that enables a querier.

The **ip igmp snooping querier address** command sets the global querier source IP address for the switch. VLANs use the global address unless overwritten with the **`ip igmp snooping vlan querier address`** command. The default global address is not defined.

The **ip igmp snooping vlan querier address** command sets the source IP address for query packets transmitted from the specified VLAN. This command overrides the **`ip igmp snooping querier address`** for the specified VLAN.

Examples

- This command sets the source IP address for query packets that the switch transmits to ***10.1.1.41***.

```
switch(config)# ip igmp snooping querier address 10.1.1.41
switch(config)#
```

- This command sets the source IP address for query packets that ***vlan 2*** transmits to ***10.14.1.1***.

```
switch(config)# ip igmp snooping vlan 2 querier address 10.14.1.1
switch(config)#
```

Membership Query Interval

The query interval is the period (seconds), between IGMP Membership Query message transmissions. The interval ranges from ***5*** to ***3600*** seconds.

The **ip igmp snooping querier query-interval** command specifies the global query interval for packets the switch sends as a snooper querier. The default global setting is ***125*** seconds.

The **ip igmp snooping vlan querier query-interval** command specifies the query interval for packets sent from the snooping querier to the specified VLAN, overriding the global setting. VLANs that do not specify a query interval use the global setting.

Examples

- This command sets a query interval of ***150*** seconds for queries transmitted from VLANs for which a query interval is not configured.

```
switch(config)# ip igmp snooping querier query-interval 150
switch(config)#
```

- This command sets the query interval of ***240*** seconds for queries transmitted from ***vlan 2***.

```
switch(config)# ip igmp snooping vlan 2 querier query-interval 240
switch(config)#
```

Membership Query Response Interval

The Max Response Time field, in Membership Query messages, specifies the longest time a host can wait before responding with a Membership Report message. In all other messages, the sender sets the field to zero and the receiver ignores it. The switch provides two values for setting this field:

- The global value is used by VLANs for which there is no Max Response Time command.
- VLAN values take precedence over the global value for the specified VLAN.

The **ip igmp snooping querier max-response-time** command specifies the global Max Response Time value used in snooping query packets transmitted from the switch. Values range from ***1*** to ***25*** seconds with a default of ***10*** seconds. VLANs use the global setting unless overwritten with the **ip igmp snooping vlan querier max-response-time** command.

The **ip igmp snooping vlan querier max-response-time** command configures the Max Response Time field contents for packets transmitted from the specified VLAN, overriding the global setting.

Examples

- This command sets the maximum response time of ***15*** seconds for queries transmitted from VLANs for which a maximum response time is not configured.

```
switch(config)# ip igmp snooping querier max-response-time 15
switch(config)#
```

- This command sets a maximum response time of ***5*** seconds for queries that ***vlan 2*** transmits.

```
switch(config)# ip igmp snooping vlan 2 querier max-response-time 5
switch(config)#
```

Last Member Query

When the querier receives an IGMP leave message, it verifies the group has no remaining hosts by sending a set of group-specific queries at a specified interval. If the querier does not receive a response to the queries, it removes the group state and discontinues multicast transmissions.

The switch provides two values for setting this field:

- The global value is used by VLANs for which there is no last-member-query-interval defined.
- VLAN values take precedence over the global value for the specified VLAN.

The **ip igmp snooping querier last-member-query-interval** command specifies the global last-member-query-interval used in snooping query packets transmitted from the switch. This value is used for VLANs that do not have a value specified. Values range from ***1*** to ***25*** seconds with a global default of one second.

The **ip igmp snooping vlan querier last-member-query-interval** command configures the last-member-query-interval field contents for packets transmitted from the specified VLAN, overriding the global setting.

Example

This command sets the global snooping querier last-member-query-interval to ***5*** seconds and the ***vlan 10*** last-member-query-interval to ***12*** seconds.

```
switch(config)# ip igmp snooping querier last-member-query-interval 5
switch(config)# ip igmp snooping vlan 10 querier last-member-query-interval 12
switch(config)#
```

Interface Restart Query Spoofing

When the port status (link status or spanning tree status) changes, an IGMP general query is spoofed based on the information of the last known IGMP querier. This facilitates faster network convergence time.

By default, interfaces wait ***2000*** milliseconds before sending the spoofed IGMP query. To configure the delay before the spoofed query is sent, use the **ip igmp snooping interface-restart-query** command. This setting is applied to all ports.

Example

This command configures the switch to send general IGMP queries at ***100*** milliseconds, ***200*** milliseconds, and ***300*** milliseconds after interface restart or spanning tree status change.

```
switch(config)# ip igmp snooping interface-restart-query 100 200 300
```

16.2.6.4 IGMP Snooping L2 Report Flooding

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. Report flooding is disabled by default and must be enabled globally before it can be enabled on individual interfaces.

The list of ports that can forward membership report messages must be explicitly configured. Commands are available to define lists of ports that are valid for all VLANs and port lists that are valid for specified VLAN ranges. Ports can forward membership reports only if they are configured to handle VLAN traffic, regardless of any report flooding configuration settings.

Enabling L2 Report Flooding

These commands enable L2 report flooding:

- **ip igmp snooping report-flooding** enables report flooding globally.
- **ip igmp snooping vlan report-flooding** enables report flooding on a specified VLAN range.

Example

These commands enable L2 report flooding globally, and on VLANs ***201-205***.

```
switch(config)# ip igmp snooping report-flooding
switch(config)# ip igmp snooping vlan 201-205 report-flooding
switch(config)#
```

Configuring Forwarding Ports

These commands specify the ports that forward membership report messages:

- **ip igmp snooping report-flooding switch-port** configures ports globally.
- **ip igmp snooping vlan report-flooding switch-port** configures ports for a specified VLAN range.

Example

These commands enable Ethernet ports ***5-9*** to forward reports on all VLANs and ports ***12-15*** on VLANs ***201-205***.

```
switch(config)# ip igmp snooping report-flooding switch-port ethernet 5-9
switch(config)# ip igmp snooping vlan 201-205 report-flooding switch-port
 ethernet
12-15
switch(config)#
```

16.2.6.5 IGMP Snooping Filters

IGMP snooping filters assigns IGMP profiles only to Layer 2 interfaces, and for Layer 3 interfaces use multicast boundary filters to control the multicast groups that the interfaces can join. An IGMP profile specifies a filter type and a list of address ranges. The address ranges comprise the multicast groups covered by the profile. The filter type determines an interface's accessibility to the multicast groups:

- Permit filters define the multicast groups the interface can join.
- Deny filters define the multicast groups the interface cannot join.

Profiles are created in IGMP-profile configuration mode, then applied to an interface in interface configuration mode.

The **ip igmp profile** command places the switch in IGMP profile configuration mode. The **permit / deny** and **range** commands specify the profile's filter type and address range. A profile may contain multiple range statements to define a discontiguous address range.

Example

These commands create an IGMP profile named ***list_1*** by entering IGMP-profile configuration mode, configure the profile to permit multicast groups ***231.22.24.0*** through ***231.22.24.127***, and return the switch to ***global*** configuration mode.

```
switch(config)#ip igmp profile list_1
switch(config-igmp-profile-list_1)#permit
switch(config-igmp-profile-list_1)#range 231.22.24.0 231.22.24.127
switch(config-igmp-profile-list_1)#exit
switch(config)#
```

The **ip igmp snooping filter** command applies an IGMP profile to the configuration mode interface.

Example

These commands apply the ***list_1*** snooping profile to ***interface ethernet 7***.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#ip igmp snooping filter list_1
```

```
switch(config-if-Et7)#
```

16.2.6.5.1 Verifying IGMP Snooping

Show commands are available to display various configurations and IGMP snooping status. IGMP snooping that are viewable include:

- **show ip igmp snooping**
- **show ip igmp snooping counters**
- **show igmp snooping querier**
- **show igmp snooping querier counters**
- **show igmp snooping querier membership**

IGMP Snooping Status

The **show ip igmp snooping** command displays the switch's IGMP snooping configuration.

Example

This command displays the switch's IGMP snooping configuration.

```
switch>show ip igmp snooping
   Global IGMP Snooping configuration:
-------------------------------------------
IGMP snooping                   : Enabled
Robustness variable             : 2

Vlan 1 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 20 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 2028 :

switch>
```

IGMP Snooping Counters

The **show ip igmp snooping counters** command displays the number of IGMP messages sent and received through each switch port. The display table sorts the messages by type.

Example

This command displays the number of messages received on each port.

```
switch>show ip igmp snooping counters
                        Input                 |            Output
Port   Queries Reports  Leaves  Others  Errors|Queries Reports  Leaves  Others
-------------------------------------------------------------------------------
Cpu      15249  106599       4  269502       0   30242  102812     972    3625
Et1          0       0       0       0       0       0       0       0       0
Et2          0       6       1      26       0    5415       0       0     731
Et3          0   10905     222    1037       0   15246       0       0    1448
Et4          0   44475      21     288       0   15247       0       0    2199
Et5          0     355       0      39       0   15211       0       0    2446
Et6          0     475      13       0       0   15247       0       0    2487
Et7          0       0       0     151       0   15247       0       0    2336
Et8          0     578       6      75       0    2859       0       0     931
Et9          0       0       0      27       0   15247       0       0    2460
```

```
Et10         0   12523     345      54       0   15247       0       0    2433
Et11         0       0       0       0       0       0       0       0       0
Et12         0    4509      41      22       0   15247       0       0    2465
Et13         0     392      29     119       0   15247       0       0    2368
Et14         0      88       3       6       0   15247       0       0    2481
Et15         0   16779     556      72       0   15117       0       0      66
Et16         0    2484      13      66       0   15247       0       0    2421
Et17         0       0       0       0       0       0       0       0       0
Et18         0      20       6     160       0    3688       0       0     803
Et19         0    4110      17       0       0   15247       0       0    2487
Et20         0       0       0       0       0       0       0       0       0
Et21         0       0       0       0       0       0       0       0       0
Et22         0       0       0      52       0   15247       0       0    2435
Et23         0    5439     181     138       0   15247       0       0    2349
Et24         0    2251      21       4       0   15247       0       0    2483
Po1      45360  540670    8853  464900       0   15249  224751     618    2576
Po2          0  101399      58      17       0   15120       0       0    1121
Switch       0       0       0       0       0       0       0       0       0
```

IGMP Snooping Querier

The **show igmp snooping querier** command displays snooping querier configuration and status information. Command provides options to only include specific VLANs.

Example

This command displays the querier IP address, version, and port servicing each VLAN.

```
switch>show igmp snooping querier
Vlan  IP Address       Version  Port
-------------------------------------
1     172.17.0.37      v2       Po1
20    172.17.20.1      v2       Po1
26    172.17.26.1      v2       Cpu
2028  172.17.255.29    v2       Po1 switch>
```

IGMP Snooping Querier Counters

The **show igmp snooping querier counters** command displays the counters from the querier, as learned through Internet Group Management Protocol (IGMP).

Example

This command displays the counters from the querier.

```
switch>show igmp snooping querier counters
---------------------------------------------------------------------
Vlan: 1    IP Addr: 100.0.0.1       Op State: Querier     Version:  v3

v1 General Queries Sent         :0
v1 Queries Received             :0
v1 Reports Received             :0
v2 General Queries Sent         :1
v2 Queries Received             :0
v2 Reports Received             :25
v2 Leaves Received              :0
v3 General Queries Sent         :655
v3 GSQ Queries Sent             :0
v3 GSSQ Queries Sent            :8
v3 Queries Received             :654
v3 Reports Received             :2385
Error Packets                   :0
Other Packets                   :0 switch>
```

IGMP Snooping Querier Membership

The **show igmp snooping querier membership** command displays the membership from the querier, as learned through Internet Group Management Protocol (IGMP).

Example

This command displays the membership from the querier from ***vlan 1***.

```
switch>show igmp snooping querier membership
-------------------------------------------------------------------------
Vlan: 1    Elected: 100.0.0.1       QQI: 125  QRV: 2  QRI: 10  GMI: 260

Groups            Mode  Ver  Num of Sources
-------------------------------------------------------------------------
10.0.0.2         EX    v3   0 []
10.0.0.3         IN    v3   2 [ 3.3.3.3, 3.3.3.4 ]
10.0.0.4         EX    v3   0 []
10.0.0.13        EX    v3   0 []
10.0.0.22        EX    v3   0 []
10.0.0.1         IN    v3   3 [ 5.6.7.9, 5.6.7.8, ... ]
switch>
```

16.2.6.6 Configuring IGMP Snooping Proxy

Use the ip igmp snooping proxy command to enable IGMP snooping proxy globally. Enabling IGMP snooping proxy enables it for all VLANs where IGMP snooping is enabled. IGMP snooping proxy is globally disabled by default.

Use the **ip igmp snooping proxy** command to enable IGMP snooping proxy globally. Use the no ip igmp snooping vlan proxy command to disable IGMP snooping proxy on specified VLANs.

Examples

- This command globally enables IGMP snooping proxy on the switch.

```
switch(config)# ip igmp snooping proxy
switch(config)#
```

- This command disables IGMP snooping proxy on ***vlan 2***, ***vlan 3***, and ***vlan 4***.

```
switch(config)# no ip igmp snooping proxy vlan 2-4 proxy
switch(config)#
```

16.2.6.6.1 Configuring Snooping Proxy Querier

To configure the IGMP snooping proxy querier use the existing **ip igmp snooping querier** commands. For more information on these commands, refer to **IGMP and IGMP Snooping Commands**.

Note: The proxy querier by default uses ***0.0.0.0*** as the IP address.

Example

In this example, IGMP snooping proxy is enabled using the **ip igmp snooping proxy** command and the snooping proxy is set to reports for all the VLANs except VLANs ***100*** through ***110*** using the **ip igmp snooping vlan** command. The proxy querier operates in version 3 and sends queries at a ***15***-second interval and hosts can take up to ***5*** seconds to respond.

```
switch(config)#ip igmp snooping proxy
switch(config)#no ip igmp snooping vlan 100-110 proxy
```

```
switch(config)#ip igmp snooping querier query-interval 15
switch(config)#ip igmp snooping querier max-response-time 5
switch(config)#ip igmp snooping querier version 3
switch(config)#ip igmp snooping querier
switch(config)#
```

16.2.7 IGMP Host Proxy

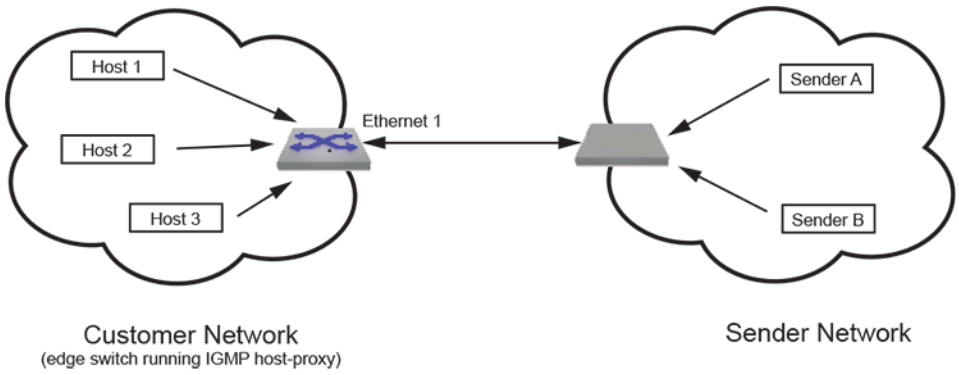
Interfaces on the switch can be configured to serve as IGMP host proxies. An IGMP host proxy exchanges IGMP reports (joins/leaves) between networks whose connection does not support PIM along network boundaries.

- IGMP Host Proxy Description
- IGMP Host Proxy Configuration

16.2.7.1 IGMP Host Proxy Description

The figure displays a typical IGMP host-proxy implementation. The customer network connects to the sender network through the edge switch's ***Ethernet 1*** interface, which is configured as an IGMP host proxy. PIM is enabled within the sender and customer networks but not on the connection between the networks.

Figure 16-3: IP IGMP Host Proxy Implementation



The IGMP proxy agent sends unsolicited IGMP joins when a (S,G) or (*,G) entry arrives in the multicast routing table (mroute table). Subsequently, IGMP reports are sent when queries or group-specific queries arrive on the host proxy interface. When the customer network is void of active listeners, the connection eventually expires and the senders stop transmitting to the network.

IGMP host proxy requires the following:

- PIM Multicast Border Router (MBR) must be enabled on the interface.
- IP IGMP and IP multicast must be enabled.
- The switch must be an RP or in each host's RP path.
- Fast-drop entries are required when there are no interested listeners for the group.

IGMP host proxy is configurable to filter for specific multicast groups and sources.

16.2.7.2 IGMP Host Proxy Configuration

Enabling IGMP Host Proxy

Enable PIM MBR on the interface using the **pim ipv4 border-router** command. The IGMP host proxy service is then configured on the interface using the **ip igmp host-proxy** command. When the host proxy is configured, it sends reports for (S,G) entries in the multicast routing (mroute) table if these are the only routes there; if there are any (*,G) entries, it sends reports only for these. To send reports for a specific group even

when there is no (*, G) entry in the mroute table for that group, include the group address in the **ip igmp host-proxy** command. Multiple **ip igmp host-proxy** statements are required to specify multiple groups. The interval between IGMP reports is configured by **ip igmp host-proxy report-interval**.

Host Proxy IGMP Version and Source Filtering

IGMP host proxies can be configured with IGMP versions 1, 2, or 3, and uses version 3 by default. When the host-proxy IGMP version is set to 3, the proxy can explicitly include or exclude source addresses. Otherwise, include/exclude configuration for source addresses is ignored. The IGMP version of unsolicited reports is specified with the **ip igmp host-proxy version** command. Reports that are triggered by IGMP queries, however, are sent in the same IGMP version as the received query. (An interface may also have a different IGMP version configured on it for other purposes using the **ip igmp version** command.)

Using ACLs

IGMP host proxy can also be enabled for the addresses defined by an ACL; if one or more groups are configured in addition to ACLs, the groups are processed first. Implicit deny in the ACL is ignored, but if the ACL includes an explicit deny rule, then the interface sends joins only to groups configured directly on the interface or included in a permit ACL. Deny rules take precedence over permit rules. If a group is configured with no filters and a host-proxy is configured with an ACL with rules having filters for the group, or configured with groups and source filters, then the filters are applied to the group.

Disabling Host Proxy or Removing an Individual Group or Source

The **no igmp host-proxy** command can be entered with group or source parameters to remove the specified group or source from the list. Entering the **no igmp host-proxy** command without specifying group or source disables the forwarding of all IGMP reports on the interface.

Examples

- These commands enable IGMP host proxy on ***interface ethernet 17*** for all multicast group addresses.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# pim ipv4 border-router
switch(config-if-Et17)# ip igmp host-proxy
switch(config-if-Et17)#
```

- These commands enable IGMP host proxy on ***interface ethernet 18*** for the multicast group at ***231.10.10.1***. The list of source addresses is not restricted.

```
switch(config)# interface ethernet 18
switch(config-if-Et18)# pim ipv4 border-router
switch(config-if-Et18)# ip igmp host-proxy 231.10.10.1
switch(config-if-Et18)#
```

- These commands enable IGMP host proxy on ***interface ethernet 19*** for the multicast group at ***231.10.10.2***. The list of source addresses only excludes ***10.4.4.1*** and ***10.4.5.2***.

```
switch(config)# interface ethernet 19
switch(config-if-Et19)# pim ipv4 border-router
switch(config-if-Et19)# ip igmp host-proxy 231.10.10.2 exclude 10.4.4.1
switch(config-if-Et19)# ip igmp host-proxy 231.10.10.2 exclude 10.4.5.2
switch(config-if-Et19)#
```

- These commands enable IGMP host proxy on ***interface ethernet 16*** for the multicast group at ***231.10.10.3***. The list of source address for this group only includes ***10.5.5.1*** and ***10.5.5.2***.

```
switch(config)# interface ethernet 16
switch(config-if-Et16)# pim ipv4 border-router
switch(config-if-Et16)# ip igmp host-proxy 231.10.10.3 include 10.5.5.1
```

```
switch(config-if-Et16)# ip igmp host-proxy 231.10.10.3 include 10.5.5.2
switch(config-if-Et16)#
```

- These commands configure an IGMP host proxy interval of ***5*** seconds on ***interface port-channel 100***.

```
switch(config)# interface port-channel 100
switch(config-if-Po100)# ip igmp host-proxy report-interval 5
switch(config-if-Po100)#
```

- These commands enable IGMP host proxy on ***interface ethernet 17*** for the group address(es) specified in ACL ***acl1***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# pim ipv4 border-router
switch(config-if-Et17)# ip igmp host-proxy access-list acl1
switch(config-if-Et17)#
```

16.2.8 IGMP and IGMP Snooping Commands

IGMP Configuration Commands (Interface Configuration Mode)

- igmp query-max-response-time
- ip igmp last-member-query-count
- ip igmp last-member-query-interval
- ip igmp query-interval
- ip igmp router-alert
- ip igmp startup-query-count
- ip igmp startup-query-interval
- ip igmp static-group
- ip igmp static-group acl
- ip igmp static-group range
- ip igmp version

IGMP Clear Commands

- clear ip igmp group
- clear ip igmp statistics

IGMP Display Commands

- show ip igmp groups
- show ip igmp groups count
- show ip igmp interface
- show ip igmp static-groups
- show ip igmp static-groups acl
- show ip igmp static-groups group
- show ip igmp statistics

IGMP Snooping Configuration Commands (Global Configuration Mode)

- ip igmp
- ip igmp profile
- ip igmp snooping
- ip igmp snooping interface-restart-query
- ip igmp snooping proxy

- ip igmp snooping querier
- ip igmp snooping querier address
- ip igmp snooping querier last-member-query-count
- ip igmp snooping querier last-member-query-interval
- ip igmp snooping querier max-response-time
- ip igmp snooping querier query-interval
- ip igmp snooping querier startup-query-count
- ip igmp snooping querier startup-query-interval
- ip igmp snooping querier version
- ip igmp snooping report-flooding
- ip igmp snooping report-flooding switch-port
- ip igmp snooping restart query-interval
- ip igmp snooping robustness-variable
- ip igmp snooping vlan
- ip igmp snooping vlan fast-leave
- ip igmp snooping vlan max-groups
- ip igmp snooping vlan member
- ip igmp snooping vlan multicast-router
- ip igmp snooping vlan proxy
- ip igmp snooping vlan querier
- ip igmp snooping vlan querier address
- ip igmp snooping vlan querier last-member-query-count
- ip igmp snooping vlan querier last-member-query-interval
- ip igmp snooping vlan querier max-response-time
- ip igmp snooping vlan querier query-interval
- ip igmp snooping vlan querier startup-query-count
- ip igmp snooping vlan querier startup-query-interval
- ip igmp snooping vlan querier version
- ip igmp snooping vlan report-flooding
- ip igmp snooping vlan report-flooding switch-port

IGMP Configuration Commands (Interface Configuration Mode)

- ip igmp snooping filter

IGMP Snooping Clear Commands

- clear ip igmp snooping counters

IGMP Snooping Display Commands

- show igmp snooping querier
- show igmp snooping querier counters
- show igmp snooping querier membership
- show ip igmp profile
- show ip igmp snooping
- show ip igmp snooping counters
- show ip igmp snooping counters ethdev-pams
- show ip igmp snooping groups
- show ip igmp snooping groups count
- show ip igmp snooping mrouter
- show ip igmp snooping report-flooding

IGMP Profile Configuration Mode Commands

- permit / deny
- range

IGMP Host Proxy Commands

- ip igmp host-proxy
- ip igmp host-proxy report-interval
- ip igmp host-proxy version
- show ip igmp host-proxy config-sanity
- show ip igmp host-proxy interface

16.2.8.1 clear ip igmp group

The **`clear ip igmp group`** command deletes IGMP cache entries as follows:

- **`clear ip igmp group`** all entries from the IGMP cache.
- **`clear ip igmp group`** ***gp_addr*** all entries for a specified multicast group.
- **`clear ip igmp group interface`** ***int_id*** all entries that include a specified interface.
- **`clear ip igmp group`** ***gp_addr*** **interface** ***int_id*** all entries for a specified interface in a specified group.

Command Mode

Privileged EXEC

Command Syntax

`clear ip igmp group` **[** ***gp_addr***][interface **INT_ID**]

Parameters

- ***gp_addr*** multicast group IP address (dotted decimal notation).
- **INT_ID** interface name. Options include:
  - **ethernet** ***e_num*** Ethernet interface specified by ***e_num***.
  - **loopback** ***l_num*** Loopback interface specified by ***l_num***.
  - **management** ***m_num*** Management interface specified by ***m_num***.
  - **port-channel** ***p_num*** Port-channel interface specified by ***p_num***.
  - **vlan** ***v_num*** VLAN interface specified by ***v_num***.
  - **vxlan** ***vx_num*** VXLAN interface specified by ***vx_num***.

Examples

- This command deletes all IGMP cache entries for the multicast group ***231.23.23.14***.

```
switch# clear ip igmp group 231.23.23.14
switch#
```

- This command deletes IGMP cache entries for ***interface ethernet 16*** in multicast group ***226.45.10.45***.

```
switch# clear ip igmp group 226.45.10.45 interface ethernet 16
switch#
```

16.2.8.2 clear ip igmp snooping counters

The **`clear ip igmp snooping counters`** command resets the snooping message counters for the specified interface. The snooping counters for all interfaces are reset if the command does not include an interface name.

The **`show ip igmp snooping counters`** command displays the counter contents. See the **show ip igmp snooping counters** command description for a list of available snooping counters.

Command Mode

Privileged EXEC

Command Syntax

`clear ip igmp snooping counters` [**INT_NAME**]

Parameters

INT_NAME interface name. Formats include:

- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **port-channel *p_num*** Port-channel interface specified by ***p_num***.
- **switch** virtual interface to an L2 querier.

Example

This command clears the snooping counters for messages received on ***interface ethernet 15***.

```
switch(config)# clear ip igmp snooping counters ethernet 15
switch(config)#
```

16.2.8.3 clear ip igmp statistics

The **`clear ip igmp statistics`** command resets IGMP transmission statistic counters for the specified interface.

Command Mode

Privileged EXEC

Command Syntax

`clear ip igmp statistics` [**INTF_ID**]

Parameters

INTF_ID nterface name. Options include:

- ***no parameter*** all interfaces.
- **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **interface loopback *l_num*** Loopback interface specified by ***l_num***.
- **interface management *m_num*** Management interface specified by ***m_num***.
- **interface port-channel *p_num*** Port-channel interface specified by ***p_num***.
- **interface vlan *v_num*** VLAN interface specified by ***v_num***.
- **interface xlan *vx_num*** VXLAN interface specified by ***vx_num***.

Example

This command resets IGMP transmission statistic counters on ***interface tehernet 1***.

```
switch# clear ip igmp statistics interface ethernet 1
switch#
```

16.2.8.4 igmp query-max-response-time

The **`igmp query-max-response-time`** command configures the **query-max-response-time** variable for the configuration mode interface. This variable is used to set the Max Response Time field in outbound

Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

The router with the lowest IP address on a subnet sends membership queries as the IGMP querier. When a membership query is received from a source with a lower IP address, the router resets its query response timer. Upon timer expiry, the router begins sending membership queries. If the router subsequently receives a membership query originating from a lower IP address, it stops sending membership queries and resets the query response timer.

The **no igmp query-max-response-time** and **default igmp query-max-response-time** commands restore the default query-max-response-time of ***10*** seconds for the configuration mode interface by removing the corresponding **igmp query max-response-time** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

igmp query-max-response-time ***period***

no igmp query-max-response-time

default igmp query-max-response-time

Parameters

period maximum response time (deciseconds). Values range from ***1*** to ***31744*** (***52*** minutes, ***54*** seconds). Default is ***100*** (***10*** seconds).

Example

This command configures the query-max-response-time of ***180*** seconds for ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# igmp query-max-response-time 180
switch(config-if-Vl4)#
```

16.2.8.5 ip igmp

The **ip igmp** command enables IGMP on a routed interface or on SVI (VLAN interface) without enabling PIM.

The **no ip igmp** command removes the corresponding ip igmp command from ***running-config***.

Command Mode

Interface Configuration

Command Syntax

ip igmp

no ip igmp

Example

This command enables IGMP on ***interface ethernet 5/2***.

```
switch(config)# interface ethernet 5/2
switch(config-if-Et5/2)# ip igmp
switch(config-if-Et5/2)#
```

16.2.8.6 ip igmp host-proxy

The **ip igmp host-proxy** command enables the IGMP host proxy service on the configuration mode interface. The IGMP host proxy performs IGMP joins and leaves between networks that are directly connected by an exchange that does not support PIM on the network boundary.

Note: For an interface to serve as an IGMP host proxy, PIM MBR must also be enabled on that interface using the **pim ipv4 border-router** command.

The IGMP host proxy sends unsolicited IGMP join reports when an (S,G) or (*,G) entry arrives in the multicast routing (mroute) table. Reports are subsequently sent upon the arrival of queries on the interface. The interval between IGMP reports is configured through **ip igmp host-proxy report-interval**.

The **ip igmp host-proxy** command can also specify a group address; this ensures that reports are generated for the specified group even if there is no (*,G) entry in the mroute table for that group. Multiple **ip igmp host-proxy** statements are required to specify multiple groups.

When the host proxy IGMP version is set to 3 using the **ip igmp host-proxy version** command, the **ip igmp host-proxy** command can also include or exclude source addresses. These options are ignored when the interface runs host proxy IGMP version 1 or 2. Note that the IGMP version set using the **ip igmp version** command does not affect host proxy behavior.

An ACL can also be used in place of a group address by using the **access-list** option. If one or more groups are configured in addition to ACLs, the groups are processed first. Implicit deny in the ACL is ignored, but if the ACL includes an explicit deny rule, then the interface sends joins only to groups configured directly on the interface or included in a permit ACL. Deny rules take precedence over permit rules. If a group is configured with no filters and a host-proxy is configured with an ACL with rules having filters for the group, or configured with groups and source filters, then the filters are applied to the group.

The **no ip igmp host-proxy** and **default ip igmp host-proxy** commands remove the corresponding **ip igmp host-proxy** command from ***running-config***. When these commands do not include a group address, all **ip igmp host-proxy** statements are deleted. When inclusion or exclusion parameters are not specified, all statements with the specified group address are deleted.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip igmp host-proxy [**GROUP_ADDRESS** [**SOURCE_ADDRESS**]]|[access-list ***acl***]

no ip igmp host-proxy [**GROUP_ADDRESS** [**SOURCE_ADDRESS**]]

default ip igmp host-proxy [**GROUP_ADDRESS** [**SOURCE_ADDRESS**]]

Parameters

- **GROUP_ADDRESS** IPv4 address of group address for which host proxy sends reports.
  - ***no parameter*** only groups for which there is a (*,G) entry in the mroute table.
  - ***ipv4_address*** IP address of multicast group (dotted decimal notation). This ensures that reports are generated for this group even if it does not have a (*,G) entry in the mroute table.
- **SOURCE_ADDRESS** IP address of a host that originates multicast data packets.
  - ***no parameter*** Proxy sends report for all received or configured groups regardless of source address.
  - **exclude *ipv4_address*** Proxy does not send reports for specified source address.
  - **include *ipv4_address*** Proxy always sends reports for specified source address.

  Commands that list at least one parameter must specify a group address. Parameters may be listed in any order. When a command specifies include and exclude parameters, the exclude parameter is ignored.

- **access-list acl** specifies an access control list (ACL); a join is sent for all groups and/or sources obtained by processing the rules from all configured ACLs.
- **version *version*** specifies the IGMP version on IGMP host-proxy interface. The value ranges from ***1*** to ***3***. Default value is ***3***.

Guidelines

Multiple statements for a group address may be configured. The effect of entering a command depends on previously entered commands. The following describes command combination:

- **`ip igmp host-proxy`**: IGMP host proxy is enabled for all multicast groups and their source addresses. When enabled for all group addresses, the source address list cannot be restricted.
- **`ip igmp host-proxy`** ***group_ipv4***: IGMP host proxy is enabled for a specified multicast group. The list of source addresses for this group is not restricted. Enabling host proxy for another group address requires another **`ip igmp host-proxy`** command.
- **`ip igmp host-proxy`** ***group_ipv4*** **exclude** ***source_ipv4***: IGMP host proxy is enabled for the specified multicast group. Sources for this group include all addresses not in an exclude statement. Multiple source addresses for the group are excluded by multiple statements.
- **`ip igmp host-proxy`** ***group_ipv4*** **include** ***source_ipv4***: IGMP host proxy is enabled for the specified group address for only the specified source address. Additional statements are required to include other source addresses for the group. The presence of one include parameter invalidates all exclude statements for the specified multicast group.
- **`ip igmp host-proxy access-list`** **acl**: IGMP host proxy is enabled for the addresses defined by the specified ACL. If one or more groups are configured in addition to ACLs, the groups are processed first. If the ACL has a **deny all** rule for a group, then this filter takes precedence over configurations with include /exclude keywords or permit/deny rules for that group. If a group is configured with no filters and a host-proxy is configured with an ACL with rules having filters for the group, or configured with groups and source filters, then the filters are applied to the group.

Examples

- These commands enable IGMP host proxy on ***interface ethernet 17*** for all multicast group addresses.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# pim ipv4 border-router
switch(config-if-Et17)# ip igmp host-proxy
switch(config-if-Et17)#
```

- These commands enable IGMP host proxy on ***interface ethernet 17*** for the multicast group at ***231.10.10.1***. The list of source addresses is not restricted.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# pim ipv4 border-router
switch(config-if-Et17)# ip igmp host-proxy 231.10.10.1
switch(config-if-Et17)#
```

- These commands enable IGMP host proxy on ***interface ethernet 17*** for the multicast group at ***231.10.10.2***. The list of source addresses only excludes ***10.4.4.1*** and ***10.4.5.2***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# pim ipv4 border-router
switch(config-if-Et17)# ip igmp host-proxy 231.10.10.2 exclude 10.4.4.1
switch(config-if-Et17)# ip igmp host-proxy 231.10.10.2 exclude 10.4.5.2
switch(config-if-Et17)#
```

- These commands enable IGMP host proxy on ***interface ethernet 17*** for the multicast group at ***231.10.10.3***. The list of source address for this group only includes ***10.5.5.1*** and ***10.5.5.2***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# pim ipv4 border-router
```

```
switch(config-if-Et17)# ip igmp host-proxy 231.10.10.3 include 10.5.5.1
switch(config-if-Et17)# ip igmp host-proxy 231.10.10.3 include 10.5.5.2
switch(config-if-Et17)#
```

- These commands enable IGMP host proxy on ***interface ethernet 17*** for the group address(es) specified in ACL ***acl1***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# pim ipv4 border-router
switch(config-if-Et17)# ip igmp host-proxy access-list acl1
switch(config-if-Et17)#
```

16.2.8.7 ip igmp host-proxy report-interval

The **`ip igmp host-proxy report-interval`** command configures the period between unsolicited join reports that the switch sends as an IGMP host proxy from the configuration mode interface to a sender network after a (S,G) or (*,G) entry arrives in the multicast route (mroute) table. When the interface receives a query in response, this interval is set to the **ip igmp last-member-query-interval**. This command also enables the host proxy on the configuration mode interface if it was not previously enabled.

The **`no ip igmp host-proxy report-interval`** and **`default ip igmp host-proxy report-interval`** commands reset the query interval to the default value of one second by removing the corresponding **`ip igmp host-proxy report-interval`** command from running-config. The **`no ip igmp host-proxy`** and **`default ip igmp host-proxy`** commands also remove the corresponding **`report-interval`** command.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip igmp host-proxy report-interval` ***period***

`no ip igmp host-proxy report-interval`

`default ip igmp host-proxy report-interval`

Parameters

period transmission interval (seconds) between consecutive reports. Value range: ***1*** (one second) to ***31744*** (8 hours, 49 minutes, 4 seconds). Default is ***1*** (one second).

Example

These commands configures a IGMP host proxy interval of ***5*** seconds on port channel ***100***.

```
switch(config)# interface port-channel 100
switch(config-if-Po100)# ip igmp host-proxy report-interval 5
switch(config-if-Po100)#
```

16.2.8.8 ip igmp host-proxy version

The **`ip igmp host-proxy version`** command configures the version number to be used in unsolicited reports when the interface is serving as an IGMP host proxy. To configure the IGMP version used by the interface for other purposes, use the **ip igmp version** command instead.

The **`no ip igmp host-proxy version`** and **`default ip igmp host-proxy version`** commands reset the version to the default value of ***3***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip igmp host-proxy version` *version_number*

`no ip igmp host-proxy version`

`default ip igmp host-proxy version`

Parameters

version_number values range from ***1*** to ***3***. The default value is ***3***.

Example

These commands configure the IGMP host proxy version on port channel interface ***100*** to ***2***.

```
switch(config)# interface port-channel 100
switch(config-if-Po100)# ip igmp host-proxy version 2
switch(config-if-Po100)#
```

16.2.8.9 ip igmp last-member-query-count

The **`ip igmp last-member-query-count`** command specifies the number of query messages the switch sends in response to a group-specific or group-source-specific leave message.

After receiving a message from a host leaving a group, the switch sends query messages at intervals specified by ip igmp last-member-query-interval. If the switch does not receive a response to the queries after sending the number of messages specified by this parameter, it stops forwarding messages to the host.

Setting the Last Member Query Count (LMQC) to ***1*** causes the loss of a single packet to stop traffic forwarding. While the switch can start forwarding traffic again after receiving a response to the next general query, the host may not receive that query for a period defined by **ip igmp query-interval**.

The **`no ip igmp last-member-query-count`** and **`default ip igmp last-member-query-count`** commands reset the LMQC to the default value by removing the corresponding **`ip igmp last-member-query-count`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip igmp last-member-query-count` *number*

`no ip igmp last-member-query-count`

`default ip igmp last-member-query-count`

Parameters

number query message quantity. Values range from ***0*** to ***3***. Default is ***2***.

Example

This command configures the last-member-query-count to ***3*** on ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp last-member-query-count 3
switch(config-if-Vl4)#
```

16.2.8.10 ip igmp last-member-query-interval

The **`ip igmp last-member-query-interval`** command configures the switch's transmission interval for sending group-specific or group-source-specific query messages from the configuration mode interface.

When a switch receives a message from a host that is leaving a group it sends query messages at intervals set by this command. The **ip igmp startup-query-count** specifies the number of messages that are sent before the switch stops forwarding packets to the host.

If the switch does not receive a response after this period, it stops forwarding traffic to the host on behalf of the group, source, or channel.

The **`no ip igmp last-member-query-interval`** and **`default ip igmp last-member-query-interval`** commands reset the query interval to the default value of one second by removing the **`ip igmp last-member-query-interval`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip igmp last-member-query-interval` ***period***

`no ip igmp last-member-query-interval`

`default ip igmp last-member-query-interval`

Parameter

period transmission interval (deciseconds) between consecutive group-specific query messages. Value range: ***10*** (one second) to ***317440*** (8 hours, 49 minutes, 4 seconds). Default is ***10*** (one second).

Example

This command configures the last member query interval of ***6*** seconds for ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp last-member-query-interval 60
switch(config-if-Vl4)#
```

16.2.8.11 ip igmp profile

The **`ip igmp profile`** command places the switch in the ***IGMP-profile*** configuration mode to configure an IGMP profile. IGMP profiles control the multicast groups that an interface can join.

Profiles consist of the filter type and an address range:

Filter types specify accessibility to the listed address range:

- Permit filters define the multicast groups the interface can join.
- Deny filters define the multicast groups the interface cannot join.

Profiles are deny filters by default.

Address ranges specify a list of addresses and ranges:

- In permit filters, permitted groups are specified by the address range.
- In deny filters, all groups are permitted except those specified by the address range.

Implementing IGMP filtering affects IGMP report forwarding as follows:

- IGMPv2: Report is forwarded to mrouters for permitted groups and dropped for disallowed groups.
- IGMPv3: There may be multiple group records in a report.
  - No groups are allowed: The report is dropped.
  - All groups are allowed: The report is forwarded to mrouter ports as normal.
  - Some groups are allowed: A revised report is forwarded to mrouter ports.

The revised report includes records for the allowed group addresses with the same source MAC and IP addresses.

The **`no ip igmp profile`** and **`default ip igmp profile`** commands delete the specified IGMP profile from ***running-config***.

The ***IGMP-profile*** configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting IGMP-profile configuration mode does not affect the configuration. The **`exit`** command returns the switch to ***global*** configuration mode.

Command Mode

Global Configuration

Command Syntax

`ip igmp profile` ***profile_name***

`no ip igmp profile` ***profile_name***

`default igmp profile` ***profile_name***

Parameters

profile_name name of the IGMP profile.

Commands Available in igmp-profile Configuration Mode

- **permit / deny**
- **range**

Related Commands

ip igmp snooping filter applies an IGMP snooping filter to a configuration mode interface.

Example

These commands enter the ***IGMP-profile*** configuration mode and configure the profile as a permit list.

```
switch(config)# ip igmp profile list_1
switch(config-igmp-profile-list_1)# permit
switch(config-igmp-profile-list_1)#
```

16.2.8.12 ip igmp query-interval

The **`ip igmp query-interval`** command configures the frequency at which the configuration mode interface, as an IGMP querier, sends host-query messages.

An IGMP querier sends host-query messages to discover the multicast groups that have members on networks attached to the interface. The switch implements a default query interval of ***125*** seconds.

The **`no ip igmp query-interval`** and **`default ip igmp query-interval`** commands reset the IGMP query interval to the default value of ***125*** seconds by removing the **`ip igmp query-interval`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip igmp query-interval` ***period***

`no ip igmp query-interval`

`default ip igmp query-interval`

Parameter

period interval (seconds) between IGMP query messages. Values range from ***1*** to ***3175*** (52 minutes, 55 seconds). Default is ***125***.

Example

This command configures the query-interval of ***2*** minutes, ***30*** seconds for ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp query-interval 150
switch(config-if-Vl4)#
```

16.2.8.13 ip igmp router-alert

The **`ip igmp router-alert`** command configures the switch disposition of inbound IGMP packets to the configuration mode interface based on the presence of the router-alert option in the IP header. By default, the port accepts all IGMP packets that arrive on the local subnet and rejects all other packets that arrive without the router-alert option.

The command provides three IGMP packet disposition options:

- **mandatory**: packets are accepted only when router-alert is present.
- **optional**: packets are accepted regardless of router-alert presence.
- **optional connected**: packets are accepted from the same subnet; other packets require router-alert.

The **`no ip igmp router-alert`** and **`default ip igmp router-alert`** commands reset the default setting of optional connected on the configuration mode interface by removing the corresponding **`ip igmp router-alert`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip igmp router-alert` **DISPOSITION**

`no ip igmp router-alert`

`default ip igmp router-alert`

Parameters

DISPOSITION IGMP packet disposition method. Options include:

- **mandatory** Rejects packets if router-alert is not present.
- **optional** Accepts packets regardless of router-alert presence.

- **optional connected** Accepts packets from same subnet. Other packets require router-alert.

Example

This command configures the switch to accept IGMP packets on ***interface ethernet 8*** only if the IP header contains router alert.

```
switch(config)# interface ethernet 8
switch(config-if-Et8)# ip igmp router-alert mandatory
switch(config-if-Et8)#
```

16.2.8.14 ip igmp snooping

The **ip igmp snooping** command enables snooping globally. By default, global snooping is enabled.

When global snooping is enabled, **ip igmp snooping vlan** enables or disables snooping on individual VLANs. When global snooping is disabled, snooping cannot be enabled on individual VLANs.

QoS cannot be used for IGMP packets when IGMP snooping is enabled.

The **no ip igmp snooping** command disables global snooping. The **default ip igmp snooping** command restores the global snooping default setting of enabled by removing the **ip igmp snooping** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping

no ip igmp snooping

default ip igmp snooping

Example

This command globally enables snooping on the switch.

```
switch(config)# ip igmp snooping
switch(config)#
```

16.2.8.15 ip igmp snooping filter

The **ip igmp snooping filter** command applies the specified IGMP snooping profile to the configuration mode interface. An IGMP snooping profile specifies the multicast groups that an interface may join. Profiles consist of the filter type and an address range:

- Filter type: Specifies accessibility to the listed address range:
  - Permit filters define the multicast groups the interface can join.
  - Deny filters define the multicast groups the interface cannot join.
- Address range: Specifies a list of addresses and ranges.
  - In permit filters, the permitted groups are specified by the address range.
  - In deny filters, all groups are permitted except those specified by the address range.

An interface without a snooping profile assignment may join any multicast group.

Snooping profiles are configured in IGMP-profile configuration mode (**ip igmp profile**).

The **no ip igmp snooping filter** and **default ip igmp snooping filter** commands restore the default setting of allowing an interface to join any multicast group by deleting the corresponding **ip igmp snooping filter** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

ip igmp snooping filter ***profile_name***

no ip igmp snooping filter [***profile_name***]

default ip igmp snooping filter [***profile_name***]

Parameters

profile_name name of profile assigned to interface.

Example

This command applies the ***list_1*** snooping profile to ***interface ethernet 7***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# ip igmp snooping filter list_1
switch(config-if-Et7)#
```

16.2.8.16 ip igmp snooping interface-restart-query

The **ip igmp snooping interface-restart-query** command configures the interface startup initial query time used for IGMP query spoofing. When an interface restarts or there is a change to the spanning tree, the interface will send general IGMP queries after this interval. The query is based on the information of the last known IGMP querier, and serves to facilitate faster network convergence times.

Multiple values can be configured with a single command; issuing the command again replaces any previously configured value(s).

The **no ip igmp snooping interface-restart-query** and **default ip igmp snooping interface-restart-query** commands restore the default setting of ***2000*** milliseconds by deleting the corresponding **ip igmp snooping interface-restart-query** command from ***running-config***.

Command Mode

General Configuration

Command Syntax

ip igmp snooping interface-restart-query ***query_time***

no ip igmp snooping interface-restart-query

default ip igmp snooping interface-restart-query

Parameters

query_time interval (in milliseconds) after an interface restart or spanning tree change at which the interface will send general IGMP queries. Values range from ***100*** to ***50000*** milliseconds; default is ***2000***.

Example

This command configures interfaces to send IGMP queries at ***100***, ***200***, and ***300*** milliseconds after an interface restart or spanning tree change.

```
switch(config)# ip igmp snooping interface-restart-query 100 200 300
switch(config)#
```

16.2.8.17 ip igmp snooping proxy

The **ip igmp snooping proxy** command enables snooping proxy globally. By default, IGMP snooping proxy is disabled globally.

When the snooping proxy is enabled globally, it enables IGMP snooping proxy on an individual VLANs and when the IGMP snooping proxy is globally disabled the snooping proxy is disabled on individual VLANs.

The **no ip igmp snooping proxy** and **default ip igmp snooping proxy** commands disable snooping proxy globally and on individual VLANs by removing the **ip igmp snooping proxy** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping proxy

no ip igmp snooping proxy

default ip igmp snooping proxy

Examples

- This command globally enables snooping proxy on the switch.

```
switch(config)# ip igmp snooping proxy
switch(config)#
```

- This command explicitly disables IGMP snooping proxy on ***vlan 20***.

```
switch(config)# no ip igmp snooping vlan 20 proxy
switch(config)#
```

16.2.8.18 ip igmp snooping querier

The **ip igmp snooping querier** command enables the snooping querier globally, which controls the querier for VLANs that are not configured with a snooping querier command. The **ip igmp snooping vlan querier** command controls the querier on individual VLANs.

The IGMP snooping querier supports snooping by sending Layer 2 membership queries to hosts attached to the switch. The snooping querier is functional on VLANs where hosts receive IP multicast traffic without access to a network IP multicast router. A snooping querier avoids flooding multicast packets in the VLAN by querying for hosts and routers.

The IGMP snooping querier is functional on VLANs that meet these criteria:

- Snooping is enabled.
- The corresponding SVI (VLAN interface) is active.
- The VLAN querier IP address or the global querier IP address is configured.

The **no ip igmp snooping querier** and **default ip igmp snooping querier** commands disable the snooping querier globally by removing the **ip igmp snooping querier** statement from ***running-config***. The snooping querier is globally disabled by default.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping querier

no ip igmp snooping querier

default ip igmp snooping querier

Guidelines

- Enabling a querier after it was disabled is equivalent to establishing a new querier.
- Changing the querier IP address is equivalent to establishing a new querier.

Example

This command globally enables the snooping querier on the switch.

```
switch(config)# ip igmp snooping querier
switch(config)#
```

16.2.8.19 ip igmp snooping querier address

The **ip igmp snooping querier address** command sets the global querier source IP address, which specifies the source address for packets transmitted from VLANs for which a querier address (**ip igmp snooping vlan querier address**) is not configured. To use a snooping querier, an address must be explicitly configured globally or for the VLAN.

The switch does not define a default global querier address.

The **no ip igmp snooping querier address** and **default ip igmp snooping querier address** commands remove the global querier address command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping querier address ***ipv4_address***

no ip igmp snooping querier address

default ip igmp snooping querier address

Parameter

ipv4_address source IPv4 address.

Example

This command sets the source IP address to ***10.1.1.41*** for query packets transmitted from the switch.

```
switch(config)# ip igmp snooping querier address 10.1.1.41
switch(config)#
```

16.2.8.20 ip igmp snooping querier last-member-query-count

The **ip igmp snooping querier last-member-query-count** command configures the global **igmp snooping querier last member query count** (LMQC) value. LMQC specifies the number of query messages the switch sends in response to group-specific or group-source-specific leave messages it receives from a host; the transmission frequency is specified by **igmp snooping querier last member query count**. The switch stops forwarding messages to the host if it does not receive a response to these query messages.

Setting LMQC to ***1*** causes the loss of one packet to stop traffic forwarding. While the switch can start forwarding traffic again after receiving a response to the next general query, the host may not receive that query for a period defined by **ip igmp snooping querier query-interval**.

VLANs use the global value when they are not assigned a value (**ip igmp snooping vlan querier last-member-query-count**). VLAN commands take precedence over the global value. The default global value is specified by the robustness variable (**ip igmp snooping robustness-variable**).

The **no igmp snooping querier last-member-query-count** and **default igmp snooping querier last-member-query-count** commands reset the LMQC to the default value by removing the corresponding **ip igmp snooping querier last-member-query-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping querier last-member-query-count ***number***

no ip igmp snooping querier last-member-query-count

default ip igmp snooping querier last-member-query-count

Parameter

number query message quantity. Value ranges from ***1*** to ***3***. Default is set by robustness-variable.

Example

This command configures the global last-member-query-count to ***3***.

```
switch(config)# ip igmp snooping querier last-member-query-count 3
switch(config)# show igmp snooping querier status
   Global IGMP Querier status
-------------------------------------------------------
admin state                     : Disabled
source IP address               : 0.0.0.0
query-interval (sec)            : 125.0
max-response-time (sec)         : 10.0
querier timeout (sec)           : 255.0
last-member-query-interval (sec) : 1.0
last-member-query-count          : 3
startup-query-interval (sec)    : 31.25 (query-interval/4)
startup-query-count             : 2 (robustness)
Vlan Admin    IP              Query    Response Querier Operational Ver
     State                    Interval Time     Timeout State
-------------------------------------------------------------------------
1    Disabled 0.0.0.0         125.0    10.0     255.0   Non-Querier v2
100  Disabled 0.0.0.0         125.0    10.0     255.0   Non-Querier v2
101  Disabled 0.0.0.0         125.0    10.0     255.0   Non-Querier v2
switch(config)#
```

16.2.8.21 ip igmp snooping querier last-member-query-interval

The **ip igmp snooping querier last-member-query-interval** command sets the global IGMP snooping last member query interval. The default interval is ***1*** second.

A multicast host sends an IGMP leave report when it leaves a group. To determine if the host was the last group member, the leave message recipient sends an IGMP query. The **last-member-query-interval** determines when the group record is deleted if no subsequent reports are received.

VLANs not assigned a **last-member-query-interval** value (**ip igmp snooping vlan querier last-member-query-interval**) use the global value. VLAN commands take precedence over the global value.

The **no ip igmp snooping querier last-member-query-interval** and **default ip igmp snooping querier last-member-query-interval** commands reset the **last-member-query-interval**

value the default interval of one second by removing the **`ip igmp snooping querier last-member-query-interval`** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping querier last-member-query-interval` ***period***

`no ip igmp snooping querier last-member-query-interval`

`default ip igmp snooping querier last-member-query-interval`

Parameter

period last member query interval (seconds). Value ranges from ***1*** to ***25***. Default is ***1*** second.

Related Commands

ip igmp snooping vlan querier last-member-query-interval assign a last member query interval value to the specified VLANs.

Example

This command sets the IGMP snooping querier last-member-query-interval to ***5*** seconds.

```
switch(config)# ip igmp snooping querier last-member-query-interval 5
switch(config)#
```

16.2.8.22 ip igmp snooping querier max-response-time

The **`ip igmp snooping querier max-response-time`** command specifies the global **max-response-time** value. The switch uses **max-response-time** to set the Max Response Time field in outbound Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

VLANs not assigned a **max-response-time** value (**ip igmp snooping vlan querier max-response-time**) use the global value. VLAN commands take precedence over the global value.

Values range from ***1*** to ***25*** seconds. The default global value is ***10*** seconds.

The **`no ip igmp snooping querier max-response-time`** and **`default ip igmp snooping querier max-response-time`** commands restore the global **max-response-time** default value by removing the **`ip igmp snooping querier max-response-time`** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping querier max-response-time` ***resp_sec***

`no ip igmp snooping querier max-response-time`

`default ip igmp snooping querier max-response-time`

Parameters

resp_sec **max-response-time** value (seconds). Values range from ***1*** to ***25***. Default (global) is ***10***.

Example

This command sets the global max-response-time to ***15*** seconds.

```
switch(config)# ip igmp snooping querier max-response-time 15
```

```
switch(config)#
```

16.2.8.23 ip igmp snooping querier query-interval

The **ip igmp snooping querier query-interval** command sets the global query interval. This command also sets the query-interval of IGMP Snooping when using IGMP version 2. Values range from ***5*** to ***3600*** seconds. The default global value is ***125*** seconds. The query interval is the period between IGMP Membership Query messages sent from the querier. The global value specifies the query interval for VLANs with no query-interval command.

VLANs not assigned a query interval value (**ip igmp snooping vlan querier query-interval**) use the global value. VLAN commands take precedence over the global value.

The **no ip igmp snooping querier query-interval** and **default ip igmp snooping querier query-interval** commands reset the global query-interval value to ***125*** seconds by removing the **ip igmp snooping querier query-interval** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping querier query-interval ***query_sec***

no ip igmp snooping querier query-interval

default ip igmp snooping querier query-interval

Parameter

query_sec query interval (seconds). Values range from ***5*** to ***3600***. Default (global) is ***125***.

Example

This command sets the global query interval to ***150*** seconds.

```
switch(config)# ip igmp snooping querier query-interval 150
switch(config)#
```

16.2.8.24 ip igmp snooping querier startup-query-count

The **ip igmp snooping querier startup-query-count** command configures the global **startup query count** value. The **startup query count** specifies the number of query messages that the querier sends on a VLAN during the **startup query interval** (**ip igmp snooping querier startup-query-interval**).

When snooping is enabled, the group state is more quickly established by sending query messages at a higher frequency. The **startup-query-interval** and **startup-query-count** parameters define the startup period by defining the number of queries to be sent and transmission frequency for these messages.

VLANs use the global **startup query count** value when they are not assigned a value (**ip igmp snooping vlan querier startup-query-count**). VLAN commands take precedence over the global value. The default global value is specified by the robustness variable (**ip igmp snooping robustness-variable**).

The **no ip igmp snooping querier startup-query-count** and **default ip igmp snooping querier startup-query-count** commands restore the default startup-query-count value by removing the corresponding **ip igmp snooping querier startup-query-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping querier startup-query-count ***number***

no ip igmp snooping querier startup-query-count

default ip igmp snooping querier startup-query-count

Parameter

number global startup query count. Value ranges from ***1*** to ***3***.

Example

These commands configure the global startup query count value of ***2***, then displays the status of the snooping querier.

```
switch(config)# ip igmp snooping querier startup-query-count 2
switch(config)# show igmp snooping querier status
   Global IGMP Querier status
-----------------------------------------------------
admin state                       : Disabled
source IP address                 : 0.0.0.0
query-interval (sec)              : 125.0
max-response-time (sec)           : 10.0
querier timeout (sec)             : 255.0
last-member-query-interval (sec)  : 1.0
last-member-query-count           : 2 (robustness)
startup-query-interval (sec)      : 31.25 (query-interval/4)
startup-query-count               : 2

Vlan Admin    IP              Query    Response Querier Operational Ver
     State                    Interval Time     Timeout State
-----------------------------------------------------------------------
1    Disabled 0.0.0.0         125.0    10.0     255.0   Non-Querier v2
100  Disabled 0.0.0.0         125.0    10.0     255.0   Non-Querier v2
101  Disabled 0.0.0.0         125.0    10.0     255.0   Non-Querier v2
switch(config)#
```

16.2.8.25 ip igmp snooping querier startup-query-interval

The **ip igmp snooping querier startup-query-interval** command configures the global startup query interval value. The **startup query interval** specifies the period between query messages that the querier sends upon startup.

When snooping is enabled, the group state is more quickly established by sending query messages at a higher frequency. The **startup-query-interval** and **startup-query-count** parameters define the startup period by defining the number of queries to be sent and transmission frequency for these messages.

VLANs use the global **startup query interval** value when they are not assigned a value (**ip igmp snooping vlan querier startup-query-interval**). VLAN commands take precedence over the global value. The default global value equals the query interval divded by four. (**ip igmp snooping querier query-interval**).

The **no ip igmp snooping querier startup-query-interval** and **default ip igmp snooping querier startup-query-interval** commands restore the default method of specifying the startup query interval by removing the corresponding **ip igmp snooping querier startup-query-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping querier startup-query-interval ***period***

no ip igmp snooping querier startup-query-count

default ip igmp snooping querier startup-query-count

Parameter

period startup query interval (seconds). Value ranges from ***1*** to ***3600*** (***1*** hour).

Example

This command configures the startup query count of one minute for ***interface vlan 4***.

```
switch(config)# ip igmp snooping querier startup-query-interval 40
switch(config)# show igmp snooping querier status
   Global IGMP Querier status
-------------------------------------------------------
admin state                       : Enabled
source IP address                 : 0.0.0.0
query-interval (sec)              : 125.0
max-response-time (sec)           : 10.0
querier timeout (sec)             : 255.0
last-member-query-interval (sec)  : 1.0
last-member-query-count           : 2 (robustness)
startup-query-interval (sec)      : 40.0
startup-query-count               : 2

Vlan Admin    IP              Query    Response Querier Operational Ver
     State                    Interval Time     Timeout State
-------------------------------------------------------------------------
1    Enabled  0.0.0.0         125.0    10.0     255.0   Non-Querier v3
100  Enabled  0.0.0.0         125.0    10.0     255.0   Non-Querier v3
101  Enabled  0.0.0.0         125.0    10.0     255.0   Non-Querier v3
switch(config)#
```

16.2.8.26 ip igmp snooping querier version

The **ip igmp snooping querier version** command configures the Internet Group Management Protocol (IGMP) snooping querier version on the configuration mode interfaces. Version 3 is the default IGMP version.

IGMP is enabled by the **pim ipv4 sparse-mode** or **pim ipv4 bidirectional** command. The **ig igmp snooping querier version** command does not affect the IGMP enabled status.

The **no ip igmp snooping querier version** and **default ip igmp snooping querier version** commands restore the configuration mode to IGMP version 3 by removing the **ip igmp snooping querier version** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping querier version ***version_number***

no ip igmp snooping querier startup-query-count

default ip igmp snooping querier startup-query-count

Parameter

version_number IGMP version number. Value ranges from ***1*** to ***3***. Default value is ***3***.

Examples

- This command configures IGMP snooping querier version 2.

```
switch(config)# ip igmp snooping querier version 2
```

```
switch(config)#
```

This command restores the IGMP snooping querier to version 3.

```
switch(config)# no ip igmp snooping querier version
switch(config)#
```

16.2.8.27 ip igmp snooping report-flooding

The **ip igmp snooping report-flooding** command globally enables L2 report flooding on the switch. When report flooding is globally enabled, the **ip igmp snooping vlan report-flooding** configures a VLAN range to forward membership report messages to specified ports. When report flooding is not globally enabled, L2 report flooding cannot be enabled on individual VLANs.

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. Relying on a single switch to maintain and send report messages can result in performance issues. L2 report flooding addresses this by facilitating report message transmissions through any network port. This allows switches to bypass the querier when forwarding multicast traffic to its interested ports.

The **no ip igmp snooping report-flooding** and **default ip igmp snooping report-flooding** commands disable global L2 report flooding by removing **ip igmp report flooding** from ***running-config***. L2 report flooding is disabled by default.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping report-flooding

no ip igmp snooping report-flooding

default ip igmp snooping report-flooding

Related Command

ip igmp snooping vlan report-flooding enables L2 report flooding on a specified VLAN range.

Example

This command globally enables the snooping L2 report-flooding.

```
switch(config)# ip igmp snooping report-flooding
switch(config)#
```

16.2.8.28 ip igmp snooping report-flooding switch-port

The **ip igmp snooping report-flooding switch-port** command specifies Ethernet ports or port channels that can forward IGMP membership report messages for all VLANs where L2 report flooding is enabled. Ports that are connected to multicast routers or queriers continue to forward traffic as previously specified and are not affected by L2 report flooding commands.

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. The **ip igmp snooping vlan report-flooding switch-port** command configures a list of forwarding ports for a specified VLAN range.

The **no ip igmp snooping report-flooding switch-port** and **default ip igmp snooping report-flooding switch-port** commands remove the specified ports from the global report flooding port list by deleting the corresponding **ip igmp snooping report-flooding switch-port** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping report-flooding switch-port` **INTERFACE**

`no ip igmp snooping report-flooding switch-port` **INTERFACE**

`default ip igmp snooping report-flooding switch-port` **INTERFACE**

Parameters

INTERFACE Membership report message forwarding is enabled on these ports:

- **ethernet *e_range*** where ***e_range*** is the number, range, or list of ethernet ports.
- **port-channel *p_range*** where ***p_range*** is the number, range, or list of channel ports.

Related Commands

- **ip igmp snooping report-flooding** globally enables L2 report flooding.
- **ip igmp snooping vlan report-flooding switch-port** specifies a port list for a VLAN range.

Example

This command configures Ethernet ports ***7-9*** for report message forwarding for any VLAN where L2 report flooding is enabled.

```
switch(config)# ip igmp snooping report-flooding switch-port ethernet 7-9
switch(config)#
```

16.2.8.29 ip igmp snooping restart query-interval

The **`ip igmp snooping restart query-interval`** command sets the query interval for all VLANs during an IGMP snooping restart. By default, the query interval during an IGMP snooping restart is a VLAN configured query interval divided by five. This accelerates the transmission of robustness queries to establish the IGMP snooping state more quickly. However, some large scale configurations may not be able to process all of the queries at this query interval rate. The restart query interval, when configured, is valid for all VLANs.

The **`no ip igmp snooping resrtart query-interval`** and **`default ip igmp snooping restart query-interval`** commands removes the global restart query interval by deleting the **ip igmp snooping restart query-interval** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping restart query-interval` ***query_sec***

`no ip igmp snooping restart query-interval`

`default ip igmp snooping restart query-interval`

Parameter

query_sec query interval (seconds). Values range from ***2*** to ***400***. Default (global) is ***125***.

Example

This command sets the global query interval to ***35*** seconds.

```
switch(config)# ip igmp snooping restart query-interval 35
switch(config)#
```

16.2.8.30 ip igmp snooping robustness-variable

The **ip igmp snooping robustness-variable** command configures the robustness variable for snooping packets sent from any VLAN.

The robustness variable specifies the number of unacknowledged snooping queries that a switch sends before removing the recipient from the group list.

The **no ip igmp snooping robustness-variable** and **default ip igmp snooping robustness-variable** commands reset the robustness variable to 2 by removing the **ip igmp snooping robustness-variable** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping robustness-variable ***robust_value***

no ip igmp snooping robustness-variable

default ip igmp snooping robustness-variable

Parameter

robust_value robustness variable. Values range from ***1*** to ***3***. Default is ***2***.

Example

This command sets the robustness-variable value to ***3***.

```
switch(config)# ip igmp snooping robustness-variable 3
switch(config)#
```

16.2.8.31 ip igmp snooping vlan

The **ip igmp snooping vlan** command enables snooping on the specified VLANs if snooping is globally enabled. IGMP snooping is globally enabled by default. The **ip igmp snooping** command enables snooping globally.

Note that if IGMP snooping is enabled, QoS will not apply to IGMP packets.

The **no ip igmp snooping vlan** command disables snooping on the specified VLANs.

The **default ip igmp snooping vlan** command returns the snooping setting for the specified VLANs to enabled by removing the corresponding **ip igmp snooping vlan** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan ***v_range***

no ip igmp snooping vlan ***v_range***

default ip igmp snooping vlan ***v_range***

Parameter

v_range VLANs upon which snooping is enabled. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.

Example

This command disables snooping on ***vlan 2***, ***vlan 3***, and ***vlan 4***.

```
switch(config)# no ip igmp snooping vlan 2-4
switch(config)#
```

16.2.8.32 ip igmp snooping vlan fast-leave

The **ip igmp snooping vlan fast-leave** command enables fast-leave processing on specified VLANs. When fast-leave processing is enabled, the removal of a VLAN interface's multicast group entry from the IGMP table is not preceded by an IGMP group-specific query to the interface. The switch removes an interface from the forwarding table when it detects an IGMP leave message on the interface. IGMP fast-leave processing is enabled on all VLANs by default.

The **no ip igmp snooping vlan fast-leave** command disables fast-leave processing on the specified VLANs. The **default ip igmp snooping vlan fast-leave** command restores fast-leave processing on the specified VLANs by removing the corresponding **no ip igmp snooping vlan fast-leave** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan ***v_range*** **fast-leave**

no ip igmp snooping vlan ***v_range*** **fast-leave**

default ip igmp snooping vlan ***v_range*** **fast-leave**

Parameter

v_range VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.

Example

This command enables IGMP fast-leave processing on ***vlan 10***.

```
switch(config)# ip igmp snooping vlan 10 fast-leave
switch(config)#
```

16.2.8.33 ip igmp snooping vlan max-groups

The **ip igmp snooping vlan max-groups** command configures the quantity of multicast groups that the specified VLAN forwarding table can contain. After the limit is reached, attempts to join new groups are ignored. There is no default limit.

The **no ip igmp snooping vlan max-groups** and **default ip igmp snooping vlan max-groups** removes the maximum group limit by deleting the **ip igmp snooping vlan max-groups** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan ***v_range*** **max-groups** ***quantity***

no ip igmp snooping vlan ***v_range*** **max-groups**

default ip igmp snooping vlan ***v_range*** **max-groups**

Parameters

- ***v_range*** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***quantity*** maximum number of groups that can access the VLAN. Value ranges from ***0*** to ***65534***.

Examples

- This command limits the number of multicast groups that hosts on ***vlan 6*** can simultaneously access to ***25***.

```
switch(config)# ip igmp snooping vlan 6 max-groups 25
switch(config)#
```

- This command allows each VLAN between ***8*** and ***15*** to receive multicast packets from ***30*** groups.

```
switch(config)# ip igmp snooping vlan 8-15 max-groups 30
switch(config)#
```

- This command removes the maximum group restriction from all VLAN interfaces between ***1*** and ***50***.

```
switch(config)# no ip igmp snooping vlan 1-50 max-groups
switch(config)#
```

16.2.8.34 ip igmp snooping vlan member

The **`ip igmp snooping vlan member`** command adds ports as static members to a multicast group. The ports must be in the specified VLAN range.

The **`no ip igmp snooping vlan member`** and **`default ip igmp snooping vlan member`** commands remove the specified ports from the multicast group by deleting the corresponding ip igmp snooping member statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping vlan` ***v_num*** **`member`** ***ipv4_addr*** interface **STATIC_INT**

`no ip igmp snooping vlan` ***v_num*** **`member`** ***ipv4_addr*** interface **STATIC_INT**

`default ip igmp snooping vlan` ***v_num*** **`member`** ***ipv4_addr*** interface **STATIC_INT**

Parameters

- ***v_num*** VLAN number. Value ranges from ***1*** to ***4094***.
- ***ipv4_addr*** multicast group IPv4 address.
- **STATIC_INT** interface the command configures as the static group member. Options include:
  - **ethernet *e_range***, where ***e_range*** is the number, range, or list of Ethernet ports.
  - **port-channel *p_range***, where ***p_range*** is the number, range, or list of channel ports.

Example

This command configures the static connection to a multicast group at ***237.2.1.4*** through ***interface ethernet 3***.

```
switch(config)# ip igmp snooping vlan 7 member 237.2.1.4 interface ethernet 3
switch(config)#
```

16.2.8.35 ip igmp snooping vlan multicast-router

The **ip igmp snooping vlan multicast-router** command adds a multicast router as a static port to the specified VLANs. The router port must be in the specified VLAN range.

Snooping may not always be able to locate the IGMP querier. This command should specify IGMP queriers that are known to connect to the network through a port on the switch.

The **no ip igmp snooping vlan multicast-router** and **default ip igmp snooping vlan multicast-router** commands remove the specified static port configuration by deleting the corresponding **ip igmp snooping vlan multicast-router** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan ***v_range*** **multicast-router interface STATIC_INT**

no ip igmp snooping vlan ***v_range*** **multicast-router interface STATIC_INT**

default ip igmp snooping vlan ***v_range*** **multicast-router interface STATIC_INT**

Parameters

- ***v_range*** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- **STATIC_INT** interface the command configures as a static port. Selection options include:
  - **ethernet *e_range*** where ***e_range*** is the number, range, or list of ethernet ports.
  - **port-channel *p_range*** where ***p_range*** is the number, range, or list of channel ports.
- The **STATIC_INT** interface must route traffic through a VLAN specified within ***v_range***.

Example

This command configures the static connection to a multicast router through ***interface ethernet 3***.

```
switch(config)# ip igmp snooping vlan 2 multicast-router interface ethernet 3
switch(config)#
```

16.2.8.36 ip igmp snooping vlan proxy

The **ip igmp snooping vlan proxy** command enables snooping proxy on individual VLAN. To enable or disable IGMP snooping vlan proxy globally, use the **ip igmp snooping proxy** command.

Note: The **ip igmp snooping proxy** command enables snooping proxy on all VLANs only where IGMP snooping is enabled.

The **no** and **default** form of **ip igmp snooping vlan proxy** command disables snooping proxy globally and on individual VLANs by removing the **ip igmp snooping vlan proxy** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan [***ID*** | ***range***]**proxy**

no ip igmp snooping vlan [***ID*** | ***range***] **proxy**

default ip igmp snooping vlan [***ID*** | ***range***] **proxy**

Parameters

- ***ID*** specifies a individual VLAN ID. Numbers range from ***1*** to ***4094***.
- ***range*** specifies the range of VLAN IDs. Numbers range from ***1*** to ***4094***.

Examples

- This command globally enables IGMP snooping proxy on the switch and on all VLANs where IGMP snooping is enabled.

```
switch(config)# ip igmp snooping proxy
switch(config)#
```

- This command enables IGMP snooping proxy on ***vlan 20***.

```
switch(config)# ip igmp snooping vlan 20 proxy
switch(config)#
```

- This command disables IGMP snooping proxy on ***vlan 20***.

```
switch(config)# no ip igmp snooping vlan 20 proxy
switch(config)#
```

16.2.8.37 ip igmp snooping vlan querier

The **`ip igmp snooping vlan querier`** command controls the querier for the specified VLANs. VLANs follow the global querier setting unless overridden by one of these commands:

- **`ip igmp snooping vlan querier`** enables the querier on specified VLANs.
- **`no ip igmp snooping vlan querier`** disables the querier on specified VLANs.

VLAN querier commands take precedence over the global querier setting. The **ip igmp snooping querier** controls the querier for VLANs with no snooping querier command.

The IGMP snooping querier supports snooping by sending Layer 2 membership queries to hosts attached to the switch. The snooping querier is functional on VLANs where hosts receive IP multicast traffic without access to a network IP multicast router. A snooping querier avoids flooding multicast packets in the VLAN by querying for hosts and routers.

The IGMP snooping querier is functional on VLANs that meet these criteria:

- Snooping is enabled.
- The corresponding SVI (VLAN interface) is active.
- The VLAN querier IP address or the global querier IP address is configured.

The **`default ip igmp snooping vlan querier`** command restores the usage of the global setting for the specified VLAN by removing the corresponding **`ip igmp snooping vlan querier`** or **`no ip igmp snooping vlan querier`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping vlan` ***v_range*** **`querier`**

`no ip igmp snooping vlan` ***v_range*** **`querier`**

`default ip igmp snooping vlan` ***v_range*** **`querier`**

Parameters

v_range VLAN IDs. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.

Examples

- These commands globally enable the snooping querier on the switch, explicitly disable snooping on VLANs ***1-3***, and explicitly enable snooping on VLANs ***4-6***.

```
switch(config)# ip igmp snooping querier
switch(config)# no ip igmp snooping vlan 1-3 querier
switch(config)# ip igmp snooping vlan 4-6 querier
```

After running these commands, the ***running-config*** file contains these lines, which indicate that the snooping querier is enabled on VLANs ***4-6***.

```
switch(config)# show running-config

                <-------OUTPUT OMITTED FROM EXAMPLE-------->
no ip igmp snooping vlan 1 querier
no ip igmp snooping vlan 2 querier
no ip igmp snooping vlan 3 querier
ip igmp snooping vlan 4 querier
ip igmp snooping vlan 5 querier
ip igmp snooping vlan 6 querier
ip igmp snooping querier
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command removes the querier setting for VLANs ***2-5***:

```
switch(config)# default ip igmp snooping vlan 2-5 querier
```

When executed after the previous commands, the snooping querier is disabled explicitly on VLANs ***1-2***, enabled implicitly on VLANs ***3-6***, and enabled explicitly on VLANs ***7-8***, as shown by ***running-config***:

```
<-------OUTPUT OMITTED FROM EXAMPLE-------->
no ip igmp snooping vlan 1 querier
ip igmp snooping vlan 6 querier
ip igmp snooping querier
                <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command sets the global snooping querier to disabled by removing the global querier setting from ***running-config***:

```
switch(config)# no ip igmp snooping querier
switch(config)#
```

When executed after the previous commands, the snooping querier is disabled explicitly on VLANs ***1-2***, disabled implicitly on VLANs ***3-6*** and enabled explicitly on VLANs ***7-8***, as shown by ***running-config***.

```
<-------OUTPUT OMITTED FROM EXAMPLE-------->
no ip igmp snooping vlan 1 querier
ip igmp snooping vlan 6 querier
                <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

16.2.8.38 ip igmp snooping vlan querier address

The **`ip igmp snooping vlan querier address`** command sets the source address for query packets sent from specified VLANs. VLANs not assigned an address use the global address (**ip igmp snooping querier address**). VLAN querier address commands take precedence over the global address.

To use a snooping querier, an address must be explicitly configured globally or for the querier VLAN.

The **`no ip igmp vlan snooping querier address`** and **`default ip igmp snooping vlan querier address`** commands reset the specified VLAN to use the global address by removing the corresponding **`ip igmp snooping vlan querier address`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping vlan` ***`v_range`*** **`querier address`** ***`ipv4_address`***

`no ip igmp snooping vlan` ***`v_range`*** **`querier address`**

`default ip igmp snooping vlan` ***`v_range`*** **`querier address`**

Parameters

- ***v_range*** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***ipv4_address*** source IPv4 address.

Example

This command sets the source IPv4 address of ***10.14.1.1***. for query packets transmitted from ***vlan 2***.

```
switch(config)# ip igmp snooping vlan 2 querier address 10.14.1.1
switch(config)#
```

16.2.8.39 ip igmp snooping vlan querier last-member-query-count

The **`ip igmp snooping vlan querier last-member-query-count`** command specifies an **`IGMP snooping querier last-member-query-count`** (LMQC) value for the specified VLANs. LMQC specifies the number of query messages the switch sends in response to group-specific or group-source-specific leave messages it receives from a host; the transmission frequency is specified by **`IGMP snooping querier last member query interval`**. The switch stops forwarding messages to the host if it does not receive a response to these query messages.

VLANs not assigned an LMQC value use the global value (**ip igmp snooping querier last-member-query-count**). VLAN commands take precedence over the global command.

Setting the last member query count (LMQC) to ***1*** causes the loss of a single packet to stop traffic forwarding. While the switch can start forwarding traffic again after receiving a response to the next general query, the host may not receive that query for a period defined by **ip igmp snooping querier query-interval l**.

The **`no igmp snooping vlan querier last-member-query-count`** and **`default igmp snooping vlan querier last-member-query-count`** commands reset the specified VLAN to use the global LMQC by removing the corresponding **`ip igmp snooping vlan querier last-member-query-count`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping vlan` ***`v_range`*** **`querier last-member-query-count`** ***`number`***

`no ip igmp snooping vlan` ***`v_range`*** **`querier address`**

`default ip igmp snooping vlan` ***`v_range`*** **`querier address`**

Parameters

- ***v_range*** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***number*** query message quantity. Value ranges from ***1*** to ***3***.

Example

This command configures the last-member-query-count to ***1*** on ***vlan 3***.

```
switch(config)# ip igmp snooping vlan 3 querier last-member-query-count 1
switch(config)#
```

16.2.8.40 ip igmp snooping vlan querier last-member-query-interval

The **ip igmp snooping vlan querier last-member-query-interval** command configures **last-member-query-interval** for packets sent from the specified VLANs. VLANs not assigned a value use the global setting (**ip igmp snooping querier last-member-query-interval**). VLAN commands take precedence over the global value. The global default is one second.

A multicast host sends an IGMP leave report when it leaves a group. To determine if the host was the last group member, the leave message recipient sends an IGMP query. The last-member-query-interval determines when the group record is deleted if no subsequent reports are received.

The **no ip igmp snooping vlan querier last-member-query-interval** and **default ip igmp snooping vlan querier last-member-query-interval** commands reset the specified VLAN to use the global **last-member-query-interval** by removing the corresponding **ip igmp snooping vlan querier last-member-query-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan ***v_range*** **querier last-member-query-interval** ***period***

no ip igmp snooping vlan ***v_range*** **querier last-member-query-interval**

default ip igmp snooping vlan ***v_range*** **querier last-member-query-interval**

Parameters

- ***v_range*** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***period*** last member query interval (seconds). Value ranges from ***1*** to ***25***.

Example

This command sets the last-member-query-interval for ***vlan 10*** to ***12*** seconds.

```
switch(config)# ip igmp snooping vlan 10 querier last-member-query-interval 12
switch(config)#
```

16.2.8.41 ip igmp snooping vlan querier max-response-time

The **ip igmp snooping vlan querier max-response-time** command configures **max-response-time** for packets sent from the specified VLANs. VLANs not assigned a value use the global setting (**ip igmp snooping querier max-response-time**). VLAN commands take precedence over the global value. The global default is ***10*** seconds.

Switches use **max-response-time** to set the Max Response Time field in outbound Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

The **no ip igmp snooping vlan querier max-response-time** and **default ip igmp snooping vlan querier max-response-time** commands reset the specified VLAN to use the global **max-response-time** by removing the corresponding **ip igmp snooping vlan querier max-response-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping vlan` ***v_range*** **`querier max-response-time`** ***resp_sec***

`no ip igmp snooping vlan` ***v_range*** **`querier max-response-time`**

`default ip igmp snooping vlan` ***v_range*** **`querier max-response-time`**

Parameters

- ***v_range*** VLAN ID. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***resp_sec*** **max-response-time** value (seconds). Values range from ***1*** to ***25***.

Example

This command sets the max-response-time for ***vlan 2*** to ***5*** seconds.

```
switch(config)# ip igmp snooping vlan 2 querier max-response-time 5
switch(config)#
```

16.2.8.42 ip igmp snooping vlan querier query-interval

The **`ip igmp snooping vlan querier query-interval`** command sets the query interval for the specified VLAN. VLANs not assigned a value use the global value (**ip igmp snooping querier query-interval**). VLAN commands have precedence over the global value. The query interval is the period between IGMP Membership Query messages sent from the querier.

The **`no ip igmp snooping vlan querier query-interval`** and **`default ip igmp snooping vlan querier query-interval`** commands reset the specified VLAN to use the global value by removing the corresponding **`ip igmp snooping vlan querier query-interval`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping vlan` ***v_range*** **`querier query-interval`** ***query_sec***

`no ip igmp snooping vlan` ***v_range*** **`querier query-interval`**

`default ip igmp snooping vlan` ***v_range*** **`querier query-interval`**

Parameters

- ***v_range*** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***query_sec*** query interval (seconds). Values range from ***5*** to ***3600***. Default (global) is ***125***.

Example

This command sets the query interval for ***vlan 10*** to ***240*** seconds.

```
switch(config)# ip igmp snooping vlan 10 querier query-interval 240
switch(config)#
```

16.2.8.43 ip igmp snooping vlan querier startup-query-count

The **`ip igmp snooping vlan querier startup-query-count`** command specifies the startup query count value for the specified VLANs. The **startup query count** specifies the number of query messages that

the querier sends on a VLAN during the **startup query interval** (**ip igmp snooping vlan querier startup-query-interval**).

When an interface starts running IGMP, it can establish the group state more quickly by sending query messages at a higher frequency. The **startup-query-interval** and **startup-query-count** parameters define the startup period and the query message transmission frequency during that period.

VLANs not assigned a **startup query count** value use the global value (**ip igmp snooping querier startup-query-count**). VLAN commands take precedence over the global command.

The **`no ip igmp snooping vlan querier startup-query-count`** and **`default ip igmp snooping vlan querier startup-query-count`** commands restore the default condition of using the global **startup query count** value by removing the corresponding **`ip igmp snooping vlan querier startup-query-count`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping vlan` ***v_range*** **`querier startup-query-count`** ***number***

`no ip igmp snooping vlan` ***v_range*** **`querier startup-query-count`**

`default ip igmp snooping vlan` ***v_range*** **`querier startup-query-count`**

Parameters

- ***v_range*** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***number*** startup query count. Value ranges from ***1*** to ***3***.

Example

This command configures the startup query count of ***3*** for ***vlan 100***.

```
switch(config)# ip igmp snooping vlan 100 querier startup-query-count 3
switch(config)#
```

16.2.8.44 ip igmp snooping vlan querier startup-query-interval

The **`ip igmp snooping vlan querier startup-query-interval`** command specifies the **startup query interval** value for the specified VLANs. The **startup query interval** specifies the period between query messages that the querier sends upon startup.

When snooping is enabled, the group state is more quickly established by sending query messages at a higher frequency. The **startup-query-interval** and **startup-query-count** parameters define the startup period by defining the number of queries to be sent and transmission frequency for these messages.

VLANs not assigned a **startup query interval** value use the global value (**ip igmp snooping querier startup-query-count**). VLAN commands take precedence over the global command.

The **`no ip igmp snooping vlan querier startup-query-interval`** and **`default ip igmp snooping vlan querier startup-query-interval`** commands restore the default condition of using the global **startup query interval** value by removing the corresponding **`ip igmp snooping vlan querier startup-query-interval`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`ip igmp snooping vlan` ***v_range*** **`querier startup-query-interval`** ***period***

no ip igmp snooping vlan *v_range* querier startup-query-interval

default ip igmp snooping vlan *v_range* querier startup-query-interval

Parameters

- ***v_range*** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***period*** startup query interval (seconds). Value ranges from ***1*** to ***3600*** (1 hour). Default is ***31***.

Example

This command configures the startup query count of ***60*** seconds for ***vlan 100***.

```
switch(config)# ip igmp snooping vlan 100 querier startup-query-interval 60
switch(config)#
```

16.2.8.45 ip igmp snooping vlan querier version

The **ip igmp snooping vlan querier version** command configures the Internet Group Management Protocol (IGMP) snooping querier function on the VLAN. Version 3 is the default IGMP snooping version.

IGMP is enabled by the **pim ipv4 sparse-mode** or **pim ipv4 bidirectional** command. The **ig igmp snooping vlan querier version** command does not affect the IGMP enabled status.

The **no ip igmp snooping vlan querier version** and **default ip igmp snooping vlan querier version** commands restore the configuration mode interface to IGMP snooping VLAN querier version 3 by removing the **ip igmp snooping vlan querier version** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan *v_range* querier version *version_number*

no ip igmp snooping vlan *v_range* querier version

default ip igmp snooping vlan *v_range* querier version

Parameters

- ***v_range*** VLAN ID. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- ***version_number*** IGMP version number. Value ranges from ***1*** to ***3***. Default value is ***3***.

Example

- The example sets the querier to ***version 2*** on ***vlan 5***.

```
switch(config)# ip igmp snooping vlan 5 querier version 2
switch(config)#
```

- This command restores IGMP snooping querier ***version 3*** to ***vlan 5***.

```
switch(config)# no ip igmp snooping vlan 5 querier version
switch(config)#
```

16.2.8.46 ip igmp snooping vlan report-flooding

The **ip igmp snooping vlan report-flooding** command enables L2 report flooding on the specified VLANs if report flooding is globally enabled. When L2 report flooding is not globally enabled, this command has no effect. The **ip igmp snooping report-flooding** command globally enables L2 report flooding.

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. Relying on a single switch to maintain and send report messages can degrade performance. L2 report flooding addresses this by facilitating report message forwarding through any network port. This allows switches to bypass the querier when forwarding multicast traffic to its interested ports.

Two commands specify the ports that forward reports:

- **ip igmp snooping vlan report-flooding switch-port** for a VLAN range.
- **ip igmp snooping report-flooding switch-port** for all VLANs where report flooding is enabled.

The **no ip igmp snooping vlan report-flooding** and **default ip igmp snooping vlan report-flooding** commands disable L2 report flooding for the specified VLAN by removing the corresponding **ip igmp snooping vlan report-flooding** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan *v_range* report-flooding

no ip igmp snooping vlan *v_range* report-flooding

default ip igmp snooping vlan *v_range* report-flooding

Parameters

v_range VLAN IDs Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.

Related Commands

ip igmp snooping report-flooding globally enables L2 report flooding.

Example

These commands enable L2 report flooding globally and on VLANs ***201*** through ***205***.

```
switch(config)# ip igmp snooping report-flooding
switch(config)# ip igmp snooping vlan 201-205 report-flooding
switch(config)#
```

16.2.8.47 ip igmp snooping vlan report-flooding switch-port

The **ip igmp snooping vlan report-flooding switch-port** command configures Ethernet ports or port channels to forward IGMP membership report messages for a specified VLAN range where L2 report flooding is enabled. Ports that are connected to multicast routers or queriers continue to forward traffic as previously specified and are not affected by L2 report flooding commands.

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. The **ip igmp snooping report-flooding switch-port** command configures a list of forwarding ports for all VLANs where L2 report flooding is enabled.

The **no ip igmp snooping vlan report-flooding switch-port** and **default ip igmp snooping vlan report-flooding switch-port** commands remove the listed ports from the specified report flooding port list by deleting the corresponding **ip igmp snooping vlan report-flooding switch-port** statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip igmp snooping vlan *v_range* report-flooding switch-port INTERFACE

`no ip igmp snooping vlan` *`v_range`* `report-flooding switch-port` INTERFACE

`default ip igmp snooping vlan` *`v_range`* `report-flooding switch-port` INTERFACE

Parameters

- **`v_range`** VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from ***1*** to ***4094***.
- **INTERFACE** Membership report message forwarding is enabled on these ports:
  - **ethernet *e_range*** where ***e_range*** is the number, range, or list of ethernet ports.
  - **port-channel *p_range*** where ***p_range*** is the number, range, or list of channel ports.

Related Commands

- **ip igmp snooping report-flooding** globally enables L2 report flooding.
- **ip igmp snooping vlan report-flooding switch-port** specifies a port list for a VLAN range.
- **ip igmp snooping report-flooding switch-port** specifies a port list for all VLANs.

Example

These commands globally enable L2 report flooding, enable flooding on VLANs ***201*** through ***205***, and specify Ethernet ports ***8-10*** as the report flooding port list for VLANS ***201-205***.

```
switch(config)# ip igmp snooping report-flooding
switch(config)# ip igmp snooping vlan 201-205 report-flooding
switch(config)# ip igmp snooping vlan 201-205 report-flooding switch-port
 ethernet 8-10
switch(config)#
```

16.2.8.48 ip igmp startup-query-count

The **`ip igmp startup-query-count`** command specifies the number of query messages that an interface sends during the startup interval defined by **ip igmp startup-query-interval**.

When an interface starts running IGMP, it can establish the group state more quickly by sending query messages at a higher frequency. The **startup-query-interval** and **startup-query-count** parameters define the startup period and the query message transmission frequency during that period.

The **`no ip igmp startup-query-count`** and **`default ip igmp startup-query-count`** commands restore the default **startup-query-count** value of ***2*** for the configuration mode interface by removing the corresponding **`ip igmp startup-query-count`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip igmp startup-query-count` ***number***

`no ip igmp startup-query-count`

`default ip igmp startup-query-count`

Parameter

number quantity of queries. Values range from ***1*** to ***65535***. Default is ***2***.

Example

This command configures the startup query count of ***10*** for ***vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp startup-query-count 10
switch(config-if-Vl4)#
```

16.2.8.49 ip igmp startup-query-interval

The **`ip igmp startup-query-interval`** command specifies the configuration mode interface IGMP startup period, during which query messages are sent at an accelerated rate.

When an interface starts running IGMP, it can establish the group state quicker by sending query messages at a higher frequency. The **startup-query-interval** and **startup-query-count** parameters define the startup period and the query message transmission frequency during that period.

The **`no ip igmp startup-query-interval`** and **`default ip igmp startup-query-interval`** commands restore the configuration mode interface default IGMP startup-query-interval of ***31*** seconds by removing the corresponding **`ip igmp startup-query-interval`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`ip igmp startup-query-interval` ***period***

`no ip igmp startup-query-interval`

`default ip igmp startup-query-interval`

Parameter

period startup query interval, in deciseconds. Value ranges from ***10*** (one second) to ***317440*** (8 hours, 49 minutes, 4 seconds). Default is ***31*** seconds.

Example

This command configures the startup query count of ***600*** deciseconds for ***vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp startup-query-interval 600
switch(config-if-Vl4)#
```

16.2.8.50 ip igmp static-group

The **`ip igmp static-group`** command configures the configuration mode interface as a static member of a specified multicast group. This allows the router to forward multicast group packets through the interface without otherwise appearing or acting as a group member. By default, static group memberships are not configured on any interfaces.

If the command includes a source address, only multicast group messages received from the specified host address are fast-switched. Otherwise, all multicast messages of the specified group are fast-switched.



Note: To become a static member of a multicast group, the switch must be the PIM Designated Router (DR) for the network. If it is not, you can use the **pim ipv4 dr-priority** command to make it the DR by configuring its PIM DR value to be the highest on the network.

The **no ip igmp static-group** and **default ip igmp static-group** commands remove the configuration mode interface group membership by removing the corresponding **ip igmp static-group** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip igmp static-group ***group_address*** [**SOURCE_ADDRESS**]

no ip igmp static-group ***group_address*** [**SOURCE_ADDRESS**]

default ip igmp static-group ***group_address*** [**SOURCE_ADDRESS**]

Parameters

- ***group_address*** IPv4 address of multicast group for which the interface fast-switches packets.
- **SOURCE_ADDRESS** IP address of host that originates multicast data packets.
  - ***no parameter*** all multicast messages of the specified group are fast-switched.
  - ***ipv4_address*** source IP address (dotted decimal notation).

Related Commands

- **ip igmp static-group acl** configures the configuration mode interface as a static member of the multicast groups specified by an IP Access Control List (ACL).
- **ip igmp static-group range** configures the configuration mode interface as a static member of multicast groups specified by an address range.

  One **ip igmp static-group range** command is equivalent to multiple **ip igmp static-group** commands.

Example

These commands configure ***interface vlan 15*** as the PIM designated router, then configure it as a static member of the multicast group at address ***231.1.1.15*** for multicast data packets that originate at ***10.1.1.1***.

```
switch(config)# interface vlan 15
switch(config-if-Vl15)# pim ipv4 dr-priority 5000
switch(config-if-Vl15)# ip igmp static-group 231.1.1.45 10.1.1.1
switch(config-if-Vl15)#
```

16.2.8.51 ip igmp static-group acl

The **ip igmp static-group acl** command configures the configuration mode interface as a static member of the multicast groups specified by an IP Access Control List (ACL). This command is a variant of the **ip igmp static-group** command that uses ACL rules to specify a set of source-multicast group address pairs instead of specifying a single pair. Multiple static-group ACLs can be assigned to an interface. Static groups can be assigned manually and through ACLs simultaneously.

Access control lists that this command references must contain rules of the following format.

- **permit <protocol><source><destination>**, where
  - **<protocol>** has no effect on the static group.
  - **<source>** address of host originating multicast data packets. Must be a host address.
  - **<destination>** multicast group IP address or subnet. Must be a valid multicast.

An ACL can contain multiple rules. An ACL can be applied to an interface only when all of its rules comply to the specified restrictions. The **show ip igmp static-groups acl** displays the source-multicast group pairs that the specified list configures and lists issues with illegal rules.

The **no ip igmp static-group acl** and **default ip igmp static-group acl** commands remove the specified static group ACL command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip igmp static-group acl ***list_name***

no ip igmp static-group acl

default ip igmp static-group acl

Parameters

list_name ACL that specifies multicast group addresses for which interface fast-switches packets.

Example

This command configures ***vlan 4*** as a static member of the multicast group specified by the ACL named ***LIST_1***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp static-group acl LIST_1
switch(config-if-Vl4)#
```

16.2.8.52 ip igmp static-group range

The **ip igmp static-group range** command configures the configuration mode interface as a static member of multicast groups specified by an address range. This allows the router to forward multicast group packets through the interface without otherwise appearing or acting as a group member. By default, no static group memberships are configured on interfaces.

This command is a variant of the **ip igmp static-group** command that allows the assignment of a subnet range of source addresses or a subnet range of multicast groups. A single **ip igmp static-group range** command is the equivalent of multiple **ip igmp static-group** commands, each of which can only assign a single multigroup-source pair to an interface. Running-config converts the range command to the equivalent list of **ip igmp static-group** commands.

If the command includes a source address range, only multicast group messages received from the range are fast-switched. Otherwise, all multicast messages of the specified group are fast-switched.

The **no ip igmp static-group range** and **default ip igmp static-group range** commands remove the specified range of static group statements from ***running-config***. The **no ip igmp static-group** and **default ip igmp static-group** commands can remove an individual static-group command that was initially added to ***running-config*** by an **ip igmp static-group range** command.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip igmp static-group range **GROUP_ADDR**[**SOURCE_ADDR**]

no ip igmp static-group range **GROUP_ADDR**[**SOURCE_ADDR**]

default ip igmp static-group range **GROUP_ADDR**[**SOURCE_ADDR**]

Parameters

- **GROUP_ADDR** address of multicast group for which the interface fast-switches packets.
  - ***gp_ipv4_addr*** multicast group IPv4 address.
  - ***gp_ipv4_subnet*** IPv4 subnet address of multicast groups (CIDR or address-mask).
- **SOURCE_ADDR** IP address of a host range that originates multicast data packets.
  - ***no parameter*** all multicast messages of the specified range are fast-switched.
  - **source *sr_ipv4_address*** source IPv4 address (dotted decimal notation).
  - **source *sr_ ipv4_subnet*** IPv4 subnet address of source hosts (CIDR or address- mask).

Note: A command cannot specify a subnet address for both multicast group and source.

Examples

- This command configures ***interface vlan 4*** as a static member of the multicast group range ***241.1.4.1/24*** for data packets that originate at ***10.1.1.1***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp static-group range 239.1.4.1/24 source
 10.1.1.1
switch(config-if-Vl4)#
```

- This command attempts to configure ***interface vlan 4*** as a static member of the multicast group range ***241.1.4.1/24*** for data packets that originate at the ***10.1.1.1/29*** subnet. Because the range and source cannot both be subnets, this command generates an error message.

```
switch(config-if-Vl4)# ip igmp static-group range 239.1.1.1/29 source
 16.1.1.1/29
% Error: cannot specify source range with group range
switch(config-if-Vl4)#
```

16.2.8.53 ip igmp version

The **ip igmp version** command configures the Internet Group Management Protocol (IGMP) version on the configuration mode interface. Version 3 is the default IGMP version.

IGMP is enabled by the **pim ipv4 sparse-mode** or **pim ipv4 bidirectional** command. The **ig igmp version** command does not affect the IGMP enabled status.

The **no ip igmp version** and **default ip igmp version** commands restore the configuration mode interface to IGMP version 3 by removing the **ip igmp version** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

ip igmp version ***version_number***

no ip igmp version

default ip igmp version

Parameters

version_number IGMP version number. Value ranges from ***1*** to ***3***.

Example

This command configures IGMP version 3 on ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# ip igmp version 3
switch(config-if-Vl4)#
```

16.2.8.54 permit / deny

The **permit** command configures the configuration mode IGMP profile as a permit list. Applying a permit list to an interface restricts that interface from joining any multicast group not included in the list.

IGMP profiles are deny lists by default. When applied to an interface, a deny list allows the interface to join any multicast group that is not included in the list.

The **deny** command restores the IGMP list to its default type by removing the corresponding **permit** statement from ***running-config***.

The **range** command adds and removes address ranges from the configuration mode profile.

Command Mode

IGMP-profile Configuration

Command Syntax

permit

deny

Related Commands

ip igmp profile places the switch in ***igmp-profile*** configuration mode.

Example

These commands enter ***igmp-profile*** configuration mode and configure the profile as permit ***list_1***.

```
switch(config)# ip igmp profile list_1
switch(config-igmp-profile-list_1)# permit
switch(config-igmp-profile-list_1)#
```

16.2.8.55 range

The **range** command specifies an address range for the configuration mode IGMP profile. A permit range specifies the groups that an interface is permitted to join. A deny range specifies the groups that an interface is not permitted to join. The **permit / deny** command specifies the range type.

A profile may contain multiple range statements to define a discontiguous address range.

The **no range** and **default range** commands remove the specified address range from a previous specified list.

Command Mode

igmp-profile Configuration

Command Syntax

range ***init_address*** [**UPPER_RANGE**]

no range ***init_address*** [**UPPER_RANGE**]

`default range` *init_address* [**UPPER_RANGE**]

Parameters

- ***init_address*** IP address of lower boundary of the address range (dotted decimal notation).
- **UPPER_RANGE** sets the upper boundary of the address range. Options include:
  - ***no parameter*** upper boundary is equal to lower boundary: range consists of one address.
  - ***range_address*** IP address of upper boundary.

    All addresses must be multicast addresses (***10.0.0.0*** to ***239.255.255.255***).

Related Commands

ip igmp profile places the switch in the ***igmp-profile*** configuration mode.

Example

These commands enter the ***igmp-profile*** configuration mode, configure the profile as a permit list, and define the permit address list of ***232.1.1.0 to 232.1.1.255*** and ***233.1.1.10***.

```
switch(config)# ip igmp profile list_1
switch(config-igmp-profile-list_1)# permit
switch(config-igmp-profile-list_1)# 232.1.1.0 232.1.1.255
switch(config-igmp-profile-list_1)# 233.1.1.10
switch(config-igmp-profile-list_1)# ip igmp profile
```

16.2.8.56 show igmp snooping querier

The **`show igmp snooping querier`** command displays snooping querier configuration and status information. Command provides options to only include specific VLANs.

Command Mode

EXEC

Command Syntax

`show igmp snooping querier` [**STATUS**][**VLAN_ID**][**DATA**]

Parameters

- **STATUS** specifies the type of information displayed. Options include:
  - ***no parameter*** querier IP address, port, and IGMP version.
  - **status** querier configuration parameters.
- **VLAN_ID** specifies VLANs for which command displays information. Options include:
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** specified VLAN.
- **DATA** specifies the type of information displayed. Options include:
  - ***no parameter*** displays VLAN number and port-list for each group.
  - **detail** displays port-specific data for each group; includes transmission times and expiration.

Examples

- This command displays the querier IP address, version, and port servicing each VLAN.

```
switch> show igmp snooping querier
Vlan  IP Address       Version  Port
---------------------------------------
1     172.17.0.37      v2       Po1
20    172.17.20.1      v2       Po1
26    172.17.26.1      v2       Cpu
```

```
2028  172.17.255.29    v2       Po1 switch>
```

- This command displays the querier configuration parameters for each VLAN.

```
switch> show igmp snooping querier status
   Global IGMP Querier status
-----------------------------------
admin state              : Enabled
source IP address        : 0.0.0.0
query-interval (sec)     : 125.0
max-response-time (sec)  : 10.0
querier timeout (sec)    : 130.0

Vlan Admin    IP              Query    Response Querier Operational
     State                    Interval Time     Timeout State
-------------------------------------------------------------------
1    Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
4    Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
20   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
22   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
28   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
```

16.2.8.57 show igmp snooping querier counters

The **show igmp snooping querier counters** command displays the counters from the querier, as learned through Internet Group Management Protocol (IGMP).

Command Mode

EXEC

Command Syntax

show igmp snooping querier counters [**VLAN_ID**]

Parameters

VLAN_ID specifies VLANs for which command displays information. Options include:

- ***no parameter*** displays information for all VLANs.
- **vlan *v_num*** displays information for specified VLAN.

Example

This command displays the counters from the querier.

```
switch> show igmp snooping querier counters
-------------------------------------------------------------------------
Vlan: 1    IP Addr: 100.0.0.1       Op State: Querier     Version:  v3

v1 General Queries Sent         :0
v1 Queries Received             :0
v1 Reports Received             :0
v2 General Queries Sent         :1
v2 Queries Received             :0
v2 Reports Received             :25
v2 Leaves Received              :0
v3 General Queries Sent         :655
v3 GSQ Queries Sent             :0
v3 GSSQ Queries Sent            :8
v3 Queries Received             :654
v3 Reports Received             :2385
Error Packets                   :0
Other Packets                   :0 switch>
```

16.2.8.58 show igmp snooping querier membership

The **show igmp snooping querier membership** command displays the membership from the querier, as learned through Internet Group Management Protocol (IGMP).

Command Mode

EXEC

Command Syntax

show igmp snooping querier membership [**VLAN_ID** [**GROUP_LIST**]]

Parameters

- **VLAN_ID** specifies VLANs for which command displays information. Options include:
  - ***no parameter*** displays information for all VLANs.
  - **vlan *v_num*** displays information for specified VLAN.
- **GROUP_LIST** list of groups for which the command displays information. Options include:
  - ***no parameter*** all multicast groups within specified VLAN.
  - **group *ipv4_addr*** single multicast group address (dotted decimal notation).

Example

This command displays the membership from the querier for ***vlan 1***.

```
switch> show igmp snooping querier membership
-------------------------------------------------------------------------
Vlan: 1    Elected: 10.0.0.1       QQI: 125  QRV: 2  QRI: 10  GMI: 260

Groups           Mode  Ver  Num of Sources
-------------------------------------------------------------------------
10.0.0.2        EX    v3   0 []
10.0.0.3        IN    v3   2 [ 3.3.3.3, 3.3.3.4 ]
10.0.0.4        EX    v3   0 []
10.0.0.13       EX    v3   0 []
10.0.0.22       EX    v3   0 []
10.0.0.1        IN    v3   3 [ 5.6.7.9, 5.6.7.8, ... ]
switch>
```

16.2.8.59 show ip igmp groups

The **show ip igmp groups** command displays multicast groups that have receivers directly connected to the switch, as learned through Internet Group Management Protocol (IGMP).

- **show ip igmp groups** all multicast groups.
- **show ip igmp groups *group_addr*** listed multicast group.
- **show ip igmp groups interface *int_name*** all multicast groups on specified interfaces
- **show ip igmp groups *group_addr* interface *int_name*** listed multicast group on specified interface.

Command Mode

EXEC

Command Syntax

show ip igmp groups GROUP_LIST [**DATA**]

Parameters

- **GROUP_LIST** list of groups for which the command displays information. Options include:
  - ***no parameter*** all multicast groups.

- ***group_addr*** single multicast group address (dotted decimal notation).
- **interface ethernet *e_num*** all multicast groups on specified Ethernet interface.
- **interface loopback *l_num*** all multicast groups on specified Loopback interface.
- **interface management *m_num*** all multicast groups on specified Management interface.
- ***interface port-channel p_num*** all multicast groups on specified Port-Channel Interface.
- ***interface vlan v_num*** all multicast groups on specified VLAN interface.
- **interface vxlan *vx_num*** all multicast groups on specified VXLAN interface.

- **DATA** specifies the type of information displayed. Options include:
  - ***no parameter*** provides uptime, expiration, and address of reporter.
  - **detail** also include group mode and group source list.

Example

This command displays multicast groups with receivers directly connected to the switch.

```
switch> show ip igmp groups
NOTE: static-group information not shown below.  Use the
      'show ip igmp static-groups' command.
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
10.12.1.1       Vlan162                 11d01h    00:02:57  172.17.2.110
10.12.1.2       Vlan162                 11d01h    00:02:57  172.17.2.110
10.12.1.3       Vlan162                 11d01h    00:02:57  172.17.2.110
10.12.1.4       Vlan162                 11d01h    00:02:57  172.17.2.110
10.12.1.5       Vlan162                 11d01h    00:02:57  172.17.2.110
 switch>
```

16.2.8.60 show ip igmp groups count

The **`show ip igmp groups count`** command displays the number of multicast groups that are joined across the specified interfaces.

Command Mode

EXEC

Command Syntax

`show ip igmp groups` [**GROUP_LIST**] **`count`**

Parameters

- **INTERF** Specifies the interface for which the command displays information. Options include:
  - ***no parameter*** all IGMP interfaces.
  - **interface ethernet *e_num*** Ethernet interface.
  - **interface loopback *l_num*** Loopback interface.
  - **interface management *m_num*** Management interface.
  - **interface port-channel *p_num*** Port-Channel Interface.
  - **interface vlan *v_num*** VLAN interface.
  - **interface vxlan *vx_num*** VXLAN interface.

Examples

- This command displays the number of multicast groups joined across all interfaces.

```
switch> show ip igmp groups count
Number of total groups joined across all IGMP interfaces: 5 switch>
```

- This command displays the number of multicast groups joined on ***interface ethernet 3/4***.

```
switch> show ip igmp groups interface ethernet 3/4 count
Number of groups joined on Ethernet3/4: 2 switch>
```

16.2.8.61 show ip igmp host-proxy config-sanity

The **show ip igmp host-proxy config-sanity** command displays diagnostic information about an IGMP host proxy configuration.

Command Mode

EXEC

Command Syntax

show ip igmp host-proxy config-sanity

Example

This command displays IGMP host proxy configuration diagnostic information.

```
switch> show ip igmp host-proxy config-sanity
Below are hints of potential IGMP Host-Proxy misconfigurations
IGMP host-proxy configured on interface Test3:
Access-lists having "deny ip any any" rule:
acl1
acl2
Groups with overlapping permit and deny configurations:
192.168.1.1/32
192.168.2.2/32
192.168.4.4/32
Groups with source filters configured with IGMP Host-Proxy set to version 2 :
192.168.2.2/32
192.168.3.0/24
192.168.3.3/32
192.168.8.8/32 switch>
```

16.2.8.62 show ip igmp host-proxy interface

The **show ip igmp host-proxy interface** command displays per-interface IGMP host-proxy configuration information, including the IGMP groups joined on the interface. Command filters allow the list to display only data for a specified interface and to include packet counter statistics in the display.

Command Mode

EXEC

Command Syntax

show ip igmp host-proxy interface [***interface***] [**detail**]

Parameters

- ***interface*** optional parameter to limit the display to a single interface. Omitting the parameter displays host-proxy configuration information for all interfaces on which IGMP host-proxy is configured. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **port-channel *p_num*** port-channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
- **detail** use this optional keyword to include packet counter statistics in the display.

Examples

- This command displays host-proxy information for all switch interfaces on which IGMP host-proxy is configured.

```
switch> show ip igmp host-proxy interface
IGMP host-proxy configured on: Test2
IGMP host-proxy version: 3
Unsolicited-report interval: 1.0
Device name: Test2
Interface Group Address IncludeSrc ExcludeSrc
Test2 172.16.89.0
Test2 172.16.0.0 20.0.0.0
Test2 172.16.0.0 10.0.0.0
Test2 192.168.110.0 20.0.0.0
```

- This command displays host-proxy information for all switch interfaces on which IGMP host-proxy is configured, plus host-proxy statistics.

```
switch> show ip igmp host-proxy interface detail
IGMP host-proxy configured on: Test2
IGMP host-proxy version: 3
Unsolicited-report interval: 1.0
Device name: Test2
Interface Group Address IncludeSrc ExcludeSrc
Test2 172.16.89.0
Test2 172.16.0.0 20.0.0.0
Test2 172.16.0.0 10.0.0.0
Test2 192.168.110.0 20.0.0.0
IGMP host-proxy statistics:
IGMP v1 Queries received: 0
IGMP v2 General-Queries received: 0
IGMP v2 Group-Queries received: 0
IGMP v3 General-Queries received: 0
IGMP v3 Group-Queries received: 0
IGMP v3 Group-Source Queries received: 0
IGMP v1 Reports sent: 0
IGMP v2 Reports sent: 0
IGMP v3 Reports sent: 1
```

16.2.8.63 show ip igmp interface

The **show ip igmp interface** command displays multicast information about the specified interface.

Command Mode

EXEC

Command Syntax

show ip igmp interface [**INT_NAME**]

Parameters

INT_NAME Interface type and number. Values include:

- ***no parameter*** Displays information for all interfaces.
- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **vlan *p_num*** VLAN interface specified by ***p_num***.
- **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.

Example

This command displays multicast related information about ***vlan 26***.

```
switch> show ip igmp interface vlan 26
Vlan26 is up
  Interface address: 172.17.26.1/23
  IGMP on this interface: enabled
  Multicast routing on this interface: enabled
  Multicast TTL threshold: 1
  Current IGMP router version: 2
  IGMP query interval: 125 seconds
  IGMP max query response time: 100 deciseconds
  Last member query response interval: 10 deciseconds
  Last member query response count: 2
  IGMP querier: 172.17.26.1
  Robustness: 2
  Require router alert: enabled
  Startup query interval: 312 deciseconds
  Startup query count: 2
  General query timer expiry: 00:00:22
  Multicast groups joined:
    239.255.255.250 switch>
```

16.2.8.64 show ip igmp profile

The **`show ip igmp profile`** command displays the contents of the specified IGMP profile. IGMP snooping filters use an IGMP profile to control the multicast groups that an interface can join.

Command Mode

EXEC

Command Syntax

`show ip igmp profile` [**PROFILES**]

Parameters

PROFILES IGMP profiles for which command displays contents. Options include:

- ***no parameter*** displays all IGMP profiles.
- ***profile_name*** displays specified profile.

Example

This command displays the IGMP profiles configured on the switch.

```
switch> show ip igmp profile
IGMP Profile list_1
permit
range 229.1.24.0 229.1.25.255
IGMP Profile list_2
deny
range 234.1.1.0 234.1.255.255 switch>
```

16.2.8.65 show ip igmp snooping

The **`show ip igmp snooping`** command displays the switch IGMP snooping configuration.

Command Mode

EXEC

Command Syntax

show ip igmp snooping [**VLAN_ID**]

Parameters

VLAN_ID specifies VLANs for which command displays information. Options include:

- ***no parameter*** displays information for all VLANs.
- **vlan *v_num*** displays information for specified VLAN.

Example

This command displays the switch IGMP snooping configuration.

```
switch> show ip igmp snooping
   Global IGMP Snooping configuration:
-----------------------------------------
IGMP snooping                   : Enabled
Robustness variable             : 2

Vlan 1 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 20 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 26 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 2028 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp switch>
```

16.2.8.66 show ip igmp snooping counters

The **show ip igmp snooping counters** command displays the number of IGMP messages sent and received through each switch port. The display table sorts the messages by type.

Command Mode

EXEC

Command Syntax

show ip igmp snooping counters [**DATA_TYPE**][**DATA_LEVEL**]

Parameters

- **DATA_TYPE** Information displayed by the command. Options include:
  - ***no parameter*** displays transmission counters.
  - **errors** displays error counters.
- **DATA_LEVEL** specifies the type of information displayed. Options include:
  - ***no parameter*** number of packets on physical ports.
  - **detail** number of packets on physical ports.

Example

This command displays the number of messages received on each port.

```
switch> show ip igmp snooping counters
                        Input                 |           Output
Port   Queries Reports  Leaves  Others  Errors|Queries Reports  Leaves  Others
-------------------------------------------------------------------------------
Cpu      15249  106599       4  269502       0   30242  102812     972    3625
Et1          0       0       0       0       0       0       0       0       0
Et2          0       6       1      26       0    5415       0       0     731
Et3          0   10905     222    1037       0   15246       0       0    1448
Et4          0   44475      21     288       0   15247       0       0    2199
Et5          0     355       0      39       0   15211       0       0    2446
Et6          0     475      13       0       0   15247       0       0    2487
Et7          0       0       0     151       0   15247       0       0    2336
Et8          0     578       6      75       0    2859       0       0     931
Et9          0       0       0      27       0   15247       0       0    2460
Et10         0   12523     345      54       0   15247       0       0    2433
Et11         0       0       0       0       0       0       0       0       0
Et12         0    4509      41      22       0   15247       0       0    2465
Et13         0     392      29     119       0   15247       0       0    2368
Et14         0      88       3       6       0   15247       0       0    2481
Et15         0   16779     556      72       0   15117       0       0      66
Et16         0    2484      13      66       0   15247       0       0    2421
Et17         0       0       0       0       0       0       0       0       0
Et18         0      20       6     160       0    3688       0       0     803
Et19         0    4110      17       0       0   15247       0       0    2487
Et20         0       0       0       0       0       0       0       0       0
Et21         0       0       0       0       0       0       0       0       0
Et22         0       0       0      52       0   15247       0       0    2435
Et23         0    5439     181     138       0   15247       0       0    2349
Et24         0    2251      21       4       0   15247       0       0    2483
Po1      45360  540670    8853  464900       0   15249  224751     618    2576
Po2          0  101399      58      17       0   15120       0       0    1121
Switch       0       0       0       0       0       0       0       0       0
```

16.2.8.67 show ip igmp snooping counters ethdev-pams

The **show ip igmp snooping counters** command displays the number of dropped IGMP packets messages sent and received through each switch port at the kernel level. The display table sorts the messages by type.

Command Mode

EXEC

Command Syntax

show ip igmp snooping counters ethdev-pams

Example

This command displays the number of messages dropped at the kernel level.

```
switch> show ip igmp snooping counters ethdev-pams
  IntfName   rxErrors   txErrors    txDrops
       et9          1          0          0
      et18          1          0          0
     mlag9          1          0          0
     mlag8          1          0          0
      et17          1          0          0
       po1          1          0          0
       po2          1          0          0
      et15          1          0          0
       et6          1          0          0
    mlag10          1          0          0
      et16          1          0          0
     mlag7          1          0          0
      et11          1          0          0
     mlag5          1          0          0
     mlag4          1          0          0
```

```
        cpu           1           0           0
       et13           1           0           0 switch>
```

16.2.8.68 show ip igmp snooping groups

The **show ip igmp snooping groups** command displays IGMP snooping statistics. Available information includes the physical ports that send and receive information, the time when multicast data was originally and most recently heard on the ports, and the version number of the IGMP messages. Command provides options that restrict the output to specific VLANs, ports, and groups.

Command Mode

EXEC

Command Syntax

show ip igmp snooping groups ***proxy*** [**VLAN_ID**][**PORT_INT**][**GROUPS**][**DATA local**]

Parameters

- **VLAN_ID** specifies VLAN for which command displays information. Options include:
  - ***no parameter*** displays information for all VLANs.
  - **vlan *v_num*** displays information for VLAN ***v_num*** (***1*** to ***4094***).
- **PORT_INT** specifies physical ports for which command displays information. Options include:
  - ***no parameter*** displays information for all physical ports.
  - **interface ethernet *e_range***, where ***e_range*** is the number, range, or list of Ethernet ports.
  - **interface port-channel *p_range***, where ***p_range*** is the number, range, or list of channel ports.
- **GROUPS** specifies the multicast groups. Options include:
  - ***no parameter*** all multicast groups on all specified ports.
  - ***mgroup_address*** multicast group specified by IPv4 address (dotted decimal notation).
  - **dynamic** multicast groups learned through IGMP.
  - **user** multicast groups manually added.
- **DATA** specifies the type of information displayed. Options include:
  - ***no parameter*** VLAN number and port-list for each group.
  - **detail** port-specific information for each group, including transmission times and expiration.
- **proxy** displays IGMP snooping proxy information.
- **local** displays the overlay with locally attached receivers.

Examples

- This command displays the port lists for all multicast groups.

```
switch> show ip igmp snooping groups
Vlan  Group            Type     Version      Port-List
------------------------------------------------------------------------
-
1     239.255.255.250  -        -            Po1, Po2
26    239.255.255.250  -        -            Cpu, Et3, Et4, Et10, Et23,
                                             Et27 switch>
```

- This command displays detailed port information of all multicast groups.

```
switch> show ip igmp snooping groups detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                     Heard     Heard                 Mode
-------------------------------------------------------------------------------
-
1    239.255.255.250 172.17.3.73     2536:15   0:47    3:33      v2  0      Po2
1    239.255.255.250 172.17.0.37     31532:48  0:18    1:27      -   -      Po1
26   239.255.255.250 172.17.26.189   5:07      0:52    3:28      v2  0      Et3
26   239.255.255.250 172.17.26.182   17:34     3:02    1:18      v2  0      Et3
26   239.255.255.250 172.17.26.245   1046:47   0:57    3:23      v2  0      Et4
```

```
26   239.255.255.250 172.17.26.184   27:41     0:53    3:27      v2  0      Et10
26   239.255.255.250 172.17.26.161   9:16      0:56    3:24      v2  0      Et23
26   239.255.255.250 172.17.26.62    90:24     0:50    3:30      v2  0      Et27
26   239.255.255.250 172.17.26.1     31532:52  0:04    1:41      -   -      Cpu switch>
```

- This command displays the port lists for all dynamic multicast groups.

```
switch> show ip igmp snooping groups dynamic
Vlan  Group            Type     Version    Port-List
-----------------------------------------------------------------
-
1     239.255.255.250  -        -          Po1, Po2
26    239.255.255.250  -        -          Cpu, Et3, Et4, Et10, Et23,
                                           Et27, Et34 switch>
```

- This command displays the detailed port information for all dynamic multicast groups.

```
switch> show ip igmp snooping groups dynamic detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                     Heard     Heard               Mode
-------------------------------------------------------------------------------
-
1    239.255.255.250 172.17.3.73     2539:16   1:37    2:43      v2  0      Po2
1    239.255.255.250 172.17.0.37     31535:49  0:19    1:26      -   -      Po1
26   239.255.255.250 172.17.26.189   8:08      3:53    0:27      v2  0      Et3
26   239.255.255.250 172.17.26.182   20:35     1:49    2:31      v2  0      Et3
26   239.255.255.250 172.17.26.245   1049:48   1:46    2:34      v2  0      Et4
26   239.255.255.250 172.17.26.184   30:42     1:44    2:36      v2  0      Et10
26   239.255.255.250 172.17.26.161   12:17     3:57    0:23      v2  0      Et23
26   239.255.255.250 172.17.26.143   1:53      1:53    2:27      v2  0      Et23
26   239.255.255.250 172.17.26.62    93:25     1:48    2:32      v2  0      Et27
26   239.255.255.250 172.17.26.164   0:32      0:31    3:49      v2  0      Et34
26   239.255.255.250 172.17.26.1     31535:53  0:05    1:40      -   -      Cpu switch>
```

- This command displays the port lists for all static (user configured) multicast groups.

```
switch> show ip igmp snooping groups user
Vlan  Group            Type     Version       Port-List
-----------------------------------------------------------------------
-
1     239.255.255.250  -        -             Po1, Po2
26    239.255.255.250  -        -             Cpu, Et3, Et4, Et10, Et23,
                                              Et27, Et34 switch>
```

- This command displays detailed port information for all user configured (static) multicast groups.

```
switch> show ip igmp snooping groups user detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                     Heard     Heard               Mode
-------------------------------------------------------------------------------
-
1    239.255.255.250 172.17.3.73     2539:50   0:06    4:14      v2  0      Po2
1    239.255.255.250 172.17.0.37     31536:23  0:23    1:22      -   -      Po1
26   239.255.255.250 172.17.26.182   21:09     0:21    3:59      v2  0      Et3
26   239.255.255.250 172.17.26.245   1050:22   0:17    4:03      v2  0      Et4
26   239.255.255.250 172.17.26.184   31:16     0:17    4:03      v2  0      Et10
26   239.255.255.250 172.17.26.161   12:51     0:17    4:03      v2  0      Et23
26   239.255.255.250 172.17.26.143   2:27      2:27    1:53      v2  0      Et23
26   239.255.255.250 172.17.26.62    93:59     0:22    3:58      v2  0      Et27
26   239.255.255.250 172.17.26.164   1:06      0:21    3:59      v2  0      Et34
26   239.255.255.250 172.17.26.1     31536:27  0:09    1:36      -   -      Cpu switch>
```

- This command displays detailed port information for multicast group ***239.255.255.253*** on ***vlan 10***.

```
switch> show ip igmp snooping groups vlan 10 239.255.255.253 detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                     Heard     Heard               Mode
-------------------------------------------------------------------------------
-
10   239.255.255.253 10.255.255.246  7177:16   0:08    2:07      v2  0      Po7
10   239.255.255.253 10.255.255.247  7177:20   0:03    2:12      v2  0      Po7
10   239.255.255.253 10.255.255.248  7177:16   0:06    2:09      v2  0      Po7
10   239.255.255.253 10.255.255.254  7177:56   0:07    1:38      -   -      Cpu switch>
```

- This command displays the groups that is present in IGMP report when a query is received on any of the ports listed under port-list.

```
switch> show ip igmp snooping groups proxy
Vlan      Group     Type     Port-List
-----------------------------------------------
10     225.0.0.1    Proxy   Cpu, Et4, Et6
10     225.2.2.2    Proxy   Cpu, Et3, Et4
10     225.3.3.3    Proxy   Cpu, Et3, Et4, Et6
```

- This command displays all the information that is present in the IGMP report if a general IGMP query was received on ***Ethernet4***.

```
switch> show ip igmp snooping groups proxy interface Ethernet4 detail
Vlan     Interface     Group          Source/Filter Mode
-----------------------------------------------------------
10       Ethernet4    225.0.0.1       Include
                                      150.227.112.250
                                      190.171.60.6
10       Ethernet4     225.2.2.2      Exclude
10       Ethernet4     225.3.3.3      Exclude
                                      150.227.112.250
```

- This command displays groups in the overlay with locally attached receivers.

```
switch> show ip igmp snooping groups local
IGMP Snooping Group Membership
EX : Filter mode Exclude
IN : Filter mode Include
IR : Ingress Replication

VLAN  Group           Members
----  --------------  -------------------------
10    227.1.1.1       Et1
10    *               Cpu
20    228.1.1.1       Et2
20    *               Cpu
30    227.1.1.1       Et3
30    *               Cpu
40    228.1.1.1       Et4
40    *               Cpu
Vlan     Interface     Group            Source/Filter Mode
-----------------------------------------------------------
10       Ethernet4    225.0.0.1        Include
                                       150.227.112.250
                                       190.171.60.6
10       Ethernet4     225.2.2.2       Exclude
10       Ethernet4     225.3.3.3       Exclude
                                       150.227.112.250
```

- This command displays groups in the overlay with remote receivers, which are learned via SMET and Join-Synch routes.

```
switch> show ip igmp snooping groups evpn
IGMP Snooping Group Membership
EX : Filter mode Exclude
IN : Filter mode Include
IR : Ingress Replication
VLAN  Group            Members
----  ---------------  ----------------
4093  227.1.1.1        PIM-Tunnel
4094  228.1.1.1        PIM-Tunnel
```

Note: Tthe port PIM-Tunnel indicates that packets destined to these groups is encapsulated in a multicast header and sent over the underlay network.

16.2.8.69 show ip igmp snooping groups count

The **`show ip igmp snooping groups count`** command displays the number of multicast groups on the switch. Command provides options to only include specific VLANs and ports.

Command Mode

EXEC

Command Syntax

`show ip igmp snooping groups` [**VLAN_ID**][**PORT_INT**] **`count`** [**DATA**]

Parameters

- **VLAN_ID** specifies VLAN for which command displays information. Options include:
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** specified VLAN.
- **PORT_INT** specifies physical ports for which command displays information. Options include:
  - ***no parameter*** all physical ports.
  - **interface ethernet *e_range*** specified Ethernet ports.
  - **interface port-channel *p_range*** specified port channels.

    Valid ***e_range*** and ***p_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **DATA** specifies the type of information displayed. Options include:
  - ***no parameter*** number of multicast group on specified VLAN and ports.
  - **detail** number of multicast group on specified VLAN and ports.

Example

This command displays the number of multicast groups on the switch.

```
switch> show ip igmp snooping groups count
Total number of multicast groups: 2 switch>
```

16.2.8.70 show ip igmp snooping mrouter

The **`show ip igmp snooping mrouter`** command displays the status of dynamic and static multicast router ports. Command provides options to include only specific VLANs.

Command Mode

EXEC

Command Syntax

`show ip igmp snooping mrouter` [**VLAN_ID**][**DATA**]

Parameters

- **VLAN_ID** specifies VLAN for which command displays information. Options include:
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** specified VLAN.
- **DATA** specifies the type of information displayed. Options include:
  - ***no parameter*** displays VLAN number and port-list for each group.
  - **detail** displays port-specific data for each group; includes transmission times and expiration.

Examples

- This command displays port information of each multicast router on all VLANs.

```
switch> show ip igmp snooping mrouter
Vlan    Interface-ports
---------------------------------------
1       Po1(dynamic)
20      Po1(dynamic)
26      Cpu(dynamic)
2028    Cpu(dynamic), Po1(dynamic)
switch>
```

- This command displays multicast router information for each port.

```
switch> show ip igmp snooping mrouter detail
Vlan   Intf      Address           FirstHeard LastHeard  Expires    Type
-------------------------------------------------------------------------
1      Po1       172.17.0.37       31549:12   0:12       1:33       pim
20     Po1       172.17.20.1       7066:51    0:19       1:26       pim
26     Cpu       172.17.26.1       31549:16   0:28       1:17       pim
2028   Po1       172.17.255.29     31549:10   0:18       1:27       pim
2028   Cpu       172.17.255.30     31549:14   0:28       1:17       pim switch>
```

16.2.8.71 show ip igmp snooping report-flooding

The **`show ip igmp snooping report-flooding`** command displays IGMP snooping L2 report flooding configuration and status information. Command provides options to only include specific VLANs.

Command Mode

EXEC

Command Syntax

`show ip igmp snooping report-flooding` [**VLAN_ID**][**DATA**]

Parameters

- **VLAN_ID** specifies VLANs for which command displays information. Options include:
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** specified VLAN.
- **DATA** specifies the type of information displayed. Options include:
  - ***no parameter*** displays VLAN number and port-list for each group.
  - **detail** displays port-specific data for each group; includes transmission times and expiration.

16.2.8.72 show ip igmp static-groups

The **`show ip igmp static-groups`** command displays information about all configured IGMP multicast static groups. IGMP multicast static groups are assigned with the **ip igmp static-group** command.

Command Mode

EXEC

Command Syntax

`show ip igmp static-groups` [**INFO_LEVEL**][interface **INT_NAME**]

Parameters

- **INFO_LEVEL** specifies the type of information displayed. Options include:
  - ***no parameter*** VLAN number and port-list for each group.
  - **detail** port-specific information for each group, including transmission times and expiration.
- **INT_NAME** Interface type and number. Values include:

- ***no parameter*** static groups on all interfaces.
- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.

Related Commands

- **show ip igmp static-groups acl**
- **show ip igmp static-groups group**

Examples

- This command displays information about all multicast static groups.

```
switch> show ip igmp static-groups
Interface Vlan281:
        Manually configured groups:
Interface Port-Channel999:
        Manually configured groups:
switch>
```

- This command displays information about the multicast static groups on ***interface vlan 21***.

```
switch> show ip igmp static-groups interface vlan 21
Interface Vlan281:
        Manually configured groups:
switch>
```

16.2.8.73 show ip igmp static-groups acl

The **`show ip igmp static-groups acl`** command displays information about the IGMP multicast static groups that are configured by the specified Access Control List (ACL). The command also displays problems with an ACL that prevent its assignment to an interface.

Command Mode

EXEC

Command Syntax

`show ip igmp static-groups acl`

Examples

- The following **`show ip igmp static-group acl`** command example references these ACLs:

```
ip access-list 1
   10 permit igmp host 10.1.1.1 10.1.1.0/29
   20 permit igmp host 10.1.1.2 10.1.1.0/29
!
ip access-list 2
   10 permit igmp 10.1.1.0/29 host 10.1.1.1
!
ip access-list 3
   10 deny igmp host 10.1.1.1 255.1.1.0/29
!
ip access-list 4
   10 permit igmp host 10.1.1.1 10.1.1.0/29
   20 permit igmp 10.1.1.0/29 host 10.1.1.1
```

- This command displays static group configuration data about the various ACLs.

```
switch> show ip igmp static-group acl 1
acl 1
        ( 10.1.1.1, 10.1.1.0/29 )
        ( 10.1.1.2, 10.1.1.0/29 )
Interfaces using this ACL for static groups:
        Ethernet12
switch>show ip igmp static-group acl 2
acl 2
        Seq no 30: source address must be a single host or *, not a range
Interfaces using this ACL for static groups:
        Ethernet8
switch>show ip igmp static-group acl 3
acl 4
        Seq no 10: action must be 'permit'
Interfaces using this ACL for static groups:
         none
switch>show ip igmp static-group acl 4
acl 5
        ( 10.1.1.1, 10.1.1.0/29 )
       Seq no 20: source address must be a single host or *, not a range
Interfaces using this ACL for static groups:
         none switch>
```

16.2.8.74 show ip igmp static-groups group

The **`show ip igmp static-groups group`** command displays information about all specified IGMP multicast static groups. IGMP multicast static groups are assigned with the **ip igmp static-group** command.

Command Mode

EXEC

Command Syntax

`show ip igmp static-groups group` [**GROUP_LIST**]

Parameters

GROUP LIST Groups for which command displays information:

- ***no parameter*** all multicast groups.
- ***group_address*** single multicast group address (dotted decimal notation).

Related Commands

show ip igmp static-groups

16.2.8.75 show ip igmp statistics

The **`show ip igmp statistics`** command displays IGMP transmission statistics for the specified interface.

Command Mode

EXEC

Command Syntax

`show ip igmp statistics` [**INTERFACE_ID**]

Parameters

INTERFACE_ID Specifies tnterface for which command returns data. Options include:

- ***no parameter*** all interfaces.
- **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **interface loopback *l_num*** Loopback interface specified by***l_num***.
- **interface management *m_num*** Management interface specified by ***m_num***.
- **interface port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **interface vlan *v_num*** VLAN interface specified by ***v_num***.
- **interface vxlan *vx_num*** VXLAN interface specified by***vx_num***.

Example

This command displays IGMP transmission statistics for ***interface ethernet 1***.

```
switch> show ip igmp statistics interface ethernet 1
IGMP counters for Ethernet1:
  V1 queries sent: 0
  V2 queries sent: 0
  V3 queries sent: 3
  Total general queries sent: 3
  V3 group specific queries sent: 0
  V3 group-source specific queries sent: 0
  V1 queries received: 0
  V2 queries received: 0
  V3 queries received: 0
  V1 reports received: 0
  V2 reports received: 0
  V3 reports received: 14
  V2 leaves received: 0
  Error Packets received: 0
  Other Packets received: 0 switch>
```

16.3 Protocol Independent Multicast

Protocol Independent Multicast (PIM) distributes multicast data using routes gathered by other protocols. Arista switches support two types of PIM: PIM Sparse Mode (PIM-SM) and Bidirectional PIM (Bidir-PIM).

These sections describe the Arista PIM implementation:

- Introduction
- Overview
- Configuring PIM
- Multicast Example
- PIM Commands

16.3.1 Introduction

Protocol Independent Multicast (PIM) distributes multicast data using routes gathered by other protocols. PIM Sparse Mode (PIM-SM), defined in ***RFC 4601***, is a multicast routing protocol intended for networks where multicast group recipients are sparsely distributed, including wide-area and inter-domain networks. Bidirectional PIM (Bidir-PIM), defined in ***RFC 5015***, is a variant of PIM-SM designed for cases in which the receivers of multicast traffic are also sources and where scalability could affect optimization.

Arista switches support both PIM Sparse-Mode (PIM-SM) and Bidirectional PIM (Bidir-PIM).

16.3.2 Overview

PIM builds and maintains multicast routing trees using Reverse Path Forwarding (RPF) on a unicast routing table. PIM is protocol-independent, and can use routing tables consisting of OSPF, BGP, RIP, and static

routes. All sources send traffic to multicast groups through shared trees that have a common root node called the Rendezvous Point (RP).

PIM uses a Multicast Routing Information Base (MRIB) that is populated from the unicast table. The MRIB provides the next-hop router for each multicast destination subnet. This determines the next-hop neighbor for sending PIM join or prune messages.

16.3.2.1 PIM Sparse Mode

In PIM-SM, each host (sender or receiver) is associated with a Designated Router (DR) that acts for all directly connected hosts in PIM-SM transactions, and trees are unidirectional. Once sufficient traffic is flowing on a route it usually does not pass through the RP.

PIM-SM establishes multicast routes through three phases:

- Establishing the RP Tree
- Eliminating Encapsulation
- Establishing the Shortest Path Tree (SPT)

16.3.2.1.1 Establishing the RP Tree (Phase 1)

The RP tree is a distribution network that all sources share to deliver multicast data. The root of the RP tree is the Rendezvous Point.

The process starts when a receiver requests multicast data from a group (G). The receiver's DR sends a PIM (*,G) Join message toward the multicast group's RP. As the message travels towards the RP, it instantiates the multicast (*,G) state in each router on the path. Join messages converge on the RP to form the RP tree.

The DR resends Join messages periodically, while it has a receiver in the group, to prevent state timeout expiry in the routers along the path. When all receivers on a DRs subnet leave a group, the DR sends a (*,G) Prune message towards the RP to remove the state from the routers.

A multicast sender transmits multicast data to the RP through its DR. The DR encapsulates the multicast packets and sends them as unicast packets. The RP extracts the native multicast packet and sends it to the RP tree towards the group members.

16.3.2.1.2 Eliminating Multicast Encapsulation (Phase 2)

Data encapsulation, while initially required before the multicast path is established, is inefficient because it requires the transmission of data that is extraneous to multicast. Phase 2 establishes states in the routers that support the transmission of native multicast packets.

When the RP receives an encapsulated packet from source S on group G, it sends an (S,G) join message toward the source. As the message travels towards S, it instantiates the (S,G) state on each router in the path which is used to forward packets from source S destined for group G. Data packets on the (S,G) path are also routed into the RP tree when they encounter an (*,G) router.

When the RP starts receiving native packets from the sources, it sends a Register-Stop message to the sources DR, halting packet encapsulation. At this time, traffic flows natively from the source along a source-specific tree to the RP, then along the shared RP tree to the receivers.

16.3.2.1.3 Establishing the Shortest Path Tree (Phase 3)

The third phase establishes the shortest path from the multicast source to all receivers.

When a multicast packet arrives at the receiver, its router (typically the DR) sends a Join message towards the source to instantiate the (S,G) state in all routers along its path. The message eventually reaches either the sources subnet or a router that already has an (S,G) state. This causes data to flow from the source to the receiver following the (S,G) path. At this time, the receiver is receiving data from the Shortest Path Tree (SPT) and the RP tree (RPT).

The DR (or upstream router) eliminates the data transmission along the RPT by sending a prune message (S,G,rpt) towards the RP. The message instantiates the state on each router in the path, continuing until it reaches the RP or a router that needs traffic from the same source for other receivers.

16.3.2.2 Bidirectional PIM

Bidirectional PIM (Bidir-PIM) builds shared trees, rooted at the rendezvous point (RP), for each multicast group. Because the trees are based only on (*,G) routes, they can accommodate a much larger number of sources without overfilling the MFIB.

In Bidir-PIM, there is no multicast encapsulation or SPT establishment. All packets are natively forwarded toward the RP along shared, bidirectional trees. There are also no designated routers. Instead, a single designated forwarder (DF) is elected on each link to each RP, usually during the RP discovery process. The DF is the router with the shortest route to the RP based on the unicast routing table. It is responsible for forwarding upstream traffic toward the RP and forwarding downstream traffic toward the groups on its link. All routes pass through the RP, and multicast packets are sent from sources toward the RP and to receivers at each hop along the route.

Bidir-PIM elects DFs when a new RP is discovered, when the DF fails, or when there is a change that affects the topology of the link.

16.3.2.3 Rendezvous Points (RP)

In PIM-SM, an RP is a router that is configured as the root of multicast groups distribution tree. These distribution trees are not source-specific. The RP is the destination for both join messages from receivers and data from senders, allowing receivers discover sender identity and begin receiving group traffic. In PIM-SM, paths through RP routers are temporary; when traffic volume reaches a sufficient level, the receiver joins a source-specific tree and the path through the RP is dropped. In Bidir-PIM, all paths pass through the RP, and all packets destined for a given multicast group are forwarded to the RP for that group.

RP addresses in Bidir-PIM must be routable from all sources in the domain, but do not have to correspond to any specific physical interface. Multiple groups can use the same RP for distribution.

The switch supports two methods of mapping RPs to multicast groups:

- **Static**: RPs are statically configured through a CLI statement.
- **Dynamic**: RPs are dynamically selected by a bootstrap router from a set of candidate RPs.

While dynamic RP mappings have priority over static maps by default, a static RP can be configured to override dynamic mappings.

Rendezvous Points (RPs) describes the configuration of rendezvous points.

16.3.3 Configuring PIM

This section describes the following configuration tasks:

- Enabling PIM IPv4 Sparse Mode
- Enabling PIM IPv6 Sparse Mode
- Enabling the S, G Expiry Timer Interval
- Enabling PIM Bidirectional
- Rendezvous Points (RPs)
- Hello Messages
- Hello Hold Time
- Designated Router Election
- Designated Forwarder Election
- Join-Prune Messages
- Legacy PIM Configuration in Global Configuration Mode
- Configuring PIM in a Non-default VRF

16.3.3.1 Enabling PIM IPv4 Sparse Mode

By default, IPv4 PIM is disabled on an interface. The **pim ipv4 sparse-mode** command enables PIM IPv4 Sparse Mode (PIM-SM) on the configuration mode interface. Enabling PIM on an interface enables IGMP on the interface as well.

Example

This command enables IPv4 PIM-SM and IGMP on ***interface vlan 8***.

```
switch(config)# interface vlan 8
switch(config-if-Vl8)# pim ipv4 sparse-mode
switch(config-if-Vl8)#
```

16.3.3.2 Enabling PIM IPv6 Sparse Mode

By default, IPv6 PIM is disabled on an interface. The **pim ipv6 sparse-mode** command enables PIM IPv6 Sparse Mode (PIM-SM) on the configuration mode interface.

Example

This command enables IPv6 PIM-SM on ***interface ethernet 15***.

```
switch(config)# interface ethernet 15
switch(config-if-Et15)# pim ipv6 sparse-mode
switch(config-if-Et15)#
```

16.3.3.3 Enabling the S, G Expiry Timer Interval

The **sg-expiry-timer** command enables expiry timer interval for the PIM-SM multicast routes. By default, thes command applies to the default VRF when the command is issued in the Router-Multicast Configuration mode. During the time of interval, there is no multicast traffic activity on the route.

Example

This command configures ***150*** seconds as the (S,G) expiry timer interval is in the default VRF.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# sg-expiry-timer 150
switch(config-router-pim-sparse-ipv4)#
```

16.3.3.4 Enabling PIM Bidirectional

By default, PIM is disabled on an interface. The **pim ipv4 bidirectional** command enables Bidirectional PIM (Bidir-PIM) on the configuration mode interface. Enabling PIM on an interface also enables IGMP on that interface.

Example

These commands enable Bidir-PIM and IGMP on ***interface vlan 9***.

```
switch(config)# interface vlan 9
switch(config-if-Vl9)# pim ipv4 bidirectional
switch(config-if-Vl9)#
```

16.3.3.5 Rendezvous Points (RPs)

The switch supports dynamic RPs, static RPs, and anycast RPs.

Configuring Static RPs

The **rp address** command configures a static RP, providing an option to override dynamic RPs.

Examples

- This command creates a static RP at ***10.17.255.83*** in the default VRF that maps to all multicast groups (***224/4***) and overrides dynamic RPs.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# rp address 10.17.255.83 override
switch(config-router-pim-sparse-ipv4)#
```

- This command creates a static RP at ***10.21.18.23*** in the default VRF that maps to the multicast groups at ***238.1.12.0/24***.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# rp address 10.21.18.23 238.1.12.0/24
switch(config-router-pim-sparse-ipv4)#
```

Configuring Dynamic RPs

Dynamic RP selection is implemented through a BootStrap Router (BSR), which is a PIM router within the PIM domain that selects RPs from a list of candidates. A subset of PIM routers within the domain are configured as Candidate BootStrap Routers (C-BSRs). Through the exchange of BootStrap Messages (BSMs), the C-BSRs elect the BSR, which then uses BSMs to inform all domain routers of its status.

The BSR holdtime defines the timeout period that an elected BSR remains valid after the receipt of a BSM and is also used in dynamic RP configuration. Holdtime is designated by the BSR router and communicated to other routers through BSMs.

Another subset of domain PIM routers are configured as Candidate RPs (C-RPs). The BSR creates a set of qualifying RPs from the list of C-RPs, then distributes the group-to RP mapping set to all domain routers through BSMs. Each PIM router, after receiving this set, uses a standard algorithm defined in ***RFC 6226*** to select one RP per multicast group.

The **candidate** command configures the switch as a Candidate BSR router (C-BSR). Command parameters specify the switchs BSR address, the interval between BSM transmissions, and the switchs BSR priority rating. Priority ratings range from ***0*** to ***255*** with a default of ***64***. Higher numbers denote higher priority during BSR elections.

Example

These commands configure the switch as a BSR candidate in the default VRF, using the IP address assigned to ***vlan interface 24*** as its BSR address. The BSM transmission interval is set to ***30*** seconds and the priority is set to ***192***.

```
switch(config)# router pim bsr
switch(config-router-pim-bsr)# ipv4
switch(config-router-pim-bsr-ipv4)# candidate vlan 24 priority 192 interval 30
switch(config-router-pim-bsr-ipv4)#
```

The **holdtime** command specifies the value the switch inserts in the ***holdtime*** field of BootStrap Messages (BSMs) that it sends. This value becomes the holdtime for the PIM domain if the switch is elected as the BSR.

Example

These commands specify ***75*** seconds as the value that the switch inserts into BSM holdtime fields in the default VRF.

```
switch(config)# router pim bsr
switch(config-router-pim-bsr)# ipv4
switch(config-router-pim-bsr-ipv4)# holdtime 75
switch(config-router-pim-bsr-ipv4)#
```

The rp candidate command configures the switch as a candidate rendezvous point (C-RP). The BSR selects a multicast groups dynamic RP set from the list of C-RPs. Command parameters specify the switchs RP address, C-RP advertisement interval, and priority rating. The priority rating is used by the BSR when selecting RPs. The C-RP advertisement interval specifies the period between successive C-RP advertisement message transmissions to the BSR.

Running-config may contain multiple **`rp candidate`** statements to support multiple multicast groups:

- All commands must specify the same interface. Issuing a command with an interface that differs from existing commands removes all existing commands from ***running-config***.
- ***Running-config*** stores the **interval** setting in a separate statement that applies to all **rp candidate** statements. Commands that specify an interval that differs from the previously configured value place the new value in ***running-config***. This new value applies to all **rp candidate** statements.

Example

These commands configure a switch as a candidate RP for the multicast group ***235.1.1.0/24***, with a priority of ***48*** and a RP advertisement interval of ***45*** seconds, in the default VRF.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# rp candidate vlan 24 235.1.1.0/24
 priority 48 interval 45
switch(config-router-pim-sparse-ipv4)#
```

By default, the switch transmits BootStrap router Messages (BSMs) over all PIM-enabled interfaces. The **pim bsr ipv4 border** command prevents the switch from transmitting BSMs over the configuration mode interface.

Example

This command prevents the switch from sending BSMs from ***interface vlan 10***.

```
switch(config)# interface vlan 10
switch(config-if-Vl10)# pim bsr ipv4 border
switch(config-if-Vl10)#
```

Anycast Rendezvous Points

A PIM anycast Rendezvous Point (anycast RP) defines a single RP address that exists on multiple devices. An anycast-RP set consists of the routers configured with the same anycast-RP address. An anycast RP provides redundancy protection and load balancing. The anycast-RP set supports all multicast groups.

The **anycast-rp** command configures the switch as a member of an anycast-RP set and establishes a communication link with another member of the set.

Example

These commands configure a switch (IP address ***10.1.1.14***) into an anycast-RP set with an RP address of ***10.17.255.2*** in the default VRF. The anycast-RP set contains three other routers, located at 1***0.1.2.14***, ***10.1.3.14***, and ***10.1.4.14***. It sets the number of unacknowledged register messages it sends to each router at ***15***.

```
switch(config)# router pim sparse-mode
```

```
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# anycast-rp 10.17.255.2 10.1.1.14
 register-count 15
switch(config-router-pim-sparse-ipv4)# anycast-rp 10.17.255.2 10.1.2.14
 register-count 15
switch(config-router-pim-sparse-ipv4)# anycast-rp 10.17.255.2 10.1.3.14
 register-count 15
switch(config-router-pim-sparse-ipv4)# anycast-rp 10.17.255.2 10.1.4.14
 register-count 15
switch(config-router-pim-sparse-ipv4)#
```

16.3.3.6 Hello Messages

PIM-SM multicast routers send PIM router query messages (hello messages) to elect a Designated Router (DR) for each subnet. The DR then sends registration messages to the RP.

The **pim ipv4 hello interval** command specifies the transmission interval between PIM hello messages originating from the specified VLAN interface.

Example

This command configures ***45*** second intervals between hello messages originating from ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# pim ipv4 hello interval 45
switch(config-if-Vl4)#
```

16.3.3.7 Hello Hold Time

A PIM interface maintains a hold timer for each of its neighbors. The timer is reset whenever a hello message is received from the neighbor. When the timer expires, the neighbor is considered DOWN. The PIM interface can advertise its neighbor to use a higher hold time by modifying the hello interval, or by setting a higher hello count using the **pim ipv4 hello count** command. The hello count specifies how many hello messages can be missed before the neighbor is considered down; the hold time is therefore the hello interval multiplied by the hello count.

Examples

- This command configures the PIM hold time on ***interface vlan2925*** to ***225*** seconds (7.5 times the 30-second hello interval).

```
switch(config)# interface vlan2925
switch(config-if-Vl2925)# pim ipv4 hello count 7.5
```

- This command displays the hold time on ***interface vlan2925***.

```
switch# show ip pim interface vlan2925 details
Interface Vlan2925 address is 1.0.1.1
Vif number is 0
PIM: enabled
PIM version: 2, mode: sparse
PIM neighbor count: 0
PIM Effective DR: 1.0.1.1 (this system)
PIM Effective DR Priority: 1
PIM Effective Propagation Delay: 500 milliseconds
PIM Effective Override Interval: 2500 milliseconds
PIM Effective Tracking Support: disabled
PIM Hello Interval: 30 seconds
PIM Hello Hold Time: 225 seconds <====== New Hold Time ( = 7.5 * 30 )
PIM Hello Priority: 1 seconds
PIM Hello Lan Delay: 500 milliseconds
```

```
PIM Assert Override Interval: 3 seconds switch#
```

16.3.3.8 Designated Router Election

PIM-SM uses these criteria for electing a Designated Router (DR):

- If at least one router does not advertise a DR priority value, then PIM-SM elects the router with the highest IP address as the DR.
- If all routers advertise a DR priority value, then PIM-SM elects the router with the highest DR priority value as the DR.

The **group-expiry-timer** command sets the DR priority value that the switch advertises. If ***running-config*** does not contain a **pim ipv4 dr-priority** statement, the switch does not advertise a DR priority value.

Examples

- This command configures a DR priority value of ***15*** on ***interface vlan 4***.

```
switch(config-if-Vl4)# pim ipv4 dr-priority 15
switch(config-if-Vl4)#
```

- This command removes the DR priority from ***interface vlan 4***.

```
switch(config-if-Vl4)# no pim ipv4 dr-priority
switch(config-if-Vl4)#
```

16.3.3.9 Designated Forwarder Election

Designated Forwarders (DFs) are elected based on route metrics in the unicast routing table; there are no configuration options that affect the selection of DFs.

16.3.3.10 Join-Prune Messages

Join/prune messages are sent by the PIM-SM Designated Router (DR) or the Bidir-PIM Designated Forwarder (DF) toward the Rendezvous Point (RP). These messages inform other PIM routers about clients that want to become receivers (join) or stop being receivers (prune) for the groups.

The **pim ipv4 join-prune interval** command specifies the period between join/prune messages that the switch originates from the specified VLAN interface and sends to the upstream RPF neighbor.

Example

This command configures ***75*** second intervals between join/prune messages originating from ***interface vlan 4***.

```
switch(config-if-Vl4)# pim ipv4 join-prune interval 75
switch(config-if-Vl4)#
```

16.3.3.11 Legacy PIM Configuration in Global Configuration Mode

Earlier versions of the EOS managed all non-interface-specific PIM configuration from Global Configuration Mode. Legacy configurations retain these global commands in ***running-config*** after upgrading to a newer version of the EOS, and the configurations are applied unchanged to the default VRF. PIM configuration commands entered in ***global*** onfiguration mode which can be applied to either PIM-SM or Bidir-PIM will be applied to PIM-SM. If any commands are added to ***running-config*** using the new configuration modes, all legacy commands will be converted to the new modal commands and applied in the default VRF.



Note: Multicast configuration commands issued in ***global*** configuration mode are now deprecated, and Arista recommends configuring all multicast parameters in the appropriate configuration mode

(i.e., ***Router-Multicast*** Configuration, ***Router-PIM Sparse-mode*** Configuration, ***Router-PIM Bidirectional*** Configuration, ***Router-PIM BSR*** Configuration, or ***Router-MSDP*** Configuration).

16.3.3.12 Configuring PIM in a Non-default VRF

For PIM to function in a non-default VRF, the VRF must be created and configured for multicast traffic, and routed ports must be added to the VRF. Once this is accomplished, configure VRF-global PIM parameters by using the **`vrf`** command within a PIM configuration mode to place the switch in a PIM VRF configuration submode.

Interface-specific PIM parameters are configured in the ***interface-configuration*** mode for VRF-member interface.

Legacy multicast routing commands issued in ***global*** configuration mode are applied to the default VRF, but are now deprecated and are available only for backward compatibility. If any PIM commands are issued in the new format, all legacy commands remaining in ***running-config*** will be replaced with their updated equivalents and applied to the default VRF.

16.3.3.12.1 Preparing the VRF for PIM Configuration

The following steps prepare a non-default VRF to use PIM:

1. Enable unicast routing in the default VRF.

```
switch(config)# ip routing vrf default
```

2. Create the non-default VRF if not already created.

```
switch(config)# vrf instance purple
```

3. Enable enable unicast routing on the new VRF.

```
switch(config-vrf-purple)# exit
switch(config)# ip routing vrf purple
```

4. Add participating routed ports to the new VRF.

```
switch(config)# interface ethernet 9/2-9/4
switch(config-if-Et9/2-4)# no switchport
switch(config-if-Et9/2-4)# vrf purple
switch(config-if-Et9/2-4)# exit
```

5. Enable multicast routing on the new VRF.

```
switch(config)# router multicast
switch(config-router-multicast)# vrf purple
switch(config-router-multicast-vrf-purple)# ipv4
switch(config-router-multicast-vrf-purple-ipv4)# routing
```

16.3.3.12.2 Configuring Global PIM Parameters in a Non-default VRF

Global PIM parameters for non-default VRFs are configured in the VRF submode of the appropriate PIM configuration mode.

Example

These commands configure a switch as a candidate RP for the multicast group ***235.1.1.0/24***, with a priority of ***48*** and a RP advertisement interval of ***45*** seconds, in VRF ***purple***.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# vrf purple
```

```
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-vrf-purple-ipv4)# rp candidate vlan 24
 235.1.1.0/24 priority 48 interval 45
switch(config-router-pim-sparse-vrf-purple-ipv4)#
```

16.3.3.12.3 Configuring Interface-specific PIM Parameters in a Non-default VRF

Interface-specific PIM parameters for member interfaces of a non-default VRF are configured just as they are for the default VRF: in the ***interface-configuration*** mode for the interface.

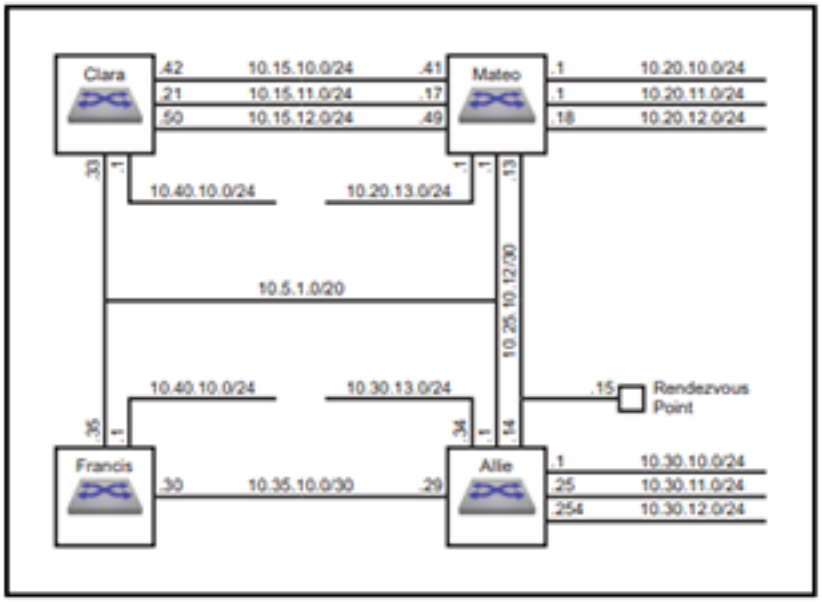
16.3.4 Multicast Example

This section provides an example network that implements multicast (PIM-SM) in the default VRF and includes the required commands.

16.3.4.1 Diagram

Figure 67: Multicast Example displays the multicast network example. The network contains four routers. Multicast routing (PIM-SM) is enabled on two switches. One switch has its IGMP Snooping Querier enabled.

Figure 16-4: Multicast Example



The example multicast network implements these multicast parameters:

Rendezvous Point Address: 10.25.10.15

Switch Clara

- IGMP Snooping: disabled
- Subnet Summary:
  - ***10.40.10.0/24***: ***vlan 11***
  - ***10.15.10.0/24***: ***vlan 12***
  - ***10.15.11.0/24***: ***vlan 13***
  - ***10.15.12.0/24***: ***vlan 14***
  - ***10.5.1.0/20***: ***vlan 10***

Switch Mateo

- IGMP Snooping: disabled
- Subnet Summary:
  - ***10.20.13.0/24***: ***vlan 18***
  - ***10.20.10.0/24***: ***vlan 15***
  - ***10.20.11.0/24***: ***vlan 16***
  - ***10.20.12.0/24***: ***vlan 17***
  - ***10.15.10.0/24***: ***vlan 12***
  - ***10.15.11.0/24***: ***vlan 13***
  - ***10.15.12.0/24***: ***vlan 14***
  - ***10.25.10.12/30***: ***vlan 19***
  - ***10.5.1.0/20***: ***vlan 10***

Switch Allie

- IGMP Snooping: enabled
- Multicast Routing: enabled
- Querier: enabled
- Rendezvous Point Address: ***10.25.10.15***
- MFIB activity polling interval: ***5*** second
- Subnet Summary:
  - ***10.30.13.0/24***: ***vlan 23***
  - ***10.30.10.0/24***: ***vlan 20*** PIM-SM enabled
  - ***10.30.11.0/24***: ***vlan 21*** PIM-SM enabled
  - ***10.30.12.0/24***: ***vlan 22***
  - ***10.25.10.12/30***: ***vlan 19***
  - ***10.35.10.0/30***: ***vlan 24*** PIM-SM enabled
  - ***10.5.1.0/20***: ***vlan 10*** PIM-SM enabled

Switch Francis

- IGMP Snooping: enabled
- Multicast Routing: enabled
- Subnet Summary:
  - ***10.40.10.0/24***: ***vlan 25*** PIM-SM enabled
  - ***10.35.10.0/30***: ***vlan 24*** PIM-SM enabled
  - ***10.5.1.0/20***: ***vlan 10***

16.3.4.2 Example

This example configures PIM-SM.

1. Configure the interface addresses.
   a. Router Clara interfaces.

```
Clara(config)# interface vlan 11
Clara(config-if-vl11)# ip address 10.40.10.1/24
Clara(config-if-vl11)# interface vlan 12
Clara(config-if-vl12)# ip address 10.15.10.42/24
Clara(config-if-vl12)# interface vlan 13
Clara(config-if-vl13)# ip address 10.15.11.21/24
Clara(config-if-vl13)# interface vlan 14
```

```
Clara(config-if-vl14)# ip address 10.15.12.50/24
Clara(config-if-vl14)# interface vlan 10
Clara(config-if-vl10)# ip address 10.5.1.33/20
Clara(config-if-vl10)# router ospf 1
Clara(config-router-ospf)# redistribute static
```

b. Router Mateo interfaces.

```
Mateo(config)# interface vlan 18
Mateo(config-if-vl18)# ip address 10.20.13.1/24
Mateo(config-if-vl18)# interface vlan 15
Mateo(config-if-vl15)# ip address 10.20.10.1/24
Mateo(config-if-vl15)# interface vlan 16
Mateo(config-if-vl16)# ip address 10.20.11.1/24
Mateo(config-if-vl16)# interface vlan 17
Mateo(config-if-vl17)# ip address 10.20.12.16/24
Mateo(config-if-vl17)# interface vlan 12
Mateo(config-if-vl12)# ip address 10.15.10.41/24
Mateo(config-if-vl12)# interface vlan 13
Mateo(config-if-vl13)# ip address 10.15.11.17/24
Mateo(config-if-vl13)# interface vlan 14
Mateo(config-if-vl14)# ip address 10.15.12.49/24
Mateo(config-if-vl14)# interface vlan 19
Mateo(config-if-vl19)# ip address 10.25.10.13/30
Mateo(config-if-vl19)# interface vlan 10
Mateo(config-if-vl10)# ip address 10.5.1.1/20
Mateo(config-if-vl10)# router ospf 1
Mateo(config-router-ospf)# redistribute static
```

c. Router Allie interfaces.

```
Allie(config)# interface vlan 23
Allie(config-if-vl23)# ip address 10.30.13.34/24
Allie(config-if-vl23)# interface vlan 20
Allie(config-if-vl20)# ip address 10.30.10.1/24
Allie(config-if-vl20)# interface vlan 21
Allie(config-if-vl21)# ip address 10.30.11.25/24
Allie(config-if-vl21)# interface vlan 22
Allie(config-if-vl22)# ip address 10.30.12.254/24
Allie(config-if-vl22)# interface vlan 19
Allie(config-if-vl19)# ip address 10.25.10.14/30
Allie(config-if-vl19)# interface vlan 24
Allie(config-if-vl24)# ip address 10.35.10.29/30
Allie(config-if-vl24)# interface vlan 10
Allie(config-if-vl10)# ip address 10.5.1.1/20
Allie(config-if-vl10)# router ospf 1
Allie(config-router-ospf)# redistribute static
```

d. Router Francis interfaces.

```
Francis(config)# interface vlan 25
Francis(config-if-vl25)# ip address 10.40.10.1/24
Francis(config-if-vl25)# interface vlan 24
Francis(config-if-vl24)# ip address 10.35.10.30/24
Francis(config-if-vl24)# interface vlan 10
Francis(config-if-vl10)# ip address 10.5.1.35/24
Francis(config-if-vl10)# router ospf 1
Francis(config-router-ospf)# redistribute static
```

2. Configure the interface multicast parameters.

a. Router Allie interfaces.

```
Allie(config-router-ospf)# interface vlan 20
Allie(config-if-vl20)# pim ipv4 sparse-mode
Allie(config-if-vl20)# interface vlan 21
Allie(config-if-vl21)# pim ipv4 sparse-mode
Allie(config-if-vl21)# interface vlan 24
Allie(config-if-vl24)# pim ipv4 sparse-mode
Allie(config-if-vl24)# interface vlan 10
Allie(config-if-vl10)# pim ipv4 sparse-mode
```

b. Router Francis interfaces.

```
Francis(config-router-ospf)# interface vlan 25
Francis(config-if-vl25)# pim ipv4 sparse-mode
Francis(config-if-vl25)# interface vlan 24
Francis(config-if-vl24)# pim ipv4 sparse-mode
```

3. Configure the router multicast parameters.

a. Router Clara parameters.

```
Clara(config-router-ospf)# exit
Clara(config)# no ip igmp snooping
```

b. Router Mateo router.

```
Mateo(config-router-ospf)# exit
Mateo(config)# no ip igmp snooping
```

c. Router Allie router.

```
Allie(config-if-vl10)# exit
Allie(config)# router multicast
Allie(config-router-multicast)# ipv4
Allie(config-router-multicast-ipv4)# routing
Allie(config-router-multicast-ipv4)# activity polling-interval 5
Allie(config-router-multicast-ipv4)# router pim sparse-mode
Allie(config-router-pim-sparse)# ipv4
Allie(config-router-pim-sparse-ipv4)# rp address 10.25.10.15
```

d. Router Francis router.

```
Francis(config-if-vl24)# exit
Francis(config)# router multicast
Francis(config-router-multicast)# ipv4
Francis(config-router-multicast-ipv4)# routing
Francis(config-router-multicast-ipv4)# router pim sparse-mode
Francis(config-router-pim-sparse)# ipv4
Francis(config-router-pim-sparse-ipv4)# rp address 10.25.10.15
```

16.3.5 PIM Commands

PIM Configuration Commands (Global)

- anycast-rp
- candidate
- fast-reroute
- group-expiry-timer

- holdtime
- ip pim dr-notify-delay
- log neighbors
- register local-interface
- router pim bidirectional
- router pim bsr
- router pim sparse-mode
- rp address
- rp allow
- rp candidate
- rp-candidate advertisement-filter
- rp hash algorithm modulo
- sg-expiry-timer
- spt threshold
- ssm range

PIM Configuration Commands (Interface)

- ipv4
- pim bsr ipv4 border
- pim ipv4 bidirectional
- pim ipv4 border-router
- pim ipv4 dr-priority
- pim ipv4 hello count
- pim ipv4 hello interval
- pim ipv4 join-prune count
- pim ipv4 join-prune interval
- pim ipv4 neighbor-filter
- pim ipv4 sparse-mode
- pim ipv6 sparse-mode
- sztimeout

PIM Display Commands

- show ip pim bsr
- show ip pim config-sanity
- show ip pim interface
- show ip pim neighbor
- show ip pim protocol counters
- show ip pim register-source
- show ip pim rp
- show ip pim rp-candidate
- show ip pim rp-hash
- show ip pim upstream joins

16.3.5.1 anycast-rp

The **`anycast-rp`** command configures the switch as a member of an anycast-RP set and establishes a communication link with another member of the set.

When the command is issued in ***router-multicast ipv4*** configuration mode it applies to the default VRF; to use this command in a non-default VRF, issue it in Router-Multicast ***router-multicast ipv4*** configuration mode.

The **no anycast-rp** and **default anycast-rp** commands remove the corresponding **anycast-rp** command from ***running-config***. When the **no** and **default** commands do not include a peer address, all commands for the specified RP address are removed.

Command Mode

Router-Multicast IPv4 Configuration

Router-Multicast VRF IPv4 Configuration

Command Syntax

anycast-rp ***rp_addr peer_addr*** [**REGISTER**]

no anycast-rp ***rp_addr*** [***peer_addr***]

default anycast-rp ***rp_addr*** [***peer_addr***]

Parameters

- ***rp_addr*** Rendezvous point IP address (dotted decimal notation).
- ***peer_addr*** IP address of another anycast-RP set member (dotted decimal notation). Note, the peer-addr for a local anycast RP must be a ***/32*** address configured under a loopback interface. To enable anycast RP on the router, each RP requires one local peer address. Use the **show ip pim anycast-rp [vrf <vrfName>]** command to verify the configured local and remote peers.
- **REGISTER** Number of unacknowledged register messages the switch sends to the peer router.
  - ***no parameter*** register count is set to default value of ***10***.
  - **register-count** ***r_num*** where ***r_num*** is an integer that ranges from *1* to ***4294967295***.
  - **register-count infinity**

Example

These commands configure a switch (IP address ***10.1.1.14***) into an anycast-RP set with an RP address of ***10.17.255.2*** in the default VRF. The anycast-RP set contains three other routers, located at ***10.1.2.14***, ***10.1.3.14***, and ***10.1.4.14***. It sets the number of unacknowledged register messages it sends to each router at ***15***.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# anycast-rp 10.17.255.2 10.1.1.14
 register-count 15
switch(config-router-pim-sparse-ipv4)# anycast-rp 10.17.255.2 10.1.2.14
 register-count 15
switch(config-router-pim-sparse-ipv4)# anycast-rp 10.17.255.2 10.1.3.14
 register-count 15
switch(config-router-pim-sparse-ipv4)# anycast-rp 10.17.255.2 10.1.4.14
 register-count 15
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.2 candidate

The **candidate** command configures the switch as a Candidate BSR router (C-BSR). A BSR is a PIM router within the PIM domain through which dynamic RP selection is implemented. The BSR selects RPs from a list of candidate RPs and exchange BootStrap Messages (BSM) with all routers in the domain. The BSR is elected from one of the C-BSRs through an exchange of BSMs.

A subset of PIM routers within the domain are configured as Candidate BootStrap Routers (C-BSRs). Through the exchange of BootStrap Messages (BSMs), the C-BSRs elect the BSR, which then uses BSMs to inform all domain routers of its status.

Command parameters specify the switchs BSR address, the interval between BSM transmissions, the length of the hash mask, and the priority assigned to the switch when electing a BSR.

Entering an **candidate** command replaces any previously configured **candidate** command. If the new command does not specify a priority, hash mask length, or interval, the previously configured values persist in ***running-config***.

When the command is issued in the ***router-pim bsr ipv4*** configuration mode it applies to the default VRF; to use this command in a non-default VRF, issue it in ***router-pim bsr vrf ipv4*** configuration mode for the appropriate VRF.

The **no candidate** and **default candidate** commands remove the corresponding **candidate** commands from ***running-config***. The **no** and **default** commands restore the priority, hash mask length, and interval parameters to their default values.

Command Mode

Router-PIM BSR IPv4 Configuration

Router-PIM BSR VRF IPv4 Configuration

Command Syntax

candidate INTERFACE [**HASHMASK_LENGTH**][**INTERVAL_PERIOD**][**PRIORITY_NUM**]

no candidate [***priority***][***interval***]

default candidate[***priority***][***interval***]

Parameters

- **INTERFACE** Switch uses IP address of specified interface as its BSR address. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
- **HASHMASK_LENGTH** Length (in bits) of the hash mask.
  - ***no parameter*** hash mask remains unchanged from previous setting.
  - **hashmask *0 - 32*** hash mask length (in bits). Default value is ***30***.
- **INTERVAL_PERIOD** Period between the transmission of BSMs (seconds). Default value is ***60***.
  - ***no parameter*** interval remains unchanged from previous setting.
  - **interval *10 - 536870906*** transmission interval in seconds.
- **PRIORITY_NUM** BSR election priority rating. Larger numbers denote higher priority. Default value is ***64***.
  - ***no parameter*** priority remains unchanged from previous setting.
  - **priority *0 - 255*** priority rating.

Example

These commands configure the switch as a BSR candidate in the default VRF, using the IP address assigned to ***interface vlan 24*** as its BSR address. The BSM transmission interval is set to ***30*** seconds and the priority is set to ***192***.

```
switch(config)# router pim bsr
switch(config-router-pim-bsr)# ipv4
switch(config-router-pim-bsr-ipv4)# candidate vlan 24 priority 192 interval 30
switch(config-router-pim-bsr-ipv4)#
```

16.3.5.3 fast-reroute

The **fast-reroute** command enables Multicast only Fast Re-Route (MoFRR) to minimize traffic loss in a network when a link or node failure occurs. Traffic loss is minimized by allowing the traffic to flow from the secondary path upon the failure of the primary path.

The **no fast-reroute** and **default fast-reroute** commands disable MoFRR by removing the corresponding fast-reroute command from ***running-config***.

Command Mode

Router-PIM BSR IPv4 Configuration

Router-PIM BSR VRF IPv4 Configuration

Command Syntax

fast-reroute ***acl_name***

no fast-reroute ***acl_name***

default fast-reroute ***acl_name***

Parameters

acl_name standard access list name.

Examples

- These commands enable fast reroute for ACL ***acl2*** in the default VRF under the IPv4 configuration.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# fast-reroute acl2
switch(config-router-pim-sparse-ipv4)#
```

- These commands enable fast reroute for ACL ***acl2*** in ***vfr red*** under the IPv4 configuration.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# vrf red
switch(config-router-pim-sparse-vrf-red)# ipv4
switch(config-router-pim-sparse-vrf-red-ipv4)# fast-reroute acl2
switch(config-router-pim-sparse-vrf-red-ipv4)#
```

16.3.5.4 group-expiry-timer

The **group-expiry-timer** command sets the group-expiry-timer in seconds after which a group with no activity gets deleted from the PIM Rendezvous Point (RP) tree.

When the command is configured in the ***global*** configuration mode, the configuration is applied globally on the switch. To apply the configuration only in the ***bidir-pim*** mode, the command is configured in the Router-PIM Bidirectional configuration mode. To apply the configuration for a specific VRF, the command is configured in the VRF sub-mode of the Router-PIM Bidirectional configuration mode. Use the **pim ipv4 bidirectional** command to enter Router-PIM Bidirectional Configuration Mode.

The **no group-expiry-timer** and **default group-expiry-timer** applies the system default configuration and removes the corresponding **group-expiry-timer** command from ***running-config***.

Command Mode

Router-PIM Bidirectional IPv4 Configuration

Router-PIM Bidirectional VRF IPv4 Configuration

Command Syntax

group-expiry-timer ***value***

no group-expiry-timer ***value***

default group-expiry-timer ***value***

Parameter

value specifies the time in seconds after which a group with no activity expires from the PIM RP. Values range from ***1*** to ***210***. There is no default value.

Examples

- This command configures PIM expiry-timer of ***40*** seconds in ***pim-bidirectional*** sub-mode.

```
switch(config)# router pim bidirectional
switch(config-router-pim-bidir)# ipv4
switch(config-router-pim-bidir-ipv4)# group-expiry-timer 40
```

- This command configures PIM expiry-timer of ***120*** seconds for ***vrf v1***.

```
switch(config)# router pim bidirectional
switch(config-router-pim-bidir)# vrf v1
switch(config-router-pim-bidir-vrf-v1)# ipv4
switch(config-router-pim-bidir-vrf-v1-ipv4)# group-expiry-timer 120
```

16.3.5.5 holdtime

The **holdtime** command specifies the value the switch inserts in the ***holdtime*** parameter field in BootStrap Messages (BSM) that it sends. The BSR holdtime defines the timeout period that an elected BSR remains valid after the receipt of a BSM and is also used in dynamic RP configuration. BSR holdtime is designated by the BSR router and communicated to other routers through BSMs.

When the command is issued in ***router-pim bsr ipv4*** configuration mode it applies to the default VRF; to use this command in a non-default VRF, issue it in ***router-pim bsr vrf ipv4*** configuration mode for the appropriate VRF.

The **no holdtime** and **default holdtime** commands restore the default holdtime parameter field insertion value of ***130*** seconds by removing the **holdtime** statement from ***running-config***.

Command Mode

Router-PIM BSR IPv4 Configuration

Router-PIM BSR VRF IPv4 Configuration

Command Syntax

holdtime ***period***

no holdtime

default holdtime

Parameter

period BSR holdtime (seconds). Value ranges from ***12*** to ***1073741823*** (1.073 billion seconds, approximately 34 years). Default is ***130***.

Example

These commands specify ***75*** seconds as the value that the switch inserts into BSM holdtime fields in the default VRF.

```
switch(config)# router pim bsr
switch(config-router-pim-bsr)# ipv4
switch(config-router-pim-bsr-ipv4)# holdtime 75
```

```
switch(config-router-pim-bsr-ipv4)#
```

16.3.5.6 ip pim dr-notify-delay

The **ip pim dr-notify-delay** command configures the designated router (DR) notification delay time. (This is equivalent to issuing the **dr-notify-delay** command in Router-PIM Sparse-mode Configuration Mode.) The command is more effective when all PIM routers on the LAN segment have PIM DR priority that is greater than ***1***.

The **no ip pim dr-notify-delay** and **default ip pim dr-notify-delay** commands remove the previously configured DR notification delay time.

Note: while this command is VRF-aware, delay time is not configurable per-VRF. The configuration will apply to all VRFs configured with PIM.

Command Mode

Global Configuration

Command Syntax

ip pim dr-notify-delay ***notify_delay_time***

no ip pim dr-notify-delay

default ip pim dr-notify-delay

Parameter

notify_delay_time The PIM designated router notify delay time in seconds. Values range from -32767 to 32768.

Guidelines

The notification delay time can be configured with a positive or negative value. The timer influences DR election timing when a router with the highest DR priority on a LAN segment is reloaded. In an MLAG configuration, the notification delay time begins shortly after the MLAG reload delay expires (before which the PIM hello messages are sent with a priority of ***1***). In a non-MLAG configuration, the notification delay time begins as soon as the PIM is first configured on the interface.

Positive values for notify delay time cause the device to send PIM hello messages with a priority of ***1*** until the time the notify delay time expires. During this time, DR responsibilities of the device will continue according to configured DR priority. Negative values configured for the notification delay time will not modify the priority sent in PIM hello messages, but the device will not perform any DR responsibility until the notify delay time expires. Positive values are used to avoid loss of multicast packets, but they may create a few duplicate packets from multiple PIM routers forwarding traffic for the same S,G. Negative values are used to avoid duplicate packets, but they may cause packet loss when there are no PIM routers forwarding traffic for an S,G.

Example

This command configures a DR notification delay time of 2 seconds.

```
switch(config)# ip pim dr-notify-delay 2
switch(config)#
```

16.3.5.7 ipv4

The **ipv4** command places the switch in the IPv4 submode for the PIM configuration mode in which it is entered.

Command Mode

Router Multicast Configuration

Router-PIM Bidirectional Configuration

Router-PIM BSR Configuration

Router-PIM Sparse-mode Configuration

Command Syntax

ipv4

Examples

- These commands place the switch in the ***router multicast ipv4*** configuration mode.

```
switch(config)# router multicast
switch(config-router-multicast)# ipv4
switch(config-router-multicast-ipv4)#
```

- These commands place the switch in the ***router-pim bidirectional ipv4*** configuration mode.

```
switch(config)# router pim bidirectional
switch(config-router-pim-bidir)# ipv4
switch(config-router-pim-bidir-ipv4)#
```

- These commands place the switch in the ***router-pim bsr ipv4*** configuration mode.

```
switch(config)# router pim bsr
switch(config-router-pim-bsr)# ipv4
switch(config-router-pim-bsr-ipv4)#
```

- These commands place the switch in the ***router-pim sparse-mode ipv4*** configuration mode.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.8 log neighbors

The **log neighbors** command configures the switch to generate a log message when a neighbor entry is added or removed from the PIM Neighbor table. This function is enabled by default.

The **no log neighbors** command disables log message generation based on changes to the PIM Neighbor table; this command is stored in the ***running-config***. The **log neighbors** and **default log neighbors** commands restore the default setting of generating log messages by deleting the **no log neighbors** statement from ***running-config***.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Router-PIM Bidirectional IPv4 Configuration

Router-PIM Bidirectional VRF IPv4 Configuration

Command Syntax

log neighbors

no log neighbors

default log neighbors

Examples

- These commands configure the switch to stop generating log messages based on PIM Neighbor table changes in the default VRF.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# no log neighbors
switch(config-router-pim-sparse-ipv4)#
```

- These commands configure the switch to generate log messages when a neighbor entry is added or removed from the PIM Neighbor table in the default VRF.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# log neighbors
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.9 pim bsr ipv4 border

The **pim bsr ipv4 border** command prevents the switch from sending BootStrap router Messages (BSMs) over the configuration mode interface. By default, BSMs are transmitted over all PIM-enabled interfaces.

The **no pim bsr ipv4 border** and **default pim bsr ipv4 border** commands restore the transmission of BSMs over the configuration mode interface by removing the corresponding **pim bsr ipv4 border** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

pim bsr ipv4 border

no pim bsr ipv4 border

default pim bsr ipv4 border

Example

This command prevents the switch from sending BSMs from ***interface vlan 10***.

```
switch(config)# interface vlan 10
switch(config-if-Vl10)# pim bsr ipv4 border
switch(config-if-Vl10)#
```

16.3.5.10 pim ipv4 bidirectional

The **pim ipv4 bidirectional** command enables PIM bidirectional and IGMP (router mode) on the configuration mode interface.



Note: PIM and Multicast Border Router (MBR) must be mutually exclusive on an interface. If the interface is configured as an MBR, do not enable PIM on the interface.

The **no pim ipv4 bidirectional**, **no pim ipv4**, **default pim ipv4 bidirectional**, and **default pim ipv4** commands restore the default PIM and IGMP (router mode) settings of ***disabled*** on the configuration mode interface by removing the **pim ipv4 bidirectional** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`pim ipv4 bidirectional`

`no pim ipv4`

`no pim ipv4 bidirectional`

`default pim ipv4`

`default pim ipv4 bidirectional`

Example

This command enables PIM bidirectional on ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# pim ipv4 bidirectional
switch(config-if-Vl4)#
```

16.3.5.11 pim ipv4 border-router

The **`pim ipv4 border-router`** command configures the configuration mode interface as a PIM Multicast Border Router (MBR). A PIM MBR interface allows multicast traffic from sources that are outside of the PIM domain.

This command does not control the transmission or reception of PIM protocol packets by the interface.

Sources learned through an MBR interface are treated as local sources (directly connected to the switch). The border-bit is set in all PIM register messages sent for these sources.

Note: Configuration as an MBR and configuration in PIM sparse mode must be mutually exclusive. Ensure that PIM sparse mode is not configured by issuing the **pim ipv4 sparse-mode** command on the interface before issuing this command.

The **`no pim ipv4 border-router`** and **`default pim ipv4 border-router`** commands removes the PIM MBR configuration for the configuration mode interface by removing the corresponding **`pim ipv4 border-router`** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`pim ipv4 border-router`

`no pim ipv4 border-router`

`default pim ipv4 border-router`

Example

These commands configure ***interface vlan 200*** as a PIM MBR, then display its status.

```
switch(config)# interface vlan 200
switch(config-if-VL200)# ip address 10.44.2.1/24
switch(config-if-VL200)# no pim ipv4 sparse-mode
```

```
switch(config-if-VL200)# pim ipv4 border-router
switch(config-if-VL200)# show active
interface Vlan200
ip address 10.44.2.1/24
pim ipv4 border-router
switch(config-if-VL200)#exit
switch(config)# show ip pim interface
AddressInterfaceModeNeighborHello DRDR
AddressPktsQedPktsDropped
CountIntvl Pri
10.44.2.1Vlan200mbr030110.44.2.100
switch(config)#
```

16.3.5.12 pim ipv4 dr-priority

PIM-SM uses these criteria for electing a Designated Router (DR):

- If at least one router does not advertise a DR priority value, then PIM-SM elects the router with the highest IP address as the DR.
- If all routers advertise a DR priority value, then PIM-SM elects the router with the highest DR priority value as the DR.

The **pim ipv4 dr-priority** command sets the DR priority value that the configuration mode interface advertises. By default, the interface does not advertise a DR priority value.

The **no pim ipv4 dr-priority** and **default pim ipv4 dr-priority** commands force the use of IP addresses to elect the designated router by removing the corresponding **pim ipv4 dr-priority** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

pim ipv4 dr-priority ***level***

no pim ipv4 dr-priority [***level***]

default pim ipv4 dr-priority [***level***]

Parameters

level DR selection priority rating. Value ranges from ***0*** to ***4294967295***.

Examples

- This command configures the dr-priority value of ***15*** on ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# pim ipv4 dr-priority 15
switch(config-if-Vl4)#
```

- This command force the use of IP addresses to elect the designated router.

```
switch(config-if-Vl4)# no pim ipv4 dr-priority
switch(config-if-Vl4)#
```

16.3.5.13 pim ipv4 hello count

The **pim ipv4 hello count** command sets the PIM hello count for the interface being configured. PIM hold time is calculated by multiplying the configured hello interval by the hello count, ensuring that the PIM neighbor stays up for the specified time after which the neighbor expires.

The no **pim ipv4 hello count** command removes the corresponding **pim ipv4 hello count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

pim ipv4 hello count [***multiple***]

no pim ipv4 hello count [***multiple***]

Parameter

multiple hello count multiplier. Value ranges from ***1.5*** to ***65535***. The hello hold time is the configured hello interval multiplied by the hello count.

Examples

- This command configures a hold time interval of ***225*** seconds on ***interface vlan2925*** by multiplying the default ***30***-second hello interval by a hello count of ***7.5***.

```
switch(config)# interface vlan2925
switch(config-if-Vl2925)# pim ipv4 hello count 7.5
switch(config-if-Vl2925)#
```

- This show command displays the hold time and other configuration details on ***interface vlan2925***.

```
switch# show ip pim interface vlan2925 details
Interface Vlan2925 address is 1.0.1.1
Vif number is 0
PIM: enabled
PIM version: 2, mode: sparse
PIM neighbor count: 0
PIM Effective DR: 1.0.1.1 (this system)
PIM Effective DR Priority: 1
PIM Effective Propagation Delay: 500 milliseconds
PIM Effective Override Interval: 2500 milliseconds
PIM Effective Tracking Support: disabled
PIM Hello Interval: 30 seconds
PIM Hello Hold Time: 225 seconds <====== New Hold Time (= 7.5 * 30)
PIM Hello Priority: 1 seconds
PIM Hello Lan Delay: 500 milliseconds
PIM Assert Override Interval: 3 seconds mrtr1#
```

16.3.5.14 pim ipv4 hello interval

The **pim ipv4 hello interval** command specifies the transmission interval between PIM hello messages originating from the configuration mode interface.

The **no pim ipv4 hello interval** and **default pim ipv4 hello interval** commands restore the default query interval of ***30*** seconds for the configuratiom mode interface by removing the corresponding **pim ipv4 hello interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

pim ipv4 hello interval ***period***

no pim ipv4 hello interval [***period***]

default pim ipv4 hello interval [***period***]

Parameter

period query interval (seconds). Value ranges from ***1*** to ***1000000*** (1 million). Default is ***30***.

Example

This command configures ***45*** second intervals between hello messages originating from ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# pim ipv4 hello interval 45
switch(config-if-Vl4)#
```

16.3.5.15 pim ipv4 join-prune count

The **pim ipv4 join-prune count** command configures the number of times a join or prune messages can be missed before the upstream neighbor time expires.

The join-prune interval multiplied by the count is considered as join or prune hold time (specified in seconds), which is used in the join or prune messages. It is recommended to use the default configuration for **pim ipv4 join-prune interval**, and modify the **pim ipv4 join-prune count** to increase the join or prune holdtime. Increasing the join-prune hold time delays the deletion of an S,G route on the upstream neighbor when join-prune messages are not sent to the neighbor. The maximum possible value for join or prune hold-time is ***65535***.

The **no pim ipv4 join-prune count** and **default pim ipv4 join-prune count** commands restore the default join or prune count for the configuration mode interface by removing the corresponding **pim ipv4 join-prune count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-VLAN Configuration

Command Syntax

pim ipv4 join-prune count ***count_value***

no pim ipv4 join-prune count ***count_value***

default pim ipv4 join-prune count ***count_value***

Parameter

count_value The number of missed join or prune after which the route expires. Value ranges from ***1.5*** to ***65535***.

Example

This command indicates the number of times a join or prune messages can be missed.

```
switch(config)# interface Ethernet 1/1
```

```
switch(config-if-Et1/1)# pim ipv4 join-prune count 5
switch(config-if-Et1/1)#
```

16.3.5.16 pim ipv4 join-prune interval

The **pim ipv4 join-prune interval** command specifies the period between join or prune messages that the configuration mode interface originates and sends to the upstream RPF neighbor.

The **no pim ipv4 join-prune interval** and **default pim ipv4 join-prune interval** commands restores the default join or prune interval to ***60*** seconds for the configuration mode interface by removing the corresponding **pim ipv4 join-prune interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

pim ipv4 join-prune interval [***period***]

no pim ipv4 join-prune interval [***period***]

default pim ipv4 join-prune interval [***period***]

Parameter

period join or prune interval (seconds). Value ranges from ***1*** to ***18724***. Default is ***60***.

Example

This command configures ***75***-second intervals between join or prune messages originating from ***interface vlan 4***.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# pim ipv4 join-prune interval 75
switch(config-if-Vl4)#
```

16.3.5.17 pim ipv4 neighbor-filter

The **pim ipv4 neighbor-filter** command configures the configuration mode interface to filter PIM control packets on the basis of neighbor addresses listed in a specified standard access list.

The **no pim ipv4 neighbor-filter** and **default pim ipv4 neighbor-filter** commands disable the configuration mode interface from filtering PIM control packets by removing the corresponding **ip pim ipv4 neighbor-filter** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

pim ipv4 neighbor-filter ***access_list***

no pim ipv4 neighbor-filter

default pim ipv4 neighbor-filter

Parameter

access_list name of the standard IP access list.

Example

This command configures the IP access list named ***filter_1*** to filter neighbor PIM control messages for ***interface vlan 4***.

```
switch(config)# ip access-list standard filter_1
switch(config-std-acl-filter_1)# permit 10.13.24.9/24
switch(config-std-acl-filter_1)# exit
switch(config)# interface vlan 4
switch(config-if-Vl4)# pim ipv4 neighbor-filter filter_1
switch(config-if-Vl4)#
```

16.3.5.18 pim ipv4 sparse-mode

The **`pim ipv4 sparse-mode`** command enables PIM IPv4 Sparse Mode (PIM-SM) and IGMP (router mode) on the configuration mode interface.

Note: PIM and Multicast Border Router (MBR) must be mutually exclusive on an interface. If the interface is configured as an MBR, do not enable PIM on the interface.

The **`no pim ipv4 sparse-mode`**, **`no pim ipv4, default pim ipv4 sparse-mode`**, and **`default pim ipv4`** commands restore the default PIM and IGMP (router mode) settings of **disabled** on the configuration mode interface by removing the **`pim ipv4 sparse-mode`** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

`pim ipv4 sparse-mode`

`no pim ipv4`

`no pim ipv4 sparse-mode`

`default pim ipv4`

`default pim ipv4 sparse-mode`

Example

This command enables PIM sparse mode on ***interface vlan 4*** interface.

```
switch(config)# interface vlan 4
switch(config-if-Vl4)# pim ipv4 sparse-mode
switch(config-if-Vl4)#
```

16.3.5.19 pim ipv6 sparse-mode

The **`pim ipv6 sparse-mode`** command enables PIM IPv6 Sparse Mode (PIM-SM) on the configuration mode interface.

Note: PIM and Multicast Border Router (MBR) must be mutually exclusive on an interface. If the interface is configured as an MBR, do not enable PIM on the interface.

The **`no pim ipv6 sparse-mode`**, **`no pim ipv6, default pim ipv6 sparse-mode`**, and **`default pim ipv6`** commands restore the default PIM settings of **disabled** on the configuration mode interface by removing the **`pim ipv6 sparse-mode`** command from the ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Interface-VLAN Configuration

Command Syntax

pim ipv6 sparse-mode

no pim ipv6

no pim ipv6 sparse-mode

default pim ipv6

default pim ipv6 sparse-mode

Example

This command enables IPv6 PIM sparse mode on ***interface vlan 8***.

```
switch(config)# interface vlan 8
switch(config-if-Vl8)# pim ipv6 sparse-mode
switch(config-if-Vl8)#
```

16.3.5.20 register local-interface

The **register local-interface** command programs the switch to fill the source field in all outbound PIM SM register packets with the IP address of a specified interface or the incoming interface of the group specified by the message. By default, the source field is filled with the IP address from the interface associated with the best route to the RP.

When the command is issued in Router-PIM Sparse-mode IPv4 ConfigurationMode, it applies to the default VRF; to use this command in a non-default VRF, issue it in Router-PIM Sparse-mode VRF IPv4 Configuration.

The **no register local-interface** and **default register local-interface** commands restore the default method of filling the register packet source field by removing the **ip register local-interface** statement from ***running-config***.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Command Syntax

register local-interface **INT_NAME**

no register local-interface

default register local-interface

Parameters

INT_NAME Interface type and number. Values include:

- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port channel interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.

Example

These commands program the switch to fill the source field of outbound PIM SM register packets in the default VRF with the IPv4 address of ***interface loopback 2***.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# register local-interface loopback 2
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.21 router pim bidirectional

The **`router pim bidirectional`** command places the switch in the ***router-pimbidirectional*** configuration mode.

Command Mode

Global Configuration

Command Syntax

`router pim bidirectional`

Example

This command places the switch in the ***router-pim bidirectional*** configuration mode.

```
switch(config)# router pim bidirectional
switch(config-router-pim-bidir)#
```

16.3.5.22 router pim bsr

The **`router pim bsr`** command places the switch in the ***router-pim bsr*** configuration mode.

Command Mode

Global Configuration

Command Syntax

`router pim bsr`

Example

This command places the switch in the ***router-pim bsr*** configuration mode.

```
switch(config)# router pim bsr
switch(config-router-pim-bsr)#
```

16.3.5.23 router pim sparse-mode

The **`router pim sparse-mode`** command places the switch in the ***router-pim sparse-mode*** configuration mode.

Command Mode

Global Configuration

Command Syntax

`router pim sparse-mode`

Example

This command places the switch in the ***router-pim sparse-mode*** configuration mode.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)#
```

16.3.5.24 rp address

The **rp address** command configures the address of a Protocol Independent Multicast (PIM) static Rendezvous Point (RP) for a specified multicast subnet. If the command does not specify a subnet, the static RP maps to all multicast groups (***224/4***). Dynamic RPs override static RPs unless the static RP is given priority by using the **override** option of this command.

Multicast groups use RPs to connect sources and receivers. A PIM domain requires that all routers have consistently configured RP addresses.

The switch uses multiple **rp address** commands to configure multiple RPs or to assign multiple subnets to an RP. When the address of a multicast group falls within multicast subnets configured by multiple **rp address** commands, the groups RP address is selected by comparing the commands multicast subnet size.

- Different size subnets: group uses command with the largest subnet.
- Same size subnets: group uses command as determined by hash algorithm.

When the command is issued in the ***router-multicast*** configuration mode it applies to the default VRF; to use this command in a non-default VRF, issue it in the ***router-multicast vrf*** configuration mode.

The **no rp address** and **default rp address** commands remove the corresponding **rp address** command from ***running-config***.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Command Syntax

rp address ***rp_addr*** [**MULTICAST_SUBNET**][**HASHMASK_LENGTH**][**BSR_OVERRIDE**] [**PRIORITY_NUM**]

no rp address ***rp_addr*** [**MULTICAST_SUBNET**]

default rp address ***rp_addr*** [**MULTICAST_SUBNET**]

Parameters

- ***rp_addr*** Rendezvous point IP address (dotted decimal notation).
- **MULTICAST_SUBNET** Multicast IP address space (CIDR or address-mask).
  - ***no parameter*** Default multicast group IP address of ***224/4***.
  - ***gp_addr*** Multicast group IP address (CIDR or address-mask).
  - **access-list** ***acl_name*** Standard access control list that specifies the multicast group address.
  - ***acl_name*** Standard access control list that specifies the multicast group address.
- **HASHMASK_LENGTH** Length (in bits) of the hash mask.
  - ***no parameter*** hash mask remains unchanged from previous setting.
  - **hashmask** ***0 - 32*** hash mask length (in bits). Default value is ***30***.
- **BSR_OVERRIDE** Configures priority relative to dynamic RPs selected by BSR.
  - ***no parameter*** Dynamic RPs have priority over specified RP.
  - **override** RP has priority over dynamic RPs.
- **PRIORITY_NUM** BSR election priority rating. Larger numbers denote higher priority. Default value is 0.
  - ***no parameter*** priority remains unchanged from previous setting.

- **priority *0 - 255*** priority rating.

Example

These commands configures ***10.17.255.2*** as a static RP for all multicast groups in the default VRF.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# rp address 10.17.255.2
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.25 rp allow

The **`rp allow`** command accepts and allows PIM (*,G) join message with an RP address that is different from the configured RPs for that particular (*,G).

The **`no rp allow`** and **`default rp allow`** commands disable this behavior by removing the corresponding **`rp allow`**command from ***running-config***.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Command Syntax

`rp allow`

`no rp allow`

`default rp allow`

Examples

- These commands configure the switch to accept PIM (*,G) join messages in the default VRF that include RP addresses not configured on the switch for that (*,G) route.

```
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# rp allow
```

- These commands configure the switch to accept PIM (*,G) join messages in VRF blue that include RP addresses not configured on the switch for that (*,G) route.

```
switch(config-router-pim-sparse)# vrf blue
switch(config-router-pim-sparse-vrf-blue)# ipv4
switch(config-router-pim-sparse-vrf-blue-ipv4)# rp allow
```

16.3.5.26 rp candidate

The **`rp candidate`** command configures the switch as a Candidate Rendezvous Point (C-RP). The BSR selects a multicast groups dynamic RP set from the list of C-RPs in the PIM domain. The command specifies the interface (used to derive the RP address), C-RP advertisement interval, and priority rating. The BSR selects the RP set by comparing C-RP priority ratings. The C-RP advertisement interval specifies the period between successive C-RP advertisement message transmissions to the BSR.

Running-config supports multiple multicast groups through multiple **`rp candidate`** statements:

- All commands must specify the same interface. Issuing a command with an interface that differs from existing commands removes all existing commands from ***running-config***.
- The ***running-config*** stores the **interval** setting in a separate statement that applies to all **`rp candidate`** statements. When a command specifies an interval that differs from the previously configured value, the

new value replaces the old value and applies to all configured **rp candidate** statements. The default **interval** value is ***60*** seconds.

When the command is issued in Router-Multicast Configuration Mode it applies to the default VRF; to use this command in a non-default VRF, issue it in the ***router-multicast vrf*** configuration mode.

The **no rp candidate** and **default rp candidate** commands remove **rp candidate** from ***running-config*** for the specified group. When these commands do not specify a multicast group, all **rp candidate** statements are removed from ***running-config***.

The **no rp candidate interval** and **default rp candidate interval** commands restore the interval setting to the default value of ***60*** seconds. The **no rp candidate priority** and **default rp candidate priority** commands restore the priority setting to the default value of ***192***.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Router-PIM Bidirectional IPv4 Configuration

Router-PIM Bidirectional VRF IPv4 Configuration

Command Syntax

Note: The **INTERFACE** parameter is always listed first. All other parameters can be placed in any order.

rp candidate INTERFACE [**GROUP_ADDR**][**PRIORITY_NUM**][**INTERVAL_PERIOD**]

no rp candidate [**INTERFACE**][**GROUP_ADDR**]

no rp candidate [**INTERFACE**] interval

no rp candidate [**INTERFACE**] priority

default rp candidate [**INTERFACE**][**GROUP_ADDR**]

default rp candidate [**INTERFACE**] interval

default rp candidate [**INTERFACE**] priority

Parameters

- **INTERFACE** Switch uses IP address of specified interface as its C-RP address. Options include:
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- **GROUP_ADDR** address of multicast group for which candidate is configured. Options include:
  - ***no parameter*** default multicast group (***224.0.0.0/4***).
  - ***net_addr*** multicast IPv4 subnet address (CIDR or address mask).
  - **access-list *acl_name*** standard access control list that specifies the multicast group address.
- **PRIORITY_NUM** RP selection priority rating. Smaller numbers denote higher priority.
  - ***no parameter*** priority rating is set to the default value of ***192***.
  - **priority *0 - 255*** priority rating.
- **INTERVAL_PERIOD** Period between consecutive RP-advertisement message transmissions (seconds). Value also applies to previously configured **rp candidate** statements.
  - ***no parameter*** interval remains unchanged from previous setting.

- **interval *10 - 16383*** transmission interval.

Example

These commands configure a switch as a candidate RP for the multicast group ***235.1.1.0/24*** with a priority of ***48*** and an RP advertisement interval of ***45*** seconds in the default VRF. The switch advertises the IP address assigned to ***vlan 24*** as its RP address.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# rp candidate vlan 24 235.1.1.0/24
 priority 48 interval 45
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.27 rp hash algorithm modulo

The **rp hash algorithm modulo** command configures the load-balancing scheme across available Rendezvous Points (RP).

The configuration results in a round robin-based load balancing across available RPs, achieved by module operation of the destination group address with the number of RPs available.

The **no rp hash algorithm modulo** and **default rp hash algorithm modulo** commands result in the default load-balancing scheme which is to use a hash function to get a group-RP mapping.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Router-PIM Sparse-mode IPv6 Configuration

Router-PIM Sparse-mode VRF IPv6 Configuration

Router-PIM Bidirectional VRF IPv4 Configuration

Command Syntax

rp hash algorithm modulo

no rp hash algorithm modulo

default rp hash algorithm modulo

Examples

- These commands configures the hash algorithm module for a VRF named ***blue*** in the ***router-pim sparse-mode ipv4*** configuration mode.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# vrf red
switch(config-router-pim-sparse-vrf-red-ipv4)# rp hash algorithm modulo
```

- These commands configures the hash algorithm module for a VRF named ***blue*** in the ***router-pim bidirectional-mode vrf ipv4*** configuration mode.

```
switch(config)# router pim bidirectional
switch(config-router-pim-bidir)# ipv4
switch(config-router-pim-bidir-ipv4)# vrf red
switch(config-router-pim-bidir-vrf-red-ipv4)# rp hash algorithm modulo
```

16.3.5.28 rp-candidate advertisement-filter

The **rp-candidate advertisement-filter** command filters the RP candidate advertisements from certain IP addresses. When **rp-candidate advertisement-filter** command is configured, PIM BSR filters RP candidate messages from ip-addresses matching the prefix list from the access-list that is configured.

The **no rp-candidate advertisement-filter** and **default rp-candidate advertisement-filter** commands removes **rp-candidate advertisement-filter** from ***running-config*** for the specified group.

Command Mode

Router-PIM BSR IPv4 Configuration

Router-PIM BSR VRF IPv4 Configuration

Command Syntax

rp-candidate advertisement-filter access-list ***access-list_name***

no rp-candidate advertisement-filter access-list***access-list_name***

default rp-candidate advertisement-filter access-list***access-list_name***

Parameter

access-list_name Standard access control list that specifies the multicast group address.

Example

These commands configure the switch as a candidate RP advertisement filter for the multicast group in the non-default VRF.

```
switch(config-router-pim-bsr)# ipv4
switch(config-router-pim-bsr-ipv4)# rp-candidate advertisement-filter access-
list test1
switch(config-router-pim-bsr-vrf-red-ipv4)# rp-candidate advertisement-filter
 access-list test2
```

16.3.5.29 sg-expiry-timer

The **sg-expiry-timer** command configures the (S, G) expiry timer interval for PIM-SM (S, G) multicast routes. The command does not apply to (*, G) mroutes.

When the command is issued in the ***router-multicast*** configuration mode it applies to the default VRF; to use this command in a non-default VRF, issue it in the ***router-multicast vrf*** configuration mode.

The **no sg-expiry-timer** and **default sg-expiry-timer** commands restore the default setting of ***210*** seconds by removing the **sg-expiry-timer** statement from ***running-config***.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Command Syntax

sg-expiry-timer ***period***

no sg-expiry-timer

default sg-expiry-timer

Parameter

period expiry timer interval (seconds). Value ranges from ***120*** to ***65535*** seconds. The default value is ***210*** seconds.

Example

These commands configure ***150*** seconds as the (S,G) expiry timer interval is in the default VRF.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# sg-expiry-timer 150
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.30 show ip pim bsr

The **show ip pim bsr** command displays the switchs BootStrap Router (BSR) information.

Command Mode

EXEC

Command Syntax

show ip pim bsr [**GROUP_FILTER**]

Parameters

GROUP_FILTER specifies groups for which command displays information.

- ***no parameter*** Displays data for all groups.
- ***net_addr*** Displays message for specified group address. (CIDR or address mask).

Example

This command configures the switch's BSR information.

```
switch> show ip pim bsr
PIMv2 Bootstrap information
This system is the Bootstrap Router (BSR)
BSR address: 10.1.1.1
Uptime:      00:14:42, BSR Priority: 0, Hash mask length: 30
Next bootstrap message in 00:00:05
```

16.3.5.31 show ip pim config-sanity

The **show ip pim config-sanity** command displays diagnostic information about the switch's PIM configuration.

Command Mode

EXEC

Command Syntax

show ip pim config-sanity

Example

This command displays PIM configuration diagnostic information.

```
switch> show ip pim config-sanity
DISCLAIMER: Below are only hints of potential PIM misconfiguration.
They do not necessary imply that there is a real problem.

The interfaces with PIM which are down: Vl4 switch>
```

16.3.5.32 show ip pim interface

The **show ip pim interface** command displays information about interfaces configured for PIM.

Command Mode

EXEC

Command Syntax

show ip pim interface [**INT_NAME**][**INFO_LEVEL**]

Parameters

- **INT_NAME** Interface type and number. Values include:
  - ***no parameter*** displays information for all interfaces.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
  - **INFO_LEVEL** specifies level of information detail provided by the command.
    - ***no parameter*** table of basic configuration information.
    - **detail** list of complete configuration information.

Examples

- This command displays information about all interfaces on which PIM is enabled.

```
switch> show ip pim interface
Address InterfaceModeNeighbor Hello DR DR Address  PktsQed   PktsDropped
 Count  Intvl Pri
10.17.254.30   Vlan3910        sparse1     30         1    10.17.254.30    0
         0
10.17.254.162  Vlan3925        sparse2     30         1    10.17.254.163   0
         0
10.17.254.106  Vlan3912        sparse1     30         1    10.17.254.106   0
         0
10.17.254.137  Ethernet12      sparse1     30         1    10.17.254.138   0
         0 switch>
```

- This command displays detailed PIM information for ***interface vlan 26***.

```
switch> show ip pim interface vlan 26 detail
Interface address is 172.17.26.1
Vif number is 1
PIM: enabled
  PIM version: 2, mode: sparse
  PIM DR: 172.17.26.1 (this system)
  PIM DR Priority: 1
  PIM neighbor count: 0
  PIM Hello Interval: 30 seconds
  PIM Hello Priority: 1
  PIM Hello Lan Delay: 500 milliseconds
  PIM Hello Override Interval: 2500 milliseconds
  PIM Hello Lan Prune Delay in use
  PIM Hello Generation ID: 0x4a05aa0
  PIM Hello Generation ID is not required
  PIM Triggered Hello Delay: 5 seconds
  PIM Join-Prune Interval: 60 seconds
  PIM State-Refresh processing: disabled
  PIM State-Refresh Interval: unknown seconds
  PIM Graft Retry Interval: unknown seconds
```

```
   PIM domain border: disabled switch>
```

16.3.5.33 show ip pim neighbor

The **show ip pim neighbor** command displays information about Protocol Independent Multicast (PIM) neighbors discovered by hello messages.

Command Mode

EXEC

Command Syntax

show ip pim neighbor [**INT_NAME**][**BFD_DATA**]

Parameters

- **INT_NAME** Interface type and number. Values include:
  - ***no parameter*** displays information for all interfaces.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.
  - **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.
- **BFD_DATA** Specifies inclusion of BFD data.
  - ***no parameter*** BFD data is not displayed.
  - **bfd** BFD data is displayed.

Examples

- This command displays information about neighbor PIM routers.

```
switch> show ip pim neighbor
PIM Neighbor Table
Neighbor Address   Interface       Uptime     Expires    Mode
10.17.255.2        Vlan2028        21d22h     00:01:31   sparse switch>
```

- This command displays information about neighbor PIM routers and the status of BFD.

```
switch> show ip pim neighbor bfd
PIM Neighbor Table
Flags: U - BFD is enabled and is UP
       I - BFD is enabled and is INIT
       D - BFD is enabled and is DOWN
       N - Not running BFD

Neighbor Address   Interface       Uptime     Expires    ModeFlags
10.17.255.2        Vlan2028        21d22h     00:01:31   sparseU switch>
```

16.3.5.34 show ip pim protocol counters

The **show ip pim protocol** command displays statistics about Protocol Independent Multicast (PIM) control messages sent and received by the switch.

Command Mode

EXEC

Command Syntax

show ip pim protocol counters [**INT_NAME**]

Parameters

INT_NAME Interface type and number. Values include:

- ***no parameter*** displays information for all interfaces.
- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vxlan *vx_num*** VXLAN interface specified by ***vx_num***.

Example

This command displays statistics about inbound and outbound PIM control messages.

```
switch> show ip pim protocol counters
PIM Control Counters
                         Received          Sent      Invalid
Assert                          0            37            0
Bootstrap Router                0             0            0
CRP Advertisement               0             0            0
Graft                           0             0            0
Graft Ack                       0             0            0
Hello                       63168        126355            0
J/P                        275714        143958            0
Join                            0             0            0
Prune                           0             0            0
Register                        0         13643            0
Register Stop               11839             0            0
State Refresh                   0             0            0 switch>
```

16.3.5.35 show ip pim register-source

The **show ip pim register-source** command displays the name of the interface from where the switch derives the IP address that it uses to fill the source field in all outbound PIM SM register packets. The **register local-interface** command specifies this interface.

By default, the source field is filled with the IP address from the interface associated with the best route to the RP. The **show ip pim register-source** command does not return a value when the source field is filled with the default value.

Command Mode

EXEC

Command Syntax

show ip pim register-source

Example

This command displays the register-source interface.

```
switch> show ip pim register-source
Ethernet22 switch>
```

16.3.5.36 show ip pim rp

The **show ip pim rp** command displays the status and multicast group of each cached Rendezvous Point (RP).

Command Mode

EXEC

Command Syntax

show ip pim rp

Example

This command displays the cached RPs.

```
switch> show ip pim rp
show ip pim rp
The PIM RP Set
Group: 224.0.0.0/4
RP: 10.1.2.3
Uptime: 00:05:12, Expires: never, Priority: 1 Override: 1
```

16.3.5.37 show ip pim rp-candidate

The **show ip pim rp-candidate** command displays the Rendezvous Point (RP) that is used for a specified multicast group.

Command Mode

EXEC

Command Syntax

show ip pim rp-candidate

Example

This command displays the switch's candidate-RP information.

```
switch> show ip pim rp-candidate
Candidate RP information
Candidate RP Address: 10.0.12.2
CRP Holdtime: 150 seconds
Group 224.2.0.0/16 Priority 2
```

16.3.5.38 show ip pim rp-hash

The **show ip pim rp-hash** displays the group to RP-hash mapping for the specified group and the list of qualifying candidate RPs.

Command Mode

EXEC

Command Syntax

show ip pim rp-hash ***ipv4_addr*** [**INFO_LEVEL**]

Parameters

- ***ipv4_addr*** multicast group IPv4 address.
- **INFO_LEVEL** specifies level of information detail provided by the command.
  - ***no parameter*** RP-hash map and list of candidate RPs.

- **detail** includes data about the selected RP.

Example

This command displays the RP that the switch uses for multicast group ***224.1.0.0***.

```
switch> show ip pim rp-hash 224.1.0.0
RP 10.1.2.3
```

16.3.5.39 show ip pim upstream joins

The **show ip pim upstream joins** command displays the join messages that the switch is scheduled to send.

Command Mode

EXEC

Command Syntax

show ip pim upstream joins [**JOIN_ADDRESSES**] [**NEIGHBOR_FILTER**]

Parameters

- **JOIN_ADDRESSES** Filters messages by source and group addresses.
  - ***no parameter*** displays all join messages.
  - ***source_addr*** displays all join messages for specified source group IPv4 address.
  - ***group_addr*** displays all join messages for specified multicast IPv4 address.
  - ***source_addr group_addr*** displays join message with specified source and group addresses.
  - ***group_addr source_addr*** displays join message with specified group and source addresses.

    group_addr must be a valid multicast IPv4 address.
- **NEIGHBOR_FILTER** specifies neighbors for which command provides data.
  - ***no parameter*** Displays messages for all neighbors.
  - **neighbor *neighbor_addr*** Displays message for specified neighbor address.

Example

This command displays the list of join messages the switch is scheduled to send. The example only displays the first two messages.

```
switch> show ip pim upstream joins

------------- show ip pim upstream joins -------------

Neighbor address: 10.1.1.1
Via interface: 10.1.1.2
Next message in 1 seconds
  Group: 10.10.10.3
    Joins:
      10.25.1.1/32 SPT
    Prunes:
      No prunes included
Neighbor address: 10.1.1.6
Via interface: 10.1.1.5
Next message in 1 seconds
  Group: 10.14.1.69
    Joins:
      10.105.14.3/32 SPT
    Prunes:
      No prunes included switch>
```

16.3.5.40 spt threshold

The **spt threshold** command configures Shortest Path Tree (SPT) threshold actions for IPv4 multicast groups. To specify the threshold action for multicast groups that match a specified Access Control List (ACL), use the **match list** option. When the command is issued without this option, it is applied throughout the configuration-mode VRF. Any ACL-based configuration overrides the ***global*** configuration.

- When ***running-config*** does not list this command, the switch joins the SPT immediately after receiving the first PIM packet from a new source. The switch joins the SPT by sending PIM join message toward the source.
- When ***running-config*** lists this command with a value of infinity, the switch never joins the SPT.

The **no spt threshold** and **default spt threshold** commands remove the corresponding **spt threshold** command from ***running-config***.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Command Syntax

spt threshold {*0* | infinity}[match list ***acl_name***]

no spt threshold {*0* | infinity}[match list ***acl_name***]

default spt threshold{*0* | infinity} [match list ***acl_name***]

Parameters

- ***0*** The switch immediately joins the SPT. This is the default value.
- **infinity** The switch never joins the SPT.
- ***acl_name*** name of access control list. If no ACL is supplied, the configuration is applied to all multicast groups within the VRF which are not configured by an ACL.

Example

This command configures the switch in the default VRF to immediately join the SPT for multicast groups matched by the ***ACL group-1***.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# spt threshold 0 match list group-1
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.41 ssm range

The **ssm range** command defines the Source Specific Multicast (SSM) range of IP multicast addresses.

SSM is a multicast packet delivery method where only packets originating from a specific source address requested by a receiver are routed to that receiver. SSM explicitly excludes the use of (*,G) join for applicable multicast groups. Source-specific multicast differs from Any-Source Multicast (ASM), where a receiver expresses interest in traffic to a multicast address, then receives traffic from all multicast sources sending to that address.

When the command is issued in the ***router-pim sparse-mode ipv4*** configuration mode it applies to the default VRF; to use this command in a non-default VRF, issue it in the ***router-pim sparse-mode vrf ipv4*** configuration mode.

The **no ssm range** and **default ssm range** commands remove the SSM IP multicast address range by deleting the **ssm range** statement from ***running-config***.

Command Mode

Router-PIM Sparse-mode IPv4 Configuration

Router-PIM Sparse-mode VRF IPv4 Configuration

Command Syntax

ssm range {***acl_name*** | **standard**}

no ssm range

default ssm range

Parameters

- ***acl_name*** sets the SSM range to address set specified by the standard ACL.
- **standard** sets the SSM range to ***232/8***.

Examples

- These commands configure the SSM address range to ***232/8*** in the default VRF.

```
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# ssm range standard
switch(config-router-pim-sparse-ipv4)#
```

- These commands configure the SSM address range to those permitted by the ***LIST_1*** standard ACL in the default VRF. The ACL permits the subnet address range ***233.0.0.0/24***.

```
switch(config)# ip access-list standard LIST_1
switch(config-std-acl-LIST_1)# permit 233.0.0.0/24
switch(config-std-acl-LIST_1)# exit
switch(config)# router pim sparse-mode
switch(config-router-pim-sparse)# ipv4
switch(config-router-pim-sparse-ipv4)# ssm range LIST_1
switch(config-router-pim-sparse-ipv4)#
```

16.3.5.42 sztimeout

The **sztimeout** command configures the maximum span of active scope-zone.

The **no sztimeout** and **default sztimeout** commands delete the current scope zoned timeout configuration.

Command Modes

Router-PIM BSR IPv4 Configuration

Router-PIM BSR VRF IPv4 Configuration

Syntax

sztimeout timeout

no sztimeout

default sztimeout

Parameter

timeout Maximum span of active scope-zone in seconds. The value ranges from ***120*** to ***4294967295***. The default value is ***1300***.

Guideline

The scope zoned timeout must contain a minimum value of ***10*** times of configured holdtime; else the system displays a warning message.

Examples

- This command configures ***600*** seconds as the maximum of active scope-zone in the ***router-pim bsr ipv4*** configuration mode.

```
switch(config)# router pim bsr
switch(config-router-pim-bsr)# ipv4
switch(config-router-pim-bsr-ipv4)# sztimeout 600
switch(config-router-pim-bsr-ipv4)#
```

- This command configures ***2200*** seconds as the maximum of active scope-zone in the ***router-pim bsr vrf ipv4*** configuration mode.

```
switch(config)# router pim bsr
switch(config-router-pim-bsr)# vrf vrf01
switch(config-router-pim-bsr-vrf-vrf01)# ipv4
switch(config-router-pim-bsr-vrf-vrf01-ipv4)# sztimeout 2200
switch(config-router-pim-bsr-vrf-vrf01-ipv4)#
```

16.4 Multicast Source Discovery Protocol (MSDP)

Multicast Source Discovery Protocol (MSDP) describes a topology that connects multiple IPv4 Protocol Independent Multicast Sparse-Mode (PIM-SM) domains. Each PIM-SM domain uses its independent Rendezvous Point (RP) without depending on RPs in other domains.

These sections describe the Arista MSDP implementation.

- MSDP Introduction
- MSDP Description
- MSDP Configuration
- MSDP Commands

16.4.1 MSDP Introduction

Arista switches support these MSDP features:

- Basic MSDP speaker functions.
- MSDP peer configuration: description, connect-source interface, keepalive time, and hold time.
- ACL filtering of inbound and outbound Source-Active (SA) messages.
- Mesh groups.
- Display of peer status.
- Display of filtered SA messages received from MSDP peers.

These MSDP features are not supported:

- MSDP is not supported with Anycast-RP (***RFC4610***).
- IP packet encapsulation.

16.4.2 MSDP Description

The Multicast Source Discovery Protocol (MSDP) defines a topology connecting Protocol Independent Multicast sparse mode (PIM-SM) domains.

MSDP provides inter-domain access to multicast sources in all domains by enabling all Rendezvous Points (RPs) to discover multicast sources outside of their domains. RPs also use MSDP to announce sources that are sending to a multicast group.

- MSDP Speakers
- Network Configuration

- MSDP Exchange Processes

16.4.2.1 MSDP Speakers

An MSDP speaker is a router in a PIM-SM domain that has MSDP peering sessions with MSDP peers in other domains. An MSDP peering session is a TCP connection through which peers exchange MSDP control information. An MSDP peer is a router that is connected to the speaker through a peering session.

PIM uses MSDP to register a local source with remote domain RPs through Source Active (SA) messages, which originate at the local domain's RP. Receivers in remote PIM-SM domains depend only on RPs in their domains to learn of multicast data sources in other domains. Multicast data is subsequently delivered from a source to receivers in different domains through a PIM-SM source tree.

MSDP Speaker Configuration describes the process of configuring MSDP speakers.

16.4.2.2 Network Configuration

The TCP connections between RPs are defined either through an underlying unicast routing table or by configuring a default MSDP peer. A typical MSDP configuration utilizes a BGP specified routing table. SA messages are MSDP control messages that peers exchange during peering sessions.

- Source Active Messages
- Reverse Path Forwarding
- Default MSDP Peers

16.4.2.2.1 Source Active Messages

A Source Active (SA) message is a message that an RP creates and sends to MSDP peers when it learns of a new multicast source through a PIM register message. RPs that intend to originate or receive SA messages must establish MSDP peering with other RPs, either directly or through intermediate MSDP peers. An RP that is not a DR on a shared network should only originate SAs in response to register messages it receives from the DR. It does not originate SA's for directly connected sources in its domain.

SA messages contain the following fields:

- Source address of the data source.
- Group address that receives data sent by the source.
- IP address of the RP.

Th SA Cache is the repository of SA messages received by the MSDP speaker. The switch always stores received SA messages. Managing the SA Cache describes procedures that limit the size of the SA cache and options for displaying the cache.

16.4.2.2.2 Reverse Path Forwarding

Reverse Path Forwarding (RPF) is a multicast packet transport technique that ensures loop-free packet forwarding by using a router's unicast routing table. Traffic forwarding is based on source addresses instead of destination addresses. RPF is implemented as defined in ***RFC 3618***.

Packet forwarding is based on the packet's unicast reverse path. An RPF router prevents network loops by only forwarding a packet when it enters through the interface holding its source routing entry.

When a multicast packet enters a router's interface, the router checks the reverse path of the packet by examining the list of networks that are reachable through the input interface. If the list contains a matching routing entry for the multicast packet's source IP address, the packet is forwarded to all other interfaces that are participants in the multicast group. Otherwise, the packet is dropped.

RPF requires that the unicast routing table is correct and converged. It also assumes that the use of symmetric forward and reverse paths between router and sender. RPF fails on uni-directional links.

Displaying RPF Peers describes commands that display RPF peers.

16.4.2.2.3 Default MSDP Peers

The default peer is the MSDP peer from which the MSDP speaker accepts SA messages. If there is only one MSDP peer, all of its SA messages will be accepted. When multiple default peers are configured the switch uses the first default peer to appear in the ***running-config***. Default MSDP peers invalidate the use of RPF over unicast routing tables.

Each default peer may be associated with a prefix list. The prefix list specifies the RPs from where the speaker accepts SA messages. When the ***running-config*** contains multiple default peers with prefix lists, an SA is accepted from the first default peer in the ***running-config*** whose prefix list contains the RP in the SA. The speaker accepts all remaining SAs from the first default peer that is not associated with a prefix list.

Configuring the Default Peer describes commands that configure default peers.

16.4.2.3 MSDP Exchange Processes

- **Control Information Exchange**
- **MSDP Data Exchange**

16.4.2.3.1 Control Information Exchange

An RP originates an SA message when a source registers with the RP to send data to a multicast group. RPs periodically originate SA messages while its registered sources send data to maintain messages in SA caches of its MSDP peers. RPs that have no registered sources periodically send keepalive messages to maintain TCP connections with its peers.

MSDP defines the following timers that specify the transmission frequency of control messages:

- SA Advertisement Time: Duration of SA Advertisement intervals. An RP sends periodic SA messages to reference each registered source once per interval. SA advertisement time is ***60*** seconds.
- Keepalive Time: Period between the transmission of consecutive keepalive messages. Default keepalive time is ***60*** seconds. Minimum keepalive time is one second.
- Hold Timer: Period an MSDP speaker maintains a peer TCP connection after receiving an SA or keepalive message from the peer. Default time is ***75*** seconds. Minimum hold time is three seconds.

16.4.2.3.2 MSDP Data Exchange

This sequence describes the exchange of multicast data across PIM domains through MSDP:

1. When a source's first data packet is registered by the first hop router, the RP extracts the data from the packet and forwards it down the shared tree in the PIM domain.
2. The RP informs MSDP peers of the new source by sending a Source-Active (SA) message that identifies the source, the recipient group, and the RP's address or originator ID.
3. Upon receiving the SA message, an MSDP peer which is the RP for a multicast tree that includes members interested in the multicast sends a PIM join message (S,G) toward the data source.
4. After the RP on another domain joins the PIM Designated Router (DR) in the first domain, multicast data traffic flows natively over the multicast tree to the second domain's RP.
5. If the source times out, this process repeats when the source goes active again.

16.4.3 MSDP Configuration

These sections describe the configuration of the switch as an MSDP speaker and the establishment of MSDP peering sessions.

- MSDP Speaker Configuration
- Establishing MSDP Peers
- MSDP Network Configuration
- Managing the SA Cache

- Configuring MSDP in a non-default VRF

MSDP requires that TCP ***port 639*** (MSDP) is open on the control plane. The default control-plane ACL includes a permit rule that allows TCP packets access through the MSDP port.

16.4.3.1 MSDP Speaker Configuration

The switch is configured as an MSDP speaker when MSDP is enabled. MSDP is enabled by configuring an MSDP peer. Configuring an MSDP Peer describes the process of configuring an MSDP peer.

Source-Address (SA) messages that an MSDP speaker originates contain the speaker's Rendezvous Point (RP) address, as configured through PIM statements and processes. MSDP provides a method of assigning an originator ID address, which the speaker uses in place of its RP address when advertising SA messages. The originator-id local-interface command configures the switch to set the RP address to the specified interface's IP address in SA messages that it originates as an MSDP speaker.

Only RPs originate SA messages and only for its registered sources. RPs do not originate periodic SA messages for sources in other PIM domains. MSDP speakers that are not RPs do not originate periodic SA messages. Intermediate MSDP speakers forward SA messages received from other domains. Intermediate speakers are not required to be RPs.

Example

These commands configure the switch to use the IP address assigned to ***interface loopback 100*** as the RP address in SA messages that it originates.

```
switch(config)# router msdp
switch (config-router-msdp)# originator-id local-interface loopback 100
switch (config-router-msdp)#
```

16.4.3.2 Establishing MSDP Peers

These sections describe MSDP Peer configuration tasks.

- Configuring an MSDP Peer
- Mesh Groups
- Filtering SA Messages
- Keep-alive, Hold Time, and Reset Time Configuration
- Displaying Peer Information

16.4.3.2.1 Configuring an MSDP Peer

The switch attempts to establish MSDP peering sessions through IP addresses configured as MSDP peers. The **peer** command configures a specified address as an MSDP peer and enables the switch as an MSDP speaker if no other peers are configured. The peering session with the device at the specified network is established over a TCP connection. The **local-interface** command can be used to specify an interface through which the switch establishes the TCP session. When no interface is specified, the connection is established through an interface determined by existing routing algorithms.

To display MSDP peer information, enter **show ip msdp peer** .

Example

These commands assign an IP address to ***loopback interface 100***, then configure ***10.4.4.12*** as an MSDP peer and establish the TCP peer session through the loopback.

```
switch(config)# interface loopback 100
switch(config-if-Lo100)# ip address 10.6.8.6/24
switch(config-if-Lo100)# exit
switch(config)# router msdp
```

```
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# local-interface loopback 100
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Connect
   Resets: 0
   Connection Source: Loopback100 ( 10.6.8.6 )
SAs accepted:
switch(config-router-msdp-peer-10.4.4.12)#
```

To associate descriptive text with the specified MSDP peer, use the description (MSDP) command.

Example

These commands associate the string ***NORTH*** with the MSDP peer located at ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# description NORTH
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Connect
   Resets: 0
   Connection Source: Loopback100 (10.6.8.6)
SAs accepted:
switch(config-router-msdp-peer-10.4.4.12)#
```

To close the peering session with the specified MSDP peer, use the disabled (MSDP) command. This terminates the TCP connection between the switch and the peer. The peer remains configured and the peer session can be resumed by removing the **disabled** command from ***running-config***.

Examples

- This command closes the peering session with the MSDP peer at ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# disabled
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Disbled
   Resets: 0
   Connection Source: Loopback100 ( 10.6.8.6 )
SAs accepted:
switch(config-router-msdp-peer-10.4.4.12)#
```

- This command reopens the peering session with the peer at ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# no disabled
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Connect
   Resets: 0
   Connection Source: Loopback100 ( 10.6.8.6 )
```

```
SAs accepted:
switch(config-router-msdp-peer-10.4.4.12)#
```

16.4.3.2.2 Mesh Groups

Each node in a fully meshed network is directly connected to every other node in the network. Each peer in a fully meshed MSDP speaker network can be configured as a member of a mesh group. SA messages received from a mesh group peer are not forwarded to other members of the mesh group.

To configure an MSDP peer connection as an MSDP mesh group member, use the **mesh-group** command. An MSDP peer can be assigned to multiple mesh groups. Multiple peer connections can be assigned to the same mesh group.

Note: Peer-specific mesh-group configuration is performed in Router MSDP Peer Configuration or Router MSDP Peer VRF Configuration Mode. To remove all configured connections from a mesh group, use the **no mesh-group** command in Router MSDP Configuration Mode.

To display the mesh group membership of configured MSDP peers, enter **show msdp mesh-group**.

Example

These commands configure the MSDP peer connection to ***10.1.1.14*** as a member of the ***AREA-1*** mesh group, then displays members of mesh groups to which configured MSDP peers belong.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.1.1.14
switch(config-router-msdp-peer-10.1.1.14)# mesh-group AREA-1
switch(config-router-msdp-peer-10.1.1.14)# show msdp mesh-group
Mesh Group: AREA-1
        10.1.1.14
Mesh Group: tier_01
        10.24.18.13
Mesh Group: tier_02
        10.26.101.18
switch(config-router-msdp-peer-10.1.1.14)#
```

16.4.3.2.3 Filtering SA Messages

The switch can filter Source-Active (SA) messages that it sends and receives with Access Control Lists (ACLs). The commands accept standard and extended ACLs. The address field in standard ACLs filters an SA message on its multicast source address.

The **sa-filter in** command assigns an ACL to filter inbound SA messages from the MSDP peer connection being configured. The switch only accepts SA messages from the peer that pass the ACL. The switch accepts all SA messages from peers that are not assigned an input ACL. A peer can be assigned only one input filter ACL. Subsequent **sa-filter in** commands for a peer replace the existing command.

The **sa-filter out** command assigns an ACL as a filter for outbound SA messages to the MSDP peer connection being configured. The switch only sends SA messages to the peer that pass the ACL. The switch sends all specified SA messages to peers not assigned an output filter ACL. A peer can be assigned only one output ACL. Subsequent **sa-filter out** commands for a peer replace the existing command.

Example

These commands assign the IP ACLs named ***LIST-IN*** as the inbound SA message filter and ***LIST-OUT*** as the outbound SA message filter for the MSDP peer connection to ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# sa-filter in list LIST-IN
switch(config-router-msdp-peer-10.4.4.12)# sa-filter out list LIST-OUT
```

```
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Listen
   Connection Source: Loopback100 (10.6.8.6)
SA Filtering:
Input Filter: LIST-IN
Output Filter: LIST-OUT
switch(config-router-msdp-peer-10.4.4.12)#
```

16.4.3.2.4 Keep-alive, Hold Time, and Reset Time Configuration

To configure the MSDP keep-alive and hold time intervals for a specified MSDP peer connection, use the **keepalive (MSDP)** command.

- Keep-alive time interval is the period between the transmission of consecutive keep-alive messages. The default keep-alive time interval is ***60*** seconds.
- Hold time interval is the period the switch waits for a KEEPALIVE or UPDATE message before it disables peering. The default hold time interval is ***75*** seconds.

The hold time interval must be longer than or equal to the keep-alive time interval.

Example

This command sets the keep-alive time to ***45*** seconds and the hold time to ***80*** seconds for the MSDP peer connection to ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp) #peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# keepalive 45 80
switch(config-router-msdp-peer-10.4.4.12)#
```

To specify the period that the switch waits after an MSDP peering session is reset before attempting to reestablish the session, enter **connection retry interval**. The default period is ***30*** seconds.

Example

This command configures the switch to wait ***45*** seconds after an MSDP peering session is reset before attempting to reestablish the session.

```
switch(config)# router msdp
switch(config-router-msdp)# connection retry interval 45
switch(config-router-msdp)#
```

To enable the encapsulation of multicast data packets on the sending MSDP peer and the decapsulation and forwarding of register packets on the receiving MSDP peer, use the **forward register-packets** command. The default is to not forward the data encapsulated in PIM register messages.

Example

This command enables the forwarding of encapsulated register packets.

```
switch(config)#router msdp
switch(config-router-msdp)#forward register-packets
switch(config-router-msdp)#
```

16.4.3.2.5 Displaying Peer Information

To display the MSDP peers, enter **show ip msdp summary**. The command also displays the operational status of each peer and the number of messages from the peers in the SA cache.

Example

This command displays the configured peers, the status of the peers, and the number of SA messages received from those peers.

```
switch(config)# show ip msdp summary
MSDP Peer Status Summary
Peer Address      State   SA Count
192.168.3.18      Up      0
192.168.3.16      Up      0
192.168.3.37      Listen  0
192.168.3.46      Up      0
192.168.3.47      Up      0
```

16.4.3.3 MSDP Network Configuration

- Displaying RPF Peers
- Configuring the Default Peer

16.4.3.3.1 Displaying RPF Peers

The switch uses the unicast routing table to define TCP connections between RPs by selecting the next hop peer toward the originating RP of an SA message as the Reverse Path Forwarding (RPF) peer. The switch forwards SA messages that it receives from the RPF peer to all other MSDP peers. The switch rejects SA messages that it receives from non-RPF peers.

To display MSDP information for the peer from which the switch accepts SA messages for a specified Rendezvous Point (RP), enter **show msdp rpf-peer**.

Example

This command displays MSDP information for the peer from which the switch accepts SA messages for the RP at ***10.5.29.4***.

```
switch(config)# show msdp rpf-peer 10.5.29.4
Rpf Peer is 10.5.29.4 for RP 10.5.29.4
```

16.4.3.3.2 Configuring the Default Peer

The default peer is the MSDP peer from which the MSDP speaker is configured to accept all SA messages. A default peer may be associated with a prefix list. The prefix list specifies the RPs from where the speaker accepts SA messages.

The switch can designate multiple default peers:

- Switch defines one peer: A default peer statement is not required; the switch accepts SA traffic from the configured peer.
- Switch defines one default peer (no prefix list): The switch accepts all SA messages from only the default peer.
- Switch defines multiple default peers (no prefix lists): The switch accepts all SA messages from only the first default peer listed in ***running-config***. Other listed default peers take effect only if the peer named in the first default-peer statement is not accessible.
- First default-peer statement includes a prefix list: The switch accepts all SA messages from the default peer whose originating RP is covered in the prefix list. The disposition of SA messages originating from other RPs is determined by subsequent **default-peer** statements.

To configure the specified MSDP peer connection as a default peer on the switch, use the **default-peer** command. The default peer address must be a previously configured MSDP peer (configured using the peer command).

Example

These commands configure an MSDP default peer.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.5.2.2
switch(config-router-msdp-peer-10.5.2.2)# default-peer
switch(config-router-msdp-peer-10.5.2.2)#
```

16.4.3.4 Managing the SA Cache

The switch stores Source Active (SA) messages after forwarding the information. This allows new group members to learn about the source before the next SA message is received. The caching action is not configurable and cannot be disabled.

SA messages have an expiration period of ***90*** seconds and remain in the SA cache until they expire. A peer's SA limit defines the number of SA messages the switch stores from the peer. The switch does not store SA messages from a peer whose SA limit is reached until its cached messages start expiring.

- Limiting SA Cache Contents
- Displaying SA Cache Contents
- Verifying Consistency Between the SA Cache and the Routing Table

16.4.3.4.1 Limiting SA Cache Contents

To configure the maximum number of SA messages from a specified MSDP peer that the switch stores in the SA cache, use the **sa-limit** command. The default limit of SA messages that the switch can store from a specified peer is ***40000***.

Example

This command sets the SA limit of ***500*** for the MSDP peer at ***10.1.1.5***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.1.1.5
switch(config-router-msdp-peer-10.1.1.5)# sa-limit 500
switch(config-router-msdp-peer-10.1.1.5)#
```

The maximum number of SA messages that the switch can store in the SA cache for a specified multicast group address is configured by the **group-limit** command. The default limit of SA messages that the switch can store from a specified group is ***40000***.

Example

This command sets the maximum number of 1000 SAs for multicast group ***225.13.15.8/29***.

```
switch(config)# router msdp
switch(config-router-msdp)# group-limit 1000 source 225.13.15.8/29
```

The maximum number of rejected SA messages that the switch can store in the SA cache is configured by the **ip msdp rejected-limit** command. The default limit of rejected SA messages that the switch can store is ***40000***.

Example

This command sets ***5000*** as the maximum number of rejected SAs that the SA cache can contain.

```
switch(config)# router msdp
switch(config-router-msdp)# ip msdp rejected-limit 5000
```

Contents of the SA message cache are removed by the **clear ip msdp sa-cache** command. The command provides options for removing all cache contents or only contents of a specific multicast group.

Example

This command deletes all SA message cache contents.

```
switch(config)# router msdp
switch(config-router-msdp)# clear ip msdp sa-cache
```

16.4.3.4.2 Displaying SA Cache Contents

SA message cache contents are displayed by the **show ip msdp sa-cache** command. Filter options provided by the command for displaying partial cache contents include:

- multicast group address: multicast group
- source address and group address

The command can also display unexpired SAs rejected by ACL filters or cache limit exceeded conditions.

Example

This command displays the contents of the SA message cache.

```
switch(config)# show ip msdp sa-cache
MSDP Source Active Cache
(10.61.71.29, 234.1.4.2), RP 10.5.29.4, heard from 10.5.29.4
(10.51.71.23, 234.1.4.1), RP 10.5.29.4, heard from 10.5.29.4
(10.53.71.27, 234.1.4.2), RP 10.3.25.4, heard from 10.3.25.4
(10.10.101.24, 234.1.4.1), RP 10.2.44.4, heard from 10.2.44.4
(10.10.151.22, 234.1.4.1), RP 10.1.12.4, heard from 10.1.12.4
```

Information about specified MSDP peers, including SAs accepted from the peer is displayed by the **show ip msdp peer** command.

Example

This command displays data for the peer at ***10.2.42.4***, including SAs accepted from the peer.

```
switch(config)# show ip msdp peer 10.2.42.4 accepted-sas
MSDP Peer 10.2.42.4
Connection status:
  State: Up
  Connection Source: Loopback4 (10.2.43.4)
SA Filtering:
Input Filter: allow-multicast-for-msdp
Output Filter: allow-multicast-for-msdp
SAs accepted:
(10.62.79.30, 234.1.4.2), RP 10.2.42.4
(10.61.79.29, 234.1.4.1), RP 10.2.42.4
(10.62.79.30, 234.1.4.1), RP 10.2.42.4
```

The SA cache for the local PIM domain is displayed by the **show ip msdp pim sa-cache** command.

Example

This command displays the SA cache for the local PIM domain.

```
switch(config)# show ip msdp pim sa-cache
MSDP Source Active Messages for local Pim RP
(10.51.71.23, 234.1.4.1), RP 10.2.43.4
(10.20.91.26, 234.1.4.1), RP 10.2.43.4
(10.20.91.26, 234.1.4.2), RP 10.2.43.4
```

```
(10.20.91.24, 234.1.4.1), RP 10.2.43.4
```

16.4.3.4.3 Verifying Consistency Between the SA Cache and the Routing Table

To check the consistency between the multicast routing table and the MSDP Source-Address (SA) caches, enter **show ip msdp sanity**. When the command detects inconsistencies, it displays the cache entries that are not in the table.

Example

This command displays a sanity check that detects inconsistencies between the SA cache and the multicast routing table.

```
switch(config)# show ip msdp sanity
PIM SA cache entries not in the MRT
Msdp-learnt MRT entries not in the SA cache
SA cache entries not in the MRT
(192.168.3.8, 224.1.154.1)
(192.168.3.35, 224.1.167.1)
(192.168.3.16, 224.1.226.1)
(192.168.3.12, 224.1.182.1)
(192.168.3.33, 224.1.150.1)
May-Notify-MSDP entries not in the PIM SA cache
(need not be an error condition)
4.1), RP 10.2.42.4
```

16.4.3.5 Configuring MSDP in a non-default VRF

The MSDP can also be configured in a non-default VRF, when the default VRF used does not have a name. The following commands configure MSDP in a non-default VRF.

Example

These commands configure MSDP peer ***1.1.1.1*** in a non-default VRF ***blue***.

```
switch(config)# router msdp
switch(config-router-msdp)# vrf blue
switch(config-router-msdp-vrf-blue)# peer 1.1.1.1
```

16.4.4 MSDP Commands

MSDP Configuration Commands (Global)

- connection retry interval
- forward register-packets (MSDP)
- group-limit
- ip msdp rejected-limit
- originator-id local-interface
- peer
- router msdp

MSDP Peer Configuration Commands

- default-peer
- description (MSDP)
- disabled (MSDP)
- keepalive (MSDP)

- local-interface
- mesh-group
- sa-filter in
- sa-filter out
- sa-limit

MSDP SA Cache Commands

- clear ip msdp sa-cache

MSDP Display Commands

- show ip msdp peer
- show ip msdp pim sa-cache
- show ip msdp sa-cache
- show ip msdp sanity
- show ip msdp summary
- show msdp mesh-group
- show msdp rpf-peer

16.4.4.1 clear ip msdp sa-cache

The **`clear ip msdp sa-cache`** command removes contents of the Source-Active (SA) message cache. The command provides these filter options for removing partial cache contents:

- contents of a multicast group by specifying its group address.
- all cache contents.

Command Mode

Router MSDP Configuration

Router MSDP VRF Configuration

Command Syntax

`clear ip msdp sa-cache` [**ADDRESS_FILTER**]

Parameters

ADDRESS_FILTER IPv4 address used to select table entries for removal.

- ***no parameter*** All SA messages.
- ***grp_addr*** Multicast group address (IPv4 address). The ***grp_addr*** must be a valid multicast address.

Example

This command deletes all SA message cache contents.

```
switch(config)# router msdp
switch(config-router-msdp)# clear ip msdp sa-cache
```

16.4.4.2 connection retry interval

The **`connection retry interval`** command specifies the period that the switch waits after an MSDP peering session is reset before trying to reestablish the session. The default period is ***30*** seconds.

The **`no connection retry interval`** and **`default connection retry interval`** commands reset the timer interval to the default period of ***30*** seconds by removing the **`connection retry interval`** command from ***running-config***.

Command Mode

Router MSDP Configuration

Router MSDP VRF Configuration

Command Syntax

`connection retry interval` ***connect_retry***

`no connection retry interval` ***connect_retry***

`default connection retry interval` ***connect_retry***

Parameter

connect_retry Reconnect period (seconds). Value ranges from ***1*** to ***65535***. Default is ***30***.

Example

This command configures the switch to wait ***45*** seconds after an MSDP peering session is reset before attempting to reestablish the session.

```
switch(config)# router msdp
switch(config-router-msdp)# connection retry interval 45
```

16.4.4.3 default-peer

The **`default-peer`** command configures the specified MSDP peer connection as a default peer on the switch. The default peer configuration defines the peers from which the switch accepts Source-Active (SA) messages. When the command includes a **prefix list** parameter, the specified peer is the default peer for only SA messages originating from rendezvous points (RPs) covered by prefix list entries. The default peer address must be a previously configured MSDP peer (configured using the **peer** command).

Default peers provide an alternative to Reverse Packet Forwarding (RPF) typically used by MSDP to specify the peers from which a switch accepts SA messages. However, RPF requires a unicast routing table that is correct and converged. RPF also assumes symmetric forward and reverse paths between router and sender. RPF fails on uni-directional links. Default MSDP peers invalidate the use of RPF over unicast routing tables.

The switch can designate multiple default peers:

- Switch defines one peer: A default peer statement is not required; the switch accepts SA traffic from the configured peer.
- Switch defines one default peer (no prefix list): The switch accepts all SA messages from only the default peer.
- Switch defines multiple default peers (no prefix lists): The switch accepts all SA messages from only the first default peer listed in ***running-config***. Other listed default peers are used only when peers listed before them in ***running-config*** are not accessible.
- First default-peer statement includes a prefix list: The switch accepts all SA messages from the default peer whose originating RP is covered in the prefix list. The disposition of SA messages originating from other RPs is determined by subsequent **`default-peer`** statements.

The **`no default-peer`** and **`default default-peer`** commands remove the corresponding **default-peer** command from ***running-config***.

Command Mode

Router MSDP Peer Configuration

Router MSDP Peer VRF Configuration

Command Syntax

`default-peer` [**PREFIX**]

no default-peer

default default-peer

Parameters

PREFIX List of RPs from the SA messages originate for whiich the default peer is valid.

- ***no parameter*** default peer is valid for SAs from all originating RPs.
- **prefix-list *list_name*** name of the prefix list that defines affected originating RP prefixes.

Example

These commands configure two MSDP peers and configure the peer at ***10.5.2.2*** as the default peer.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.6.2.2
switch(config-router-msdp-peer-10.6.2.2)# exit
switch(config-router-msdp)# peer 10.5.2.2
switch(config-router-msdp-peer-10.5.2.2)# default-peer
switch(config-router-msdp-peer-10.5.2.2)#
```

16.4.4.4 description (MSDP)

The **description** command associates descriptive text with the configuration-mode MSDP peer.

The **no description** and **default description** commands remove the text association from the specified peer.

Command Mode

Router MSDP Peer Configuration

Router MSDP Peer VRF Configuration

Command Syntax

description ***description_string***

no description

default description

Parameters

description_string text string that is associated with the peer.

Example

These commands associate the string ***NORTH*** with the MSDP peer located at ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# description NORTH
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Connect
   Resets: 0
   Connection Source: Loopback100 (10.6.8.6)
SAs accepted:
switch(config-router-msdp-peer-10.4.4.12)#
```

16.4.4.5 disabled (MSDP)

The **disabled** command closes the peering session with the specified MSDP peer by terminating the TCP connection between the switch and the peer. The connection is not resumed until the shutdown command is removed from ***running-config***.

The **no disabled** and **default disabled** commands establish an MSDP peering session with the specified peer by removing the corresponding **disabled** command from ***running-config***.

Command Mode

Router MSDP Peer Configuration

Router MSDP Peer VRF Configuration

Command Syntax

disabled

no disabled

default disabled

Examples

- This command closes the peering session with the MSDP peer at ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# disabled
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Disbled
   Resets: 0
   Connection Source: Loopback100 ( 10.6.8.6 )
SAs accepted:
switch(config-router-msdp-peer-10.4.4.12)#
```

- This command reopens the peering session with the peer at ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# no disabled
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Connect
   Resets: 0
   Connection Source: Loopback100 ( 10.6.8.6 )
SAs accepted:
switch(config-router-msdp-peer-10.4.4.12)#
```

16.4.4.6 group-limit

The **group-limit** command specifies the maximum number of Source-Active (SA) messages that the switch allows in the SA cache for a specified multicast group address.

SA messages have an expiration period of ***90*** seconds and remain in the SA cache until they expire. The switch does not accept SA messages for a group whose cache limit is reached until its cached messages start expiring.

The **no group-limit** and **default group-limit** command removes the maximum group limit for the specified prefix by removing the corresponding **group-limit** statement from ***running-config***.

Command Mode

Router MSDP Configuration

Router MSDP VRF Configuration

Command Syntax

group-limit ***quantity*** source***src_subnet***

no group-limit ***quantity*** source ***src_subnet***

default group-limit ***quantity*** source ***src_subnet***

Parameters

- ***quantity*** maximum number of groups that can access the interface. Value ranges from ***1*** to ***40000***.
- ***src_subnet*** Source IPv4 subnet (CIDR or address-mask notation).

Example

This command sets the maximum number of ***1000*** SAs for multicast group ***10.13.15.8/29***.

```
switch(config)# router msdp
switch(config-router-msdp)# group-limit 1000 source 10.13.15.8/29
```

16.4.4.7 ip msdp rejected-limit

The **ip msdp rejected-limit** command specifies the maximum number of rejected Source-Active messages that the switch allows in the SA cache.

SA messages have an expiration period of ***90*** seconds. They remain in the SA cache during this time. The default limit of rejected SA messages that the switch can store is ***40000***.

The **no ip msdp rejected-limit** and **default ip msdp rejected-limit** commands restore the rejected SA limit of ***40000*** by removing the **ip msdp rejected-limit** statement from ***running-config***.

Command Mode

Router MSDP Configuration

Router MSDP VRF Configuration

Command Syntax

ip msdp rejected-limit ***quantity***

no ip msdp rejected-limit

default ip msdp rejected-limit

Parameter

quantity maximum rejected SA messages the SA cache can store. Value ranges from ***0*** to ***40000***.

Example

This command sets ***5000*** as the maximum number of rejected SAs that the SA cache can contain.

```
switch(config)# router msdp
switch(config-router-msdp)# ip msdp rejected-limit 5000
```

16.4.4.8 keepalive (MSDP)

The **keepalive** command configures the MSDP keep-alive and hold time intervals for a specified MSDP peer connection.

- Keep-alive time interval is the period between the transmission of consecutive keep-alive messages. The default keep-alive time interval is ***60*** seconds.
- Hold time interval is the period the switch waits for a KEEPALIVE or UPDATE message before it disables peering. The default hold time interval is ***75*** seconds.

The **no keepalive** and **default keepalive** commands restore the default keep-alive and hold time intervals for the specified MSDP peer connection by removing the corresponding **keepalive** command from ***running-config***.

Command Mode

Router MSDP Peer Configuration

Router MSDP Peer VRF Configuration

Command Syntax

keepalive ***keep_alive hold_time***

no keepalive

default keepalive

Parameters

- ***keep_alive*** keep-alive period in seconds. Value ranges from ***1*** to ***65535***. Default value is ***60***.
- ***hold_time*** hold time in seconds. Value ranges from ***1*** to ***65535***. Default value is ***75***.

Note: The hold time interval must be longer than or equal to the keep-alive time interval.

Example

This command sets the keep-alive time to ***45*** seconds and the hold time to ***80*** seconds for the connection with the MSDP peer at ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# keepalive 45 80
switch(config-router-msdp-peer-10.4.4.12)#
```

16.4.4.9 local-interface

MSDP peering sessions are established over a TCP connection. The **local-interface** command specifies the interface through which the TCP connection is established with the configuration-mode MSDP peer. When the **local-interface** command is not used to specify an interface, the connection is established through an interface determined by existing routing algorithms.

The **no local-interface** and **default local-interface** commands remove the corresponding **local-interface** command from ***running-config***, returning selection of the connecting interface to the routing algorithm.

Command Mode

Router MSDP Peer Configuration

Router MSDP VRF Peer Configuration

Command Syntax

local-interface ***interface***

no local-interface

default local-interface

Parameters

interface local interface through which the TCP connection is established. Options include:

- **ethernet *e_num*** Ethernet interface.
- **loopback *l_num*** Loopback interface.
- **management *m_num*** Management interface.
- **port-channel *p_num*** Port-Channel Interface.
- **vlan *v_num*** VLAN interface.
- **vxlan *vx_num*** VXLAN interface.

Example

These commands assign an IP address to ***interface loopback 100***, then establish the TCP peer session to the MSDP peer at ***10.4.4.12*** through the loopback in the default VRF.

```
switch(config)# interface loopback 100
switch(config-if-Lo100)# ip address 10.6.8.6/24
switch(config-if-Lo100)# exit
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# local-interface loopback 100
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Connect
   Resets: 0
   Connection Source: Loopback100 (10.6.8.6)
SAs accepted:
switch(config-router-msdp-peer-10.4.4.12)#
```

16.4.4.10 mesh-group

The **mesh-group**command configures the configuration-mode MSDP peer connection as an MSDP mesh group member. A peer can be assigned to multiple mesh groups. Multiple MSDP peers can be assigned to a common mesh group.

An MSDP mesh group is a network of MSDP speakers where each speaker directly connects to every other speaker. The switch does not forward Source-Active (SA) messages that it receives from a mesh group peer to other peers of the same group.

The **no mesh-group** and **default mesh-group** commands delete the configuration-mode peer connection from a mesh group by removing the corresponding **mesh-group** command from ***running-config*** when issued in the ***router msdp peer*** configuration or the ***router msdp peer vrf*** configuration mode.

Note: To delete all configured connections from a specified mesh group, use the **no mesh-group** command in Router MSDP Configuration mode.

Command Mode

Router MSDP Configuration

Router MSDP Peer Configuration

Router MSDP Peer VRF Configuration

Command Syntax

mesh-group *group_name*

no mesh-group ***group_name***

default mesh-group ***group_name***

Parameters

group_name name of mesh group.

Related Command

show msdp mesh-group .

Examples

- These commands configure the MSDP peer connection to ***10.1.1.14*** as a member of the ***AREA-1*** mesh group, then display members of mesh groups to which configured MSDP peers belong.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.1.1.14
switch(config-router-msdp-peer-10.1.1.14)# mesh-group AREA-1
switch(config-router-msdp-peer-10.1.1.14)# show msdp mesh-group
Mesh Group: AREA-1
        10.1.1.14
Mesh Group: tier_01
        10.24.18.13
Mesh Group: tier_02
        10.26.101.18
switch(config-router-msdp-peer-10.1.1.14)#
```

- These commands delete all configured connections from the ***AREA-1*** mesh group.

```
switch(config)# router msdp
switch(config-router-msdp)# no mesh-group AREA-1
switch(config-router-msdp)#
```

16.4.4.11 originator-id local-interface

The **originator-id local-interface** command configures an originator ID to replace the Rendezvous Point (RP) address in Source-Address (SA) messages that it originates as an MSDP speaker.

SA messages that an MSDP speaker originates contain the speaker's rendezvous point (RP) address, as configured through PIM statements and processes. An originator ID is an alternative IPv4 address that a speaker uses in place of its RP address when advertising SA messages. This command configures the switch to use the specified interface's IP address as the RP address in SA messages that it originates.

The **no originator-id local-interface** and **default originator-id local-interface** commands configure the switch to use its RP address in SA messages that it sends by removing the **originator-id local-interface** command from ***running-config***.

Command Mode

Router MSDP Configuration

Router MSDP VRF Configuration

Command Syntax

originator-id local-interface INTERFACE

no originator-id local-interface INTERFACE

default originator-id local-interface INTERFACE

Parameters

INTERFACE Specifies the interface from which the IP address is derived. Options include:

- **ethernet *e_num*** Ethernet interface.
- **loopback *l_num*** Loopback interface.
- **management *m_num*** Management interface.
- **port-channel *p_num*** Port-Channel Interface.
- **vlan *v_num*** VLAN interface.
- **vxlan *vx_num*** VXLAN interface.

Example

These commands configure the switch to use the IP address assigned to ***interface loopback 100*** as the RP address in SA messages that it originates.

```
switch(config)# router msdp
switch (config-router-msdp)# originator-id local-interface loopback 100
switch (config-router-msdp)#
```

16.4.4.12 peer

The **peer** command configures the specified address as an MSDP peer, enables MSDP on the switch if it was not previously enabled, and places the switch in Router MSDP Peer Configuration Mode for the specified peer.

The peering session with the device at the specified network is established over a TCP connection. The **local-interface** command can specify an interface through which the TCP connection is established. When the **local-interface** command is not used to specify an interface, the connection is established through an interface determined by existing routing algorithms.

The **no peer** and **default peer** commands remove the specified MSDP peer configuration by deleting the corresponding **peer** command from ***running-config***. MSDP is disabled when the last **peer** command is removed.

Command Mode

Router MSDP Configuration

Command Syntax

peer *ip_address*

Parameters

ip_address IP address of the MSDP peer to be configured.

Commands Available in Router MSDP Peer Configuration Mode

- **default-peer**
- **description (MSDP)**
- **disabled (MSDP)**
- **keepalive (MSDP)**
- **local-interface**
- **mesh-group**
- **sa-filter in**
- **sa-filter out**
- **sa-limit**

Example

These commands establish an MSDP peer relationship with the peer at ***192.168.3.17*** and place the switch in the ***router msdp peer*** configuration mode for that peer.

```
switch(config)# router msdp
```

```
switch(config-router-msdp)# peer 192.168.3.17
switch(config-router-msdp-peer-192.168.3.17)#
```

16.4.4.13 router msdp

The **router msdp** command places the switch in the ***router msdp*** configuration mode, and allows to configure the *global* IP configuration commands and VRF commands in this mode.

The **no router msdp** and **default router msdp** commands removes the corresponding **router msdp** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

router msdp

no router msdp

default router msdp

Related Commands

- **connection retry interval**
- **default-peer**
- **description (MSDP)**
- **disabled (MSDP)**
- **group-limit**
- **ip msdp rejected-limit**
- **keepalive (MSDP)**
- **mesh-group**
- **originator-id local-interface**
- **peer**
- **sa-filter in**
- **sa-filter out**
- **sa-limit**

Example

This command places the switch in the ***router msdp*** configuration mode.

```
switch(config)# router msdp
switch(config-router-msdp)#
```

16.4.4.14 sa-filter in

The **sa-filter in** command assigns an IP Access Control List (ACL) as a filter for inbound Source-Active (SA) messages from the configuration-mode MSDP peer connection. The switch only accepts SA messages from the peer that are accepted by the assigned ACL. The switch accepts all SA messages from the peer when an ACL is not assigned as an inbound filter.

Only one ACL can be assigned as an inbound filter to an MSDP peer. Any subsequent **sa-filter in** commands for the peer replace the existing command.

The **no sa-filter in** and **default sa-filter in** commands remove the ACL assignment as an inbound filter by removing the corresponding **sa-filter in** command from ***running-config***.

Command Mode

Router MSDP Peer Configuration

Router MSDP Peer VRF Configuration

Command Syntax

`sa-filter in` list ***list_name***

`no sa-filter in`

`default sa-filter in`

Parameters

- ***peer_id*** MSDP peer address (IPv4 address).
- ***list_name*** name of ACL that filters SA messages.

Related Command

sa-filter out

Guideline

The command accepts standard and extended ACLs. The address field in a standard ACL filters an SA message on its group address.

Example

These commands create an IP ACL named LIST-IN as the inbound SA message filter for the MSDP peer connection to ***10.4.4.12***. The ACL permits SAs from the multicast group ***239.14.4.2/28***.

```
switch(config)# ip access-list LIST-IN
switch(config-acl-LIST-IN)# permit ip any 239.14.4.2/28
switch(config-acl-LIST-IN)# exit
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# sa-filter in list LIST-IN
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Listen
   Connection Source: Loopback100 (10.6.8.6)
SA Filtering:
Input Filter: LIST-IN
```

16.4.4.15 sa-filter out

The **`sa-filter out`** command assigns an IP Access Control List (ACL) as a filter for outbound Source-Active (SA) messages to the configuration-mode MSDP peer connection, after which the permit statement allows matching SAs to be advertised outbound to the peer. The **deny any/deny ip any any** at the end of an ACL statement filters any other SAs not matching explicit permit statements. The switch sends all SA messages to the peer when an ACL is not assigned as an output filter to the peer.

Only one ACL can be assigned as an outbound filter to an MSDP peer. Any subsequent **`sa-filter out`** commands for the peer replace the existing command.

The **`no sa-filter out`** and **`default sa-filter out`** commands remove the ACL assignment as an outbound filter by removing the corresponding **`sa-filter out`** command from ***running-config***.

Command Mode

Router MSDP Peer Configuration

Router MSDP Peer VRF Configuration

Command Syntax

sa-filter out list ***list_name***

no sa-filter out

default sa-filter out

Parameters

- ***peer_id*** MSDP peer address (IPv4 address).
- ***list_name*** name of ACL that filters SA messages.

Related Command

sa-filter in assigns an IP ACL to filter inbound SA messages from the MSDP peer being configured.

Guidelines

The command accepts standard and extended ACLs. The address field in a standard ACLs filters an SA message on its multicast stream source addresss.

Example

These commands assign the IP ACL named LIST-OUT as the outbound SA message filter for the MSDP peer connection to ***10.4.4.12***.

```
switch(config)# router msdp
switch(config-router-msdp)# ip access-list LIST-OUT
switch(config-acl-LIST-OUT)# permit ip any  239.14.4.2/28
switch(config-acl-LIST-OUT)# exit
switch(config)# router msdp
switch(config-router-msdp)# peer 10.4.4.12
switch(config-router-msdp-peer-10.4.4.12)# sa-filter out list LIST-OUT
switch(config-router-msdp-peer-10.4.4.12)# show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Listen
   Connection Source: Loopback100 ( 10.6.8.6 )
SA Filtering:
Output Filter: LIST-OUT
switch(config-router-msdp-peer-10.4.4.12)#
```

16.4.4.16 sa-limit

The **sa-limit** command specifies the maximum number of Source-Active messages from a specified MSDP peer that the switch allows in the SA cache. SA messages have an expiration period of ***90*** seconds, during which time they remain in the SA cache. The switch does not accept SA messages from a peer after the peer's SA limit is reached. By default, The limit to the number of SA messages that the switch can store from a specified peer is ***40000***, by default.

The **no sa-limit** and **default sa-limit** commands restore the SA limit of ***40000*** for the specified MSDP peer by removing the corresponding **sa-limit** statement from ***running-config***.

Command Mode

Router MSDP Peer Configuration

Router MSDP Peer VRF Configuration

Command Syntax

sa-limit ***quantity***

no sa-limit

default sa-limit

Parameters

- ***peer_id*** MSDP peer (IPv4 address).
- ***quantity*** maximum number of SA messages that the switch can store. Value ranges from ***0*** to ***40000***.

Example

This command sets the SA limit of ***500*** for the MSDP peer at ***10.1.1.5***.

```
switch(config)# router msdp
switch(config-router-msdp)# peer 10.1.1.5
switch(config-router-msdp-peer-10.1.1.5)# sa-limit 500
switch(config-router-msdp-peer-10.1.1.5)#
```

16.4.4.17 show ip msdp peer

The **show ip msdp peer** command displays information about specified MSDP peers. The command includes an optional parameter for displaying SAs accepted from the peer.

Command Mode

EXEC **Command Syntax**

show ip msdp peer [**PEER_ADDR**][**SA_ACCEPT**]

Parameters

- ***PEER_ADDR*** Peers for which command displays information.
  - ***no parameter*** All peers configured on the switch.
  - ***ipv4_addr*** Address of specified MSDP peer.
- **SA_ACCEPT** Command displays SAs accepted from the specified peers.
  - ***no parameter*** Accepted SAs are not displayed.
  - **accepted-sas** Accepted SAs are displayed.

Example

This command displays MSDP information concerning the peer located at ***10.2.42.4***, including SAs that the switch accepted from this peer.

```
switch(config)# show ip msdp peer 10.2.42.4 accepted-sas
MSDP Peer 10.2.42.4
Connection status:
  State: Up
  Connection Source: Loopback4 ( 10.2.43.4 )
SA Filtering:
Input Filter: allow-multicast-for-msdp
Output Filter: allow-multicast-for-msdp
SAs accepted:
(10.62.79.30, 234.1.4.2), RP 10.2.42.4
(10.61.79.29, 234.1.4.1), RP 10.2.42.4
(10.62.79.30, 234.1.4.1), RP 10.2.42.4
```

16.4.4.18 show ip msdp pim sa-cache

The **show ip msdp pim sa-cache** command displays the SA cache for the local PIM domain configured on the switch. An SA cache is a table of Source-Active messages that are generated or accepted by the PIM domain.

Command Mode

EXEC

Command Syntax

show ip msdp pim sa-cache

Example

This command displays the SA cache for the local PIM domain.

```
switch(config)# show ip msdp pim sa-cache
MSDP Source Active Messages for local Pim RP
(10.51.71.23, 234.1.4.1), RP 10.2.43.4
(10.20.91.26, 234.1.4.1), RP 10.2.43.4
(10.51.71.23, 234.1.4.2), RP 10.2.43.4
(10.20.91.21, 234.1.4.1), RP 10.2.43.4
(10.51.79.23, 234.1.4.1), RP 10.2.43.4
(10.20.91.24, 234.1.4.2), RP 10.2.43.4
(10.51.79.23, 234.1.4.2), RP 10.2.43.4
(10.20.91.21, 234.1.4.2), RP 10.2.43.4
(10.20.91.26, 234.1.4.2), RP 10.2.43.4
(10.20.91.24, 234.1.4.1), RP 10.2.43.4
```

16.4.4.19 show ip msdp sa-cache

The **show ip msdp sa-cache** command displays contents of the Source-Active (SA) message cache. The command provides these filter options for displaying partial cache contents:

- multicast group address: multicast group
- source address and group address

The command can also display unexpired SAs that were rejected by ACL filters or cache limit exceeded conditions.

Command Mode

EXEC

Command Syntax

show ip msdp sa-cache [**ADDRESS_FILTER**][**CONTENTS**]

Parameters

- **ADDRESS_FILTER** IPv4 address used to filter SA messages.
  - ***no parameter*** All SA messages.
  - ***grp_addr*** Multicast group address (IPv4 address).
  - ***src_addr grp_addr*** Source and multicast group addresses (two IPv4 addresses).
  -
- ***grp_addr*** must be a valid multicast address.
  - ***no parameter*** Displays contents of SA Cache.
  - **rejected** Displays rejected SAs in addition to the SA cache contents.
- **CONTENTS** type of SAs that the command displays.

Example

This command displays the contents of the SA message cache.

```
switch(config)# show ip msdp sa-cache
MSDP Source Active Cache
(10.61.71.29, 234.1.4.2), RP 10.5.29.4, heard from 10.5.29.4
(10.51.71.23, 234.1.4.1), RP 10.5.29.4, heard from 10.5.29.4
(10.61.79.29, 234.1.4.2), RP 10.5.29.4, heard from 10.5.29.4
(10.53.71.27, 234.1.4.2), RP 10.3.25.4, heard from 10.3.25.4
```

```
(10.10.101.24, 234.1.4.1), RP 10.2.44.4, heard from 10.2.44.4
(10.10.151.22, 234.1.4.2), RP 10.1.12.4, heard from 10.1.12.4
(10.61.71.29, 234.1.4.1), RP 10.5.29.4, heard from 10.5.29.4
(10.20.91.21, 234.1.4.1), RP 10.2.44.4, heard from 10.2.44.4
(10.61.79.29, 234.1.4.1), RP 10.2.42.4, heard from 10.2.42.4
(10.53.79.27, 234.1.4.2), RP 10.3.25.4, heard from 10.3.25.4
(10.10.151.28, 234.1.4.2), RP 10.3.25.4, heard from 10.3.25.4
(10.52.79.25, 234.1.4.2), RP 10.2.44.4, heard from 10.2.44.4
(10.52.71.25, 234.1.4.2), RP 10.2.44.4, heard from 10.2.44.4
(10.20.91.24, 234.1.4.1), RP 10.5.29.4, heard from 10.5.29.4
(10.10.151.22, 234.1.4.1), RP 10.1.12.4, heard from 10.1.12.4
```

16.4.4.20 show ip msdp sanity

The **show ip msdp sanity** command performs a consistency check between the multicast routing table and the MSDP Source-Address (SA) caches. When the command detects inconsistencies, it displays the cache entries that are not in the table.

Command Mode

EXEC

Command Syntax

show ip msdp sanity

Examples

- This command displays a sanity check that detects no inconsistencies between the SA cache and the multicast routing table.

```
switch(config)# show ip msdp sanity
PIM SA cache entries not in the MRT
Msdp-learnt MRT entries not in the SA cache
SA cache entries not in the MRT
May-Notify-MSDP entries not in the PIM SA cache
(need not be an error condition)
```

- This command displays inconsistencies between the SA cache and the multicast routing table.

```
switch(config)# show ip msdp sanity
PIM SA cache entries not in the MRT
Msdp-learnt MRT entries not in the SA cache
SA cache entries not in the MRT
(192.168.3.8, 224.1.154.1)
(192.168.3.35, 224.1.167.1)
(192.168.3.16, 224.1.226.1)
(192.168.3.19, 224.1.246.1)
(192.168.3.17, 224.1.204.1)
(192.168.3.12, 224.1.182.1)
(192.168.3.33, 224.1.150.1)
(192.168.3.26, 224.1.198.1)
(192.168.3.33, 224.1.195.1)
(192.168.3.4, 224.1.246.1)
(192.168.3.37, 224.1.188.1)
(192.168.3.12, 224.1.245.1)
(192.168.3.31, 224.1.206.1)
(192.168.3.35, 224.1.178.1)
(192.168.3.6, 224.1.155.1)
May-Notify-MSDP entries not in the PIM SA cache
(need not be an error condition)
4.1), RP 10.2.42.4
```

16.4.4.21 show ip msdp summary

The **show ip msdp summary** command displays a list of peer addresses, the operational status of the peer, and the number of Source-Active messages in the SA cache from that peer.

Command Mode

EXEC

Command Syntax

show ip msdp summary

Example

This command displays the configured peers, the status of the peers, and the number of SA message received from those peers.

```
switch(config)# show ip msdp summary
MSDP Peer Status Summary
Peer Address      State   SA Count
192.168.3.18      Up      0
192.168.3.16      Up      0
192.168.3.37      Listen  0
192.168.3.46      Up      0
192.168.3.47      Up      0
```

16.4.4.22 show msdp mesh-group

The **show msdp mesh-group** command displays the mesh group membership of MSDP peers that are configured on the switch. An MSDP mesh group is a network of MSDP speakers where each speaker is directly connected to every other speaker. The switch does not forward Source-Active (SA) messages that it receives from a mesh group peer to other peers of the same group.

Command Mode

EXEC

Command Syntax

show msdp mesh-group

Related Command

mesh-group configures the MSDP peer connection as an MSDP mesh group member.

Example

This command displays the mesh group membership of configured MSDP peers.

```
switch(config)# show msdp mesh-group
Mesh Group: tier_01
        10.24.18.13
Mesh Group: tier_02
        10.26.101.18
```

16.4.4.23 show msdp rpf-peer

The **show msdp rpf-peer** command displays MSDP information for the peer from which the switch accepts SA messages for a specified Rendezvous Point (RP).

The switch examines the BGP routing table to determine the next hop peer toward the originating RP of an SA message. This next hop peer is the Reverse Path Forwarding (RPF) peer. Because the switch receives

SA messages from the RPF peer, it forwards the message to all other MSDP peers. The switch rejects identical SA messages that it receives from a non-RPF peer.

Command Mode

EXEC

Command Syntax

show msdp rpf-peer *rp_addr*

Parameter

rp_addr PIM RP IPv4 address.

Example

This command displays MSDP information for the peer from which the switch accepts SA messages for the RP at ***10.5.29.4***.

```
switch(config)# show msdp rpf-peer 10.5.29.4
Rpf Peer is 10.5.29.4 for RP 10.5.29.4
```

16.5 Audio Video Bridging (AVB)

Arista switches support Audio Video Bridging (AVB) and the associated protocols. This section describes AVB concepts and the implementation of associated protocols.

Topics in this section include:

- AVB Overview
- AVB Protocols
- AVB Configuration
- AVB Commands

16.5.1 AVB Overview

Audio Video Bridging (AVB) is a protocol set that provides precision time synchronization, admission control, queuing reservation, and guaranteed bandwidth of professional grade quality audio and video across an IP network.

Supported AVB protocols include:

- Generalized Precision Time Protocol (gPTP)
- Multiple Stream Reservation Protocol (MSRP)
- Multiple VLAN Registration Protocol (MVRP)

These AVB features are supported on Arista 7280, 7150 Series, and 7500E Series switches:

- gPTP with hardware time stamping
- gPTP Grandmaster function
- MSRP protocol on Ethernet interfaces: stream admission control and propagation
- Control plane protection for PTP and MSRP control frames
- MVRP
- Traffic classes 2 and 3 for AVB traffic
- Traffic shaping on egress ports

These AVB features are not available on Arista switches:

- MSRP protocol on LAGs
- MSRP co-ordination with gPTP; streams are allowed even when gPTP is not in sync

- MMRP
- Signaling message support in gPTP
- Running peer delay mechanism on STP blocked ports
- Grandmaster-specific state machines (gPTP)

16.5.2 AVB Protocols

This section describes supported AVB protocols:

- gPTP
- MVRP
- MSRP
- MRP

16.5.2.1 gPTP

Generalized Precision Time Protocol (gPTP) is a network time synchronization standard for bridged Local Area Networks based on the IEEE 1588v2 Precision Time Protocol and supports the AVB protocol standards.

Time synchronization in a gPTP domain is conducted the same way as in a PTP 1588 domain. A grandmaster is selected through the best grand master clock algorithm and distributes timing synchronization information to all directly attached peers. This information is propagated across the network to provide a common time reference to all Audio and Video end stations.

16.5.2.2 MVRP

Multiple VLAN Registration Protocol (MVRP) is an application of Multiple Registration Protocol used by AVB endpoints to dynamically register and unregister VLANs on an interface.

When an interface wishes to join a VLAN advertised by an MSRP talker (to receive a stream), MVRP sends a Join message. On receiving the Join message, the interface is added to the VLAN. If the VLAN does not already exist, MVRP dynamically creates the VLAN and propagates it through the network.

MVRP events post Syslog messages, with the severity level of **INFO** for each message.

- **MVRP_VLAN_JOIN**

MVRP VLAN Join received/transmitted on an interface.

- **MVRP_VLAN_LV**

MVRP VLAN Leave received/transmitted on an interface.

- **MVRP_ERROR**

MVRP Join was discarded due to an error.

16.5.2.3 MSRP

Multiple Stream Registration Protocol (MSRP) is a signaling protocol that allows end stations (nodes) to reserve network resources and ensure QoS for communicating with other end stations.

MSRP nodes are specified as talkers or listeners:

- Talker nodes transmit multimedia streams to other nodes in the AVB network.
- Listener nodes receive multimedia streams from the AVB talker nodes.

MSRP is implemented by the switch on individual interfaces. MSRP is active when it is enabled on at least one interface, and stopped when it is disabled on all interfaces. MSRP uses Multiple Registration Protocol (MRP) to facilitate attribute registrations and distribution across connected end points in a LAN environment.

MSRP events post Syslog messages, with the severity level of INFO for each message.

- **MSRP_SR_CLASS_TRANSITION**

  MSRP SR Class state transition occurred on an interface.

- **MSRP_TALKER_ADV_JOIN**

  Talker Advertise Join message for a stream was transmitted/received on an interface.
- **MSRP_TALKER_FAIL_JOIN**

  Talker Failed Join message for a stream was transmitted/received on an interface.
- **MSRP_LISTENER_JOIN**

  Listener Join message for a stream was transmitted/received on an interface.
- **MSRP_DOMAIN_JOIN**

  Domain Join message was transmitted/received on an interface.
- **MSRP_TALKER_ADV_LV**

  Talker Advertise Leave message for a stream was transmitted/received on an interface.
- **MSRP_TALKER_FAIL_LV**

  Talker Failed Leave message for a stream was transmitted/received on an interface.
- **MSRP_LISTENER_LV**

  Listener Leave message for a stream was transmitted/received on an interface.
- **MSRP_DOMAIN_LV**

  Domain Leave message was transmitted/received on an interface.
- **MSRP_BW_ALLOC_SUCCESS**

  MSRP Bandwidth was allocated for a listener on an interface.
- **MSRP_BW_ALLOC_FAIL**

  MSRP Bandwidth could not be allocated for a listener on an interface.
- **MSRP_BW_DEALLOC**

  MSRP Bandwidth was de-allocated for a listener on an interface.
- **MSRP_ERROR**

  MSRP Join was discarded because of an error.

16.5.2.4 MRP

Multiple Registration Protocol (MRP) protocol includes MSRP and MVRP, and allows participants in an MRP application to register attributes with participants in a Bridged Local Area Network (BLAN).

16.5.3 AVB Configuration

This section describes the AVB configuration:

- Enabling gPTP
- Enabling MSRP
- Displaying MSRP Configuration and Status
- Enabling MVRP
- Displaying MVRP Configuration and Status

16.5.3.1 Enabling gPTP

Configure gPTP on the switch though the **`ptp mode gptp`** command. PTP is enabled on individual interfaces with the **`ptp enable`** command.

Example

These commands configure gPTP on the switch and enable PTP on interface ethernet ***41-45***.

```
switch(config)# ptp mode gptp
switch(config)# interface ethernet 41-45
switch(config-if-Et41-45)# ptp enable
switch(config)# show running-config
    <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ptp mode gptp
!
    <-------OUTPUT OMITTED FROM EXAMPLE-------->
end
switch(config-if-Et41-45)#show active
interface Ethernet41
   speed forced 10000full
   msrp
   ptp enable
interface Ethernet42
   speed forced 10000full
   msrp
   ptp enable
interface Ethernet43
   speed forced 10000full
   msrp
   ptp enable
interface Ethernet44
   speed forced 10000full
   ptp enable
interface Ethernet45
   ptp enable
switch(config-if-Et41-45)#
```

16.5.3.2 Enabling MSRP

MSRP is enabled on an interface with the **msrp** command.

Example

These commands enable MSRP on interface ethernet ***41-43***.

```
switch(config)# interface ethernet 41-43
switch(config-if-Et41-43)#msrp
switch(config-if-Et41-43)#show active
interface Ethernet41
   speed forced 10000full
   msrp
interface Ethernet42
   speed forced 10000full
   msrp
interface Ethernet43
   speed forced 10000full
   msrp
switch(config-if-Et41-43)#
```

16.5.3.3 Displaying MSRP Configuration and Status

MSRP configuration information and status is displayed with the **show msrp** command.

Example

This command displays the MSRP status for interfaces ethernet ***41-43***.

```
switch(config)# show msrp interfaces ethernet 41-43

  MSRP Global Status : Enabled
  Max Frame Size : 1522
  Max Fan-In Ports : No limit

                            Delta
  Class Supported Priority Bandwidth
  ----- --------- -------- ---------
      A         Y        3       75%
      B         Y        2        0%

Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

       Admin    Sr                            Talkers    Listeners    Bandwidth
  Port State    Pvid  Class  Oper State      Adv  Fail  Rdy  AskFail  Allocated
  ---- -------- ----- ------ --------------- ---- ----- ---- -------- ---------
  Et41 Active      5      A  Boundary          1     0    1        0    200kbps
                          B  Core              0     0    0        1    100kbps
  Et42 Active      3      A  Core              0     0    1        1     50kbps
                          B  WaitingForPeer    1     0    0        0     20kbps
  Et43 Disabled    3

switch(config)#
```

Stream data is available for the talker and listener. The **show msrp interfaces** command displays the status and configuration information for each stream.

Examples

- This command displays data for listener station streams on interfaces ethernet ***1-2***.

```
switch(config)# show msrp interfaces ethernet 1-2
MSRP Global Status : Enabled
  Max Frame Size : 1522
  Max Fan-In Ports : No limit

                              Delta
  Class Supported Priority Bandwidth
  ----- --------- -------- ---------
      A         Y        3       75%
      B         Y        2        0%

Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

                                  Listeners
Port     Stream Id             Dec       Dir
-------- --------------------- --------- ------
Et1      0000.0000.0000.002a   AskFail   Tx
         0000.0000.0000.029a   RdyFail   Rx
         0000.0000.0000.038f   AskFail   Rx
Et2      0000.0000.0000.002a   AskFail   Rx
         0000.0000.0000.029a   RdyFail   Rx
         0000.0000.0000.038f   AskFail   Tx

switch(config)#
```

- This command displays data for talker station streams on interfaces ethernet ***1-2***.

```
switch(config)# show msrp interfaces ethernet 1-2 talkers
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail

                                              Talkers
  Port     Stream Id            Dec       Dir    FailCode
  -------- -------------------- --------- ------ ---------
  Et1      0000.0000.0000.002a  Adv       Rx     --
           0000.0000.0000.038f  Fail      Tx     7

  Et2      0000.0000.0000.002a  Adv       Tx     --
           0000.0000.0000.038f  Adv       Rx     7

switch(config)#
```

16.5.3.4 Enabling MVRP

MVRP is disabled by default. To enable MVRP on an interface, use the **mvrp** command. MVRP is enabled globally if it is enabled on at least one interface.

Example

These commands enable MVRP on ***interface ethernet 34***.

```
switch(config)# interface ethernet 34
switch(config-if-Et34)# mvrp
switch(config-if-Et34)#
```

16.5.3.5 Displaying MVRP Configuration and Status

MVRP configuration information and status are displayed with the **show mvrp** command.

Example

This command displays the MVRP status for interfaces Ethernet ***30-40***.

```
switch(config)# show mvrp interfaces Ethernet 30-40

  MVRP Global Status : Enabled

  Port      Admin State   Registered Vlans      Declared Vlans
  --------  ------------  --------------------  --------------------
  Et30      Disabled
  Et31      Disabled
  Et32      Disabled
  Et33      Disabled
  Et34      Active
  Et35      Disabled
  Et36      Disabled
  Et37      Disabled
  Et38      Disabled
  Et39      Disabled
  Et40      Disabled
switch(config)#
```

16.5.4 AVB Commands

MSRP Commands

- msrp
- msrp streams load-file
- show msrp
- show msrp interfaces
- show msrp streams

MRP Commands

- mrp leave-all-timer
- mrp leave-timer

MVRP Commands

- show mvrp

16.5.4.1 mrp leave-all-timer

The **`mrp leave-all-timer`** command specifies the mrp leave all timer interval for the configuration mode interface.

When starting MRP, a participant starts its LeaveAll timer. Upon timer expiry, it sends a LeaveAll message and restarts its timer. When other participants receive the message, they register their attributes and restart their leave-all timers.

The default leave-all timer interval is a randomly selected value from ***10*** to ***15*** seconds. Under normal conditions, this value should not be adjusted.

The **`no mrp leave-all-timer`** and **`default mrp leave-all-timer`** commands restore the default leave-all timer interval on the configuration mode interface by removing the corresponding **`mrp leave-all-timer`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`mrp leave-all-timer` ***period***

`mrp leave-all-timer`

`default mrp leave-all-timer`

`no mrp leave-all-timer`

Parameter

period leave all timer interval (seconds). Values range from ***10*** to ***60***. Default value is a randomly selected value from ***10*** to ***15***.

Example

This command sets the MRP leave-all timer interval on ***interface ethernet 17*** to ***12*** seconds.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# mrp leave-all-timer 12
switch(config-if-Et17)#
```

16.5.4.2 mrp leave-timer

The leave-timer controls the deregistration of attributes. If an MRP participant needs other participants to unregister their attributes, it sends a Leave message. When receiving a Leave message, the Leave-timer starts and unregisters the attributes if it does not receive Join messages for the attributes before the Leave-timer expires.

The **mrp leave-timer** command specifies the mrp leave-timer interval for the configuration mode interface. The default leave-timer interval is ***0.6*** seconds. Under normal operation conditions, this value should not be adjusted.

The **no mrp leave-timer** and **default mrp leave-timer** commands restore the default leave-timer interval of ***0.6*** seconds on the configuration mode interface by removing the corresponding **mrp leave-timer** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

mrp leave-timer ***period***

no mrp leave-timer

default mrp leave-timer

Parameter

period leave all timer interval (seconds). Values range from ***10*** to ***60***. Default value is a randomly selected value from ***10*** to ***15***.

Example

This command sets the MRP leave timer interval on ***interface ethernet 17*** to ***0.8*** seconds.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# mrp leave-timer 12
switch(config-if-Et17)#
```

16.5.4.3 msrp

MSRP enables Multiple Stream Registration Protocol (MSRP), which is a signaling protocol that provides nodes with the ability to reserve network resources to ensure Quality of Service (QoS) between talker and listener endpoints. The Stream Reservation Protocol (SRP) utilizes MSRP to reserve bandwidth for data streams, and configure a complete path between endpoints.

The **msrp** command enables MSRP on the configuration mode interface. If MSRP was not previously enabled on any interface, the MSRP agent is launched by this command.

The **no msrp** and **default msrp** commands disable MSRP on the configuration mode interface, and removes the corresponding **msrp** command from ***running-config***. The command stops the MSRP agent when MSRP is no longer enabled on any interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

msrp

no msrp

default msrp

Examples

- These commands enable MSRP on ***interface ethernet 3/3/3***. Because it was not previously enabled on any other interface, the command launches the MSRP agent.

```
switch(config)# interface ethernet 3/3/3
switch(config-if-Et3/3/3)# msrp
Launching MSRP Agent
switch(config-if-Et3/3/3)# show active
 interface Ethernet3/3/3
 msrp
switch(config-if-Et3/3/3)#
```

- These commands disable the MSRP agent on ***interface ethernet 3/3/3***. Because it is not enabled on any other interface, the command stops the MSRP agent.

```
switch(config-if-Et3/3/3)# no msrp
Stopping MSRP agent
switch(config-if-Et3/3/3)# show active
interface Ethernet3/3/3
switch(config-if-Et3/3/3)# msrp
```

16.5.4.4 msrp streams load-file

The load-file for MSRP streams provides a file that contains an alias that can be substituted in the name (stream-id) of a string.

The **`msrp streams load-file`** command allows users to include a line in the file (example: ***0102.0304.0506 XYZW4*** or ***0102.0304 XYZW5***) that causes the bytes to be replaced with the accompanying string in **`show msrp streams`** commands.

The **`no msrp streams load-file`** and **`default msrp streams load-file`** commands remove the alias assignment by removing the corresponding **`msrp streams load-file`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`msrp streams load-file` [**FILE TYPE**]

`no msrp streams load-file`

`default msrp streams load-file`

Parameters

FILE TYPE The options include:

- **certificate:** device name, directory, or file name
- **extension:** device name, directory, or file name
- **file:** device name, directory, or file name
- **flash:** device name, directory, or file name
- **ftp:** device name, directory, or file name
- **http:** device name, directory, or file name
- **https:** device name, directory, or file name
- **scp:** device name, directory, or file name
- **sftp:** device name, directory, or file name
- **sslkey:** device name, directory, or file name
- **system:** device name, directory, or file name
- **terminal:** device name, directory, or file name

- **tftp:** device name, directory, or file name
- **usb1:** device name, directory, or file name

Example

This command indicates that the file named ***file1*** contains the alias names that is used in MSRP stream names.

```
switch(config)# msrp streams load-file file1
switch(config)#
```

16.5.4.5 show msrp

The **`show msrp`** command displays MSRP operational information for the specified interfaces.

Command Mode

EXEC

Command Syntax

`show msrp` [**INTERFACE_NAME**]

Parameters

INTERFACE_NAME Interface type and number. Values include:

- ***no parameter*** all Ethernet interfaces.
- interfaces ethernet ***e_range*** Ethernet interface list.
- Valid ***e_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Example

This command displays the MSRP status for interfaces ethernet ***41-43***.

```
switch(config)# show msrp interfaces ethernet 41-43

  MSRP Global Status : Enabled
  Max Frame Size : 1522
  Max Fan-In Ports : No limit

  Class Supported Priority Delta Bandwidth
  ----- --------- -------- ---------
      A         Y        3       75%
      B         Y        2        0%

Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

       Admin    Sr                          Talkers     Listeners    Bandwidth
  Port State    Pvid  Class  Oper State      Adv  Fail  Rdy  AskFail  Allocated
  ---- -------- ----- ------ --------------- ---- ----- ---- -------- ---------
  Et41 Active      5      A  Boundary          1     0    1        0    200kbps
                          B  Core              0     0    0        1    100kbps
  Et42 Active      3      A  Core              0     0    1        1     50kbps
                          B  WaitingForPeer    1     0    0        0     20kbps
  Et43 Disabled    3

switch(config)#
```

16.5.4.6 show msrp interfaces

The **`show msrp interfaces`** command displays station stream information for the specified station type, interfaces and streams.

Command Mode

EXEC

Command Syntax

`show msrp interfaces` [**INTERFACE_NAME**] **STATION_TYPE_NAME** [**STREAMS**]

Parameters

- **INTERFACE_NAME** Interface type and number. Values include:
  - ***no parameter*** all Ethernet interfaces.
  - **ethernet *e_range*** Ethernet interface list.
  - **STATION_TYPE** Endpoint type. Values include:
  - **`talker`** Command displays data for talker station streams.
  - **`listeners`** Command displays data for listener station streams.
  - **`streams`** Command displays data for talker and listener station streams.
- **STREAMS** Streams for which command displays information. Options include:
  - ***no parameter*** all streams.
  - ***stream-id hex_string*** specifies the stream command for which command displays information.
  - Valid ***e_range*** formats include number, range, or comma-delimited list of numbers and ranges.
  - Valid ***hex_string*** formats include <H>, <H.H>, <H.H.H>, or <H.H.H.H>, where H is a four-digit hex number that ranges from ***0*** to ***FFFF***.

Examples

- This command displays data for listener station streams on interfaces ethernet ***1*** and ***2***.

```
switch(config)# show msrp interfaces ethernet 1-2 listeners
Legend
------
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

                                  Listeners
  Port     Stream Id            Dec       Dir
  -------- -------------------- --------- ------
  Et1      0000.0000.0000.002a  AskFail   Tx
           0000.0000.0000.029a  RdyFail   Rx
           0000.0000.0000.038f  AskFail   Rx

  Et2      0000.0000.0000.002a  AskFail   Rx
           0000.0000.0000.029a  RdyFail   Rx
           0000.0000.0000.038f  AskFail   Tx

switch(config)#
```

- This command displays data for talker station streams on interfaces ethernet ***1*** and ***2***.

```
switch(config)# show msrp interfaces ethernet 1-2 talkers
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail

                                           Talkers
  Port     Stream Id            Dec       Dir    FailCode
  -------- -------------------- --------- ------ ---------
  Et1      0000.0000.0000.002a  Adv       Rx     --
           0000.0000.0000.038f  Fail      Tx     7

  Et2      0000.0000.0000.002a  Adv       Tx     --
           0000.0000.0000.038f  Adv       Rx     7
```

```
switch(config)#
```

16.5.4.7 show msrp streams

The **`show msrp streams`** command displays configuration and status information on the specified MSRP streams.

Command Mode

EXEC

Command Syntax

`show msrp streams` [**STREAM_NAME**] [**INFO_LEVEL**]

Parameters

- **STREAMS** Streams for which command displays information. Options include:
  - ***no parameter*** all streams.
  - **stream-id *hex_string*** specifies the stream command for which command displays information.
  - Valid ***hex_string*** formats include <H>, <H.H>, <H.H.H>, or <H.H.H.H>, where H is a four-digit hex number that ranges from ***0*** to ***FFFF***.
- **INFO_LEVEL** type of information that the command displays. Options include:
  - ***no parameter*** command displays stream identification information.
  - **detail** command displays identification and transmission characteristics.
  - **propagation** command displays ingress and egress port information.

Examples

- This command displays stream identification information.

```
switch(config)# show msrp interfaces streams
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail

  Stream Id            DMAC                Port     Dec    Vlan Class Bandwidth
  -------------------- ------------------- -------- ------ ---- ----- ----------
  0000.0000.0000.002a  00:11:22:33:44:55   Et1      Adv      24     A   8000kbps
  0000.0000.0000.029a  22:33:44:55:66:77   --       --     4095     A      0kbps
  0000.0000.0000.038f  11:22:33:44:55:66   Et2      Adv     119     B   1000kbps

switch(config)#
```

- This command displays stream identification and status information.

```
switch(config)# show msrp streams detail
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail

Stream Id            DMAC                Port     Dec    Vlan Class Bandwidth
-------------------- ------------------- -------- ------ ---- ----- ----------
  0000.0000.0000.002a  00:11:22:33:44:55 Et1      Adv    24   A     8000kbps
                 Latency (nsec): 800
                 Max Frame Size: 1522
                 Max Interval Frames: 2

  0000.0000.0000.029a  22:33:44:55:66:77 --       --     4095 A     0kbps
                 Latency (nsec): 0
                 Max Frame Size: 1100
                 Max Interval Frames: 3

switch(config)#
```

- This command displays stream ingress and egress port information.

```
switch(config)# show msrp streams propagation
```

```
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

Stream Id            DMAC                Port     Dec    Vlan Class Bandwidth
-------------------- ------------------- -------- ------ ---- ----- ----------
0000.0000.0000.002a  00:11:22:33:44:55   Et1      Adv    24   A     8000kbps

    Talker Propagation:
         Ingress      Ingress      Propagated     Propagated     Egress
         Dec          Port         Dec            Port           Dec
         --------     --------     -----------    -----------    --------
         Adv         --> Et1         --> Adv       --> Et2        --> Adv
                                                    Et4         --> Adv
                                                    Et5         --> Adv

    Listener Propagation:
         Egress       Egress       Propagated     Listener       Ingress
         Dec          Port         Dec            Port           Dec
         --------     --------     -----------    -----------    --------
         AskFail  <--  Et1         <-- AskFail   <-- Et2       <-- AskFail
                                   AskFail  <-- Et4       <-- AskFail
                                   AskFail  <-- Et5       <-- AskFail

0000.0000.0000.029a  22:33:44:55:66:77   --       --     4095  A    0kbps

    Talker Propagation:
         Ingress      Ingress      Propagated     Propagated     Egress
         Dec          Port         Dec            Port           Dec
         --------     --------     -----------    -----------    --------

    Listener Propagation:
         Egress       Egress       Propagated     Listener       Ingress
         Dec          Port         Dec            Port           Dec
         --------     --------     -----------    -----------    --------
                                   RdyFail   <-- Et1       <-- RdyFail
                                   RdyFail   <-- Et2       <-- RdyFail

0000.0000.0000.038f  11:22:33:44:55:66   Et2      Adv     119   B   1000kbps

    Talker Propagation:
         Ingress      Ingress      Propagated     Propagated     Egress
         Dec          Port         Dec            Port           Dec
         --------     --------     -----------    -----------    --------
         Adv        -->  Et2         --> Fail      --> Et1       --> Fail

    Listener Propagation:
         Egress       Egress       Propagated     Listener       Ingress
         Dec          Port         Dec            Port           Dec
         --------     --------     -----------    -----------    --------
         AskFail  <--  Et2        <--  AskFail   <-- Et1      <-- Rdy

switch(config)#
```

16.5.4.8 show mvrp

The **show mvrp** command displays MVRP operational information for the specified interfaces.

Command Mode

EXEC

Command Syntax

show mvrp [**INTERFACE_NAME**]

Parameters

INTERFACE_NAME Interface type and number. Values include:

- ***no parameter*** all Ethernet interfaces.
- **interfaces ethernet *e_range*** Ethernet interface list.
- Valid ***e_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Example

This command displays the MVRP status for interfaces Ethernet ***30-40***.

```
switch(config)# show mvrp interfaces Ethernet 30-40
MVRP Global Status : Enabled

Port      Admin State   Registered Vlans  Declared Vlans
--------  ------------  ----------------  ---------------
Et30      Disabled
Et31      Disabled
Et32      Disabled
Et33      Disabled
Et34      Active
Et35      Disabled
Et36      Disabled
Et37      Disabled
Et38      Disabled
Et39      Disabled
Et40      Disabled

switch(config)#
```

Chapter 17

Virtual Extensible LANs (VXLANs)

This chapter describes Arista's Virtual Extensible LAN (VXLAN) implementation. Sections in this chapter include:

- VXLAN Introduction
- VXLAN Description
- VXLAN Configuration
- VXLAN Commands

17.1 VXLAN Introduction

Virtual Extensible LAN (VXLAN) is a networking technology that encapsulates MAC-based Layer 2 Ethernet frames within Layer 3 UDP packets to aggregate and tunnel multiple Layer 2 networks across a Layer 3 infrastructure. VXLAN scales up to 16 million logical networks and supports Layer 2 adjacency across IP networks. Multicast transmission architecture is used for broadcast, multicast, and unknown unicast traffic.

For a list of VXLAN feature support in a specific EOS release, consult the appropriate release notes here: https://www.arista.com/en/support/software-download.

For a list of VXLAN feature support by platform in the latest EOS release, see https://www.arista.com/en/support/product-documentation/supported-features.



Note: VXLAN and NAT cannot co-exist.

17.2 VXLAN Description

These sections describe VXLAN architecture, the data objects that comprise a VXLAN network, and process of bridging packets through a VXLAN network.

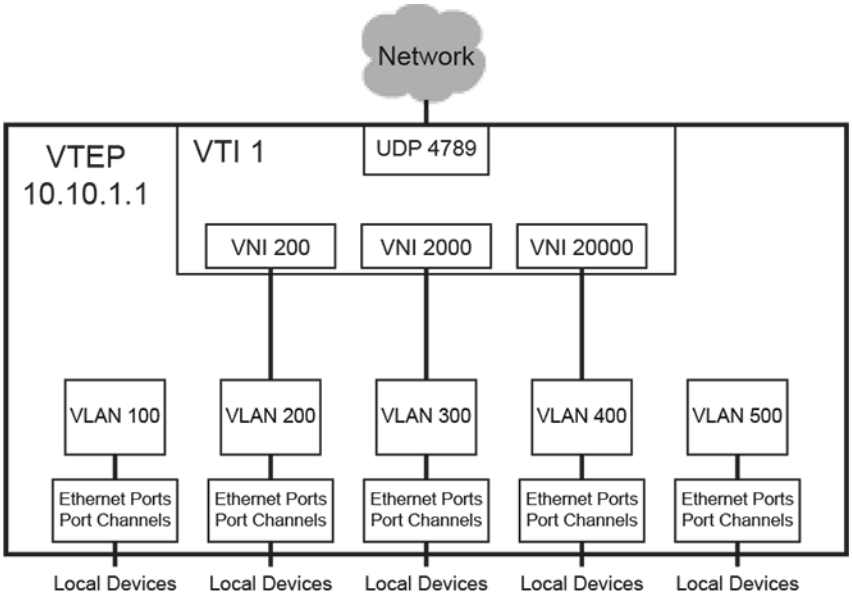
- VXLAN Architecture
- VXLAN Gateway
- VXLAN Processes
- Multicast and Broadcast over VXLAN
- VXLAN and MLAG
- VXLAN Bridging and Routing Support
- Data Structures

17.2.1 VXLAN Architecture

The VXLAN architecture extends a Layer 2 network by connecting VLANs from multiple hosts through UDP tunnels called VXLAN segments. VXLAN segments are identified by a 24-bit Virtual Network Identifier (VNI). Within a host, each VLAN whose network is extended to other hosts is associated with a VNI. An extended Layer 2 network comprises the devices attached to VLANs from all hosts that are on VLANs that are associated with the same VNI.

The following figure displays the data objects that comprise a VXLAN implementation on a local host.

Figure 17-1: VXLAN Architecture



- **VXLAN Tunnel End Point (VTEP):** a host with at least one VXLAN Tunnel Interface (VTI).
- **VXLAN Tunnel Interface (VTI):** a switchport linked to a UDP socket that is shared with VLANs on various hosts. Packets bridged from a VLAN to the VTI are sent out the UDP socket with a VXLAN header. Packets arriving on the VTI through the UDP socket are demuxed to VLANs for bridging.
- **Virtual Network Identifier (VNI):** a 24-bit number that distinguishes between the VLANs carried on a VTI. It facilitates the multiplexing of several VLANs over a single VTI.

VNIs can be expressed in digital or dotted decimal formats. VNI values range from ***1*** to ***16777215*** or from ***0.0.1*** to ***255.255.255***.

The network in the figure above has the following assignments:

- VTEP IP address of ***10.10.1.1***.
- UDP port of ***4789***.
- One VTI that supports three VXLAN segments (UDP tunnels): VNI ***200***, VNI ***2000***, and VNI ***20000***
- Five VLANs, of which three VLANs can communicate with remote devices over Layer 2.

17.2.2 VXLAN Gateway

A VXLAN gateway is a service that exchanges VXLAN data and packets with devices connected to different network segments. VXLAN traffic must pass through a VXLAN gateway to access services on physical devices in a distant network.

A VXLAN gateway requires the following information:

- An IP address that is designated as the VXLAN interface source.
- VLAN to VNI mapping.
- VTEP list for each VNI.
- A method for handling broadcast, unknown unicast, and multicast (BUM) packets.

Arista switches manually perform VXLAN gateway services. The switch connects to VXLAN gateways that serve other network segments. MAC address learning is performed in hardware from inbound VXLAN packets.

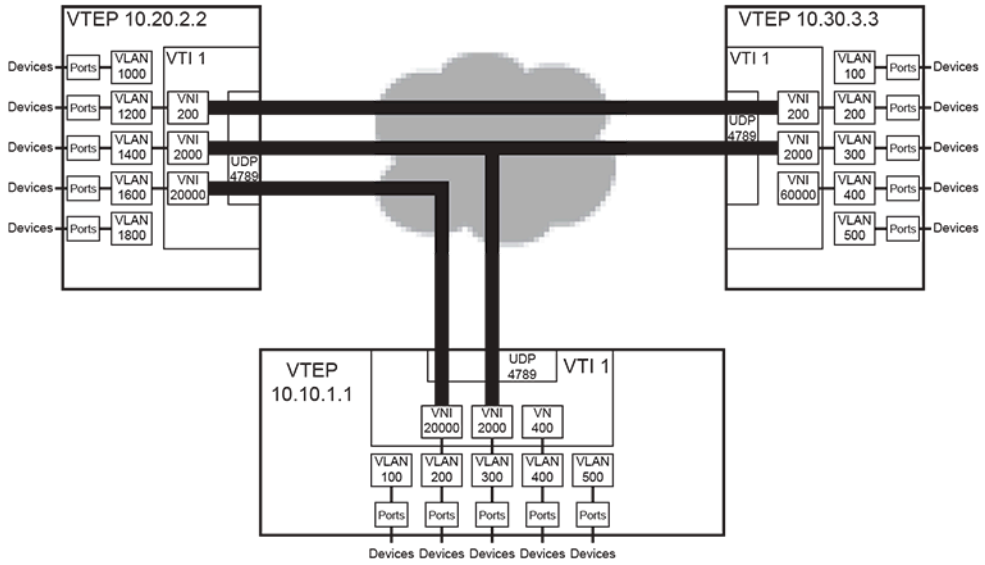
17.2.3 VXLAN Processes

When a packet enters a VLAN from a member (ingress) port, the VLAN learns the source address by adding an entry to the MAC address table that associates the source to the ingress-port. The VLAN then searches the table for destination address. If the MAC address table lists the address, the packet is sent out the corresponding port. If the MAC address table does not lists the address, the packet is flooded to all ports except the ingress port.

VXLANs extend VLANs through the addition of a VXLAN address table that correlates remote MAC addresses to their port and resident host IP address. Packets that are destined to a remote device are sent to the VXLAN tunnel interface (VTI), which is the switchport that is linked to the UDP socket. The packet is encapsulated with a VXLAN header which includes the VNI associated with the VLAN and the IP mapping of the destination host. The packet is sent through a UDP socket to the destination VTEP IP. The VTI on the remote host extracts the original packet and bridges it to the VLAN associated with the VNI on the remote host.

UDP port 4789 is recognized as the VXLAN socket and listed as the destination port on the UDP packets. The UDP source port field is filled with a hash of the inner header to facilitate load balancing.

Figure 17-2: VXLAN Implementation



The figure above displays a configuration that includes three VTEPs. The VXLAN defines three inter-host L2 networks. The VLANs that comprise the networks include:

- ***VNI 200***: ***VTEP 10.20.2.2: VLAN 1200*** and ***VTEP 10.30.3.3: VLAN 200***
- ***VNI 2000***: ***VTEP 10.10.1.1: VLAN 300***, ***VTEP 10.20.2.2: VLAN 1400***, and ***VTEP 10.30.3.3: VLAN 300***
- ***VNI 20000***: ***VTEP 10.10.1.1: VLAN 200***, and ***VTEP 10.20.2.2: VLAN 1600***

VXLAN Routing

VXLAN routing is enabled by creating a VLAN interface on the VXLAN-enabled VLAN and assigning an IP address to the VLAN interface. The IP address serves as VXLAN gateway for devices that are accessible from the VXLAN-enabled VLAN.

17.2.4 Multicast and Broadcast over VXLAN

These sections describe multicast and broadcast over VXLANs. Multicast packet flooding describes broadcast and multicast transmission by associating a multicast group to a VTI through a configuration command.

Head-end Replication (HER) optimizes flooding of inter VTEP broadcast, unknown unicast and broadcast (BUM) traffic by using hardware and flood lists to perform replication on the supported platform.

- Multicast Packet Flooding
- Head-end Replication

17.2.4.1 Multicast Packet Flooding

Multicast packet flooding is supported with VXLAN bridging without MLAG. A VTI is associated with a multicast group through a configuration command.

VXLAN and Broadcast

When a VLAN receives or sends a broadcast packet the VTI is treated as a bridging domain L2 interface. The packet is sent from this interface on the multicast group associated with the VTI. The VTIs on remote VTEPs that receive this packet extract the original packet, which is then handled by the VLAN associated with the packet's VNI. The VLAN floods the packet, excluding the VTI. When the broadcast results in a response, the resulting packet can be unicast back to the originating VTEP because the VXLAN address table obtained the host MAC to VTEP association from the broadcast packet.

VXLAN and Multicast

A VTI is treated as an L2 interface in the VLAN for handling multicast traffic, which is mapped from the VLAN to the multicast group associated with the VTI. All VTEPs join the configured multicast group for inter-VTEP communication within a VXLAN segment; this multicast group is independent of any other multicast groups that the hosts in the VLAN join.

The IP address space for the inter-host VXLAN communication may be sourced from a different VRF than the address space of the hosts in the VLAN. The multicast group for inter-VTEP transmissions must not be used for other purposes by any device in the VXLAN segment space.

17.2.4.2 Head-end Replication

Head-end replication uses a flood list to support broadcast, unknown unicast, and multicast (BUM) traffic over VXLAN. The flood list specifies a list of remote VTEPs. The switch replicates BUM data locally for bridging across the remote VTEPs specified by the flood list. This data flooding facilitates remote MAC address learning by forwarding data with unknown MAC addresses.

Head-end replication is required for VXLAN routing and to support VXLANs over MLAG.

17.2.5 VXLAN and MLAG

VXLAN over MLAG provides redundancy in hardware VTEPs. VTI configuration must be identical on each MLAG peer for them to act as a single VTEP. This also prevents the remote MAC from flapping between the remote VTEPs by ensuring that the rest of the network sees a host that is connected to the MLAG interface as residing behind a single VTEP.

Differences between VXLAN bridging and routing implementations over MLAG are applicable for the DCS-7050X series platform.

- VXLAN routing recirculates a packet twice, with the first iteration performing the routing action involving an L2 header rewrite, and the second recirculation performing VXLAN encap and decap operations. Recirculation is achieved by MAC loopback on dedicated loopback interfaces.
- The configuration for VXLAN routing on an MLAG VTEP includes separate Recirc-Channel configuration on both peers. The virtual IP, virtual MAC, and virtual VARP VTEP IP addresses are identical on both peers.

The following VTI elements must be configured identically on both MLAG peers:

- VLAN-VNI mappings

- VTEP IP address of the source loopback interface
- Flood VTEP list used for head-end replication

If OSPF is also in use, configure the OSPF router ID manually to prevent the switch from using the common VTEP IP address as the router ID.

The following rules are observed by MLAG switches so that they behave as a single VXLAN VTEP:

- Only the MLAG peer that receives a packet performs VXLAN encapsulation on it.
- Packets are not VXLAN encapsulated if they are received from the peer link.
- If a packet is decapsulated and sent over the peer link, it should not be flooded to active MLAG interfaces.
- If a packet is sent over the peer link to the CPU, it is not head-end replicated to other remote VTEPs.
- If a packet's destination is the VTEP IP address, it is terminated by the MLAG peer that receives it.

Examples

- These commands complete the configuration required for a VXLAN routing deployment.

```
switch(config)# interface Vxlan1
switch(config-if-Vx1)# vxlan source-interface Loopback0
switch(config-if-Vx1)# vxlan udp-port 4789
switch(config-if-Vx1)# vxlan vlan 2417 vni 8358534
switch(config-if-Vx1)# vxlan flood vtep 1.0.1.1 1.0.2.1
switch(config-if-Vx1)# interface Vlan2417
switch(config-if-Vl2417)# ip address 1.0.4.1/24
switch(config-if-Vl2417)# interface Loopback0
switch(config-if-Lo0)# ip address 1.0.1.1/32
switch(config-if-Lo0)# ip routing
switch(config)# interface Recirc-Channel627
switch(config-if-Re627)# switchport recirculation features vxlan
switch(config-if-Re627)# interface Ethernet 1
switch(config-if-Et1)# traffic-loopback source system device mac
switch(config-if-Et1)# channel-group recirculation 627
switch(config-if-Et1)# exit
switch(config)# interface Ethernet 2
switch(config-if-Et2)# traffic-loopback source system device mac
switch(config-if-Et2)# channel-group recirculation 627
switch(config-if-Et2)#
```

- **`show running interface Loopback`** allows remote VTEP tunnels to be routed over L3 interfaces.

```
switch# show running interface Loopback298
interface Loopback298
   ip address 1.0.1.1/32
switch#
switch# show running interface Ethernet54/1.4095
interface Ethernet54/1.4095
   mtu 9214
   encapsulation dot1q vlan 267
   ip address 1.0.4.1/24
switch#
switch# show running interface Port-Channel1.4095
interface Port-Channel1.4095
   mtu 9214
   encapsulation dot1q vlan 1043
   ip address 1.0.88.1/24
switch#
switch# show running interface Vxlan1
interface Vxlan1
   vxlan source-interface Loopback298
   vxlan udp-port 4789
   vxlan vlan 2156 vni 15613244
```

```
   vxlan vlan 2393 vni 3610141
   vxlan vlan 2156 flood vtep 1.0.2.1 1.0.3.1
   vxlan vlan 2393 flood vtep 1.0.2.1 1.0.3.1
switch#
switch# show port-channel
Port Channel Port-Channel1:
  Active Ports: Ethernet51/1
switch#
switch# show ip route
VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
     . . .
     . . .
 S        1.0.2.1/32 [1/0] via 1.0.4.2, Ethernet54/1.4095
 S        1.0.3.1/32 [1/0] via 1.0.88.2, Port-Channel1.4095
```

There are 2 remote VTEPs configured ***1.0.2.1***, and ***1.0.3.1***. The remote ***VTEP 1.0.2.1*** is reachable through ***Ethernet54/1.4095*** and remote ***VTEP 1.0.3.1*** is reachable through ***port-channel1.4095***.

Configuring Unconnected Ethernet Interfaces for Recirculation

On systems where bandwidth is not fully used by the front panel ports, unused bandwidth is used for recirculation.

The following example is applicable to the DCS-7050X series platform.

Example

These commands expose unconnected Ethernet interfaces which are used for recirculation, in order to use them to replace or use along with front panel Ethernet interfaces.

```
switch(config)# service interface unconnected expose
switch(config)# interface UnconnectedEthernet 2
switch(config-if-Ue2)# traffic-loopback source system device mac
switch(config-if-Ue2)# channel-group recirculation 627
```

The following example enables display of the inactive interfaces using the show command.

Example

```
switch(config)# service interface inactive expose
```

Running a show command generates the following output:

```
switch(config)# switch(config)#show int et21/1-4 stat
```

```
Port    Name   Status    Vlan  Duplex Speed  Type      Flags Encapsulation
Et21/1         connected 1     full   100G   100GBASE-CR4
Et21/2         inactive  1     full   25G    100GBASE-CR4
Et21/3         inactive  1     full   25G    100GBASE-CR4
Et21/4         inactive  1     full   25G    100GBASE-CR4
```

On previous releases, Ethernet 21/2, 21/4 do not exist and the output would be the following:

```
switch(config)# switch(config)#show int et21/1-4 stat
```

```
Port    Name   Status    Vlan  Duplex Speed  Type      Flags Encapsulation
Et21/1         connected 1     full   100G   100GBASE-CR4
```

```
Et21/3          inactive  1     full    25G     100GBASE-CR4
```

17.2.6 VXLAN Bridging and Routing Support

Describes the support of VXLAN Bridging and Routing on the R3 series of DCS 7280, 7500, and 7800 Arista switches.

17.2.6.1 Differences with DCS-7500R2 Implementation

The following are notable differences with respect to implementation of VXLAN on the R3 Series of switches.

- There is no need to configure the VXLAN-routing TCAM profile to enable VXLAN routing on the R3 Series switches. The command is still accepted for backward compatibility reasons.
- CPU bound traffic after VXLAN decapsulation (such as routing protocol packets) use the same CoPP queues used by the non VXLAN decapsulated packets. This is an improvement over the R2 series behavior where the CPU bound traffic after VXLAN decapsulation took a different CoPP queue that was shared with other IP Unicast packets.

17.2.6.2 Limitations

There is no EVPN VXLAN Multicast (Type 6/7/8 NLRI) support.

17.2.7 Data Structures

VXLAN implementation requires two VXLAN tables and a MAC address table accommodation.

17.2.7.1 MAC Address Table VXLAN Support

MAC address table entries correlate MAC addresses with the port upon which packets arrive. In addition to Ethernet and port channels, the port column may specify a VTI for packets that arrive on a VLAN from a remote port through the VXLAN segment.

17.2.7.2 VTEP-MAC Address Table

VTEP-MAC address table entries correlate MAC address with the IP address of the VTEP from where packets bearing the MAC address arrive. The VTI uses this table to determine the destination address for packets that are sent to remote hosts.

17.2.7.3 VNI-VLAN Map

The VNI-VLAN map displays the one-to-one correspondence between the VNIs assigned on the switch and the VLANs to which they are assigned. Each VNI can be assigned to only one VLAN; each VLAN can be assigned a maximum of one VNI. Each VNI-VLAN assignment constitutes a VXLAN segment.

17.3 VXLAN Configuration

These sections describe VXLAN configuration tasks:

- Configuring the VTI
- Head End Replication Configuration
- VXLAN Routing Configuration
- Configuring VXLAN Routing with Overlay VRFs
- Configuring VXLAN over MLAG
- Configuring VXLAN Control Service
- Configuring VXLAN Multicast Decapsulation
- VXLAN Rules Support for Mirror ACLs Configuration
- Configuring EVPN VXLAN

- Displaying VXLAN Configuration
- Displaying VXLAN Bridging and Routing Support

17.3.1 Configuring the VTI

Configuring the VTI enables VXLAN bridging and is a requirement for VXLAN Routing. The following sections describe the steps required to enabling VXLAN bridging by bringing up the VXLAN line protocol. **VXLAN Routing Configuration** describes the additional steps required to enable VXLAN routing.

Instantiating the VTI and VXLAN Configuration Mode

The **interface vxlan** command places the switch in VXLAN-interface configuration mode for modifying the specified VXLAN Tunnel Interface (VTI). The command also instantiates the interface if it was not previously created.

VXLAN interface configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Example

These commands create VXLAN tunnel interface ***1***, place the switch in VXLAN-interface configuration mode, and display parameters of the new VTI.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

Assigning an IP address to the VTEP

The **vxlan source-interface** command specifies the loopback interface from which the VTEP derives the source address (IP) that it uses when exchanging VXLAN frames. This address is used by UDP headers to specify source and destination addresses of hosts that send or receive VXLAN encapsulated packets.

There is no default source interface assignment. A valid VXLAN configuration requires the assignment of a loopback interface to the VTEP and the assignment of a valid IP address to the specified interface.

Example

These commands configure VTI 1 to use IP address ***10.25.25.3*** (***interface loopback 15***) as the source interface in the encapsulation fields of outbound VXLAN frames.

```
switch(config)# interface loopback 15
switch(config-if-Lo15)# ip address 10.25.25.3/24
switch(config-if-Lo15)# exit
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan source-interface loopback 15
switch(config-if-Vx1)#  show active
interface Vxlan1
   vxlan source-interface Loopback15
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

Assigning a UDP Port to the VTEP

Packets bridged to the VTI from a VLAN are encapsulated with a VXLAN header, then sent through a pre-configured UDP port. Packets that arrive through this port are assumed to be VXLAN encapsulated and sent to the bridging domain of the recipient VLAN as determined by the VNI in the VXLAN header and the VNI-VLAN map.

The **vxlan udp-port** command associates a UDP port with the configuration mode VXLAN Interface (VTI). By default, UDP ***port 4789*** is associated with the VTI.

Note: UDP ***port 4789*** is reserved by convention for VXLAN usage. Under most typical applications, this parameter should be set to the default value.

Examples

- This command associates UDP ***port 5500*** with ***interface vxlan 1***.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan udp-port 5500
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 5500
switch(config-if-Vx1)#
```

- This command resets the ***interface vxlan 1*** UDP port association of ***4789***.

```
switch(config-if-Vx1)# no vxlan udp-port
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

Assigning a VNI to a VLAN

When a VLAN bridges a packet to the VTI, the packet is encapsulated with a VXLAN header that includes the VNI associated with the VLAN. Packets that arrive on the VTI's UDP socket are bridged to the VLAN that is associated with the VNI specified by the VXLAN header that encapsulates the packet.

The VTI requires a one-to-one correspondence between specified VLANs and VNI values. Commands that assign a new VNI to a previously configured VLAN replace existing VLAN assignment statements in **`running-config`**. Commands that attempt to assign a VNI value to a second VLAN generate a CLI error.

The **vxlan vlan vni** command associates a VLAN ID with a Virtual Network Identifier (VNI).

Example

These commands associate ***vlan 100*** to ***vni 100*** and ***vlan 200*** to ***vni 10.10.200***.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan vlan 100 vni 100
switch(config-if-Vx1)# vxlan vlan 200 vni 10.10.200
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 200 vni 658120
   vxlan vlan 100 vni 100
switch(config-if-Vx1)# vxlan vni notation dotted
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 100 vni 0.0.100
   vxlan vlan 200 vni 10.10.200
switch(config-if-Vx1)#
```

Verifying the VXLAN Configuration

The **`show interface vxlan 1`** displays the configuration and connection status of the VXLAN.

Example

This command indicates that the VXLAN line protocol status is ***up*.**

```
switch(config-if-Vx1)# show interface vxlan 1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback15 and is active with 10.25.25.3
  Static vlan to vni mapping is
    [100, 0.0.100]    [200, 10.10.200]
switch(config-if-Vx1)#
```

17.3.2 Head End Replication Configuration

Head-end replication is a data distribution method that supports broadcast, unknown unicast traffic over VXLANs by replicating BUM data locally for transmission to the set of remote VTEPs specified by a flood list. This data flooding facilitates remote MAC address learning through the forwarding of data with unknown MAC addresses.

Each **vxlan flood vtep** statement in ***running-config*** associates a set of VTEP addresses to an access VNI. A default flood list is also configurable that applies to all VNIs for which a flood list is not configured.

The VTEP flood list is created and modified through the **vxlan flood vtep** command.

Examples

- These commands create a default VXLAN head-end replication flood list.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan flood vtep 10.1.1.1 10.1.1.2
switch(config-if-Vx1)# show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- These commands create VXLAN head-end replication flood lists for the VNIs accessed through ***vlan 101*** and ***vlan 102***.

```
switch(config-if-Vx1)# vxlan vlan 101-102 flood vtep 11.1.1.1 11.1.1.2
 11.1.1.3
switch(config-if-Vx1)# show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan vlan 101 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan vlan 102 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

17.3.3 VXLAN Routing Configuration

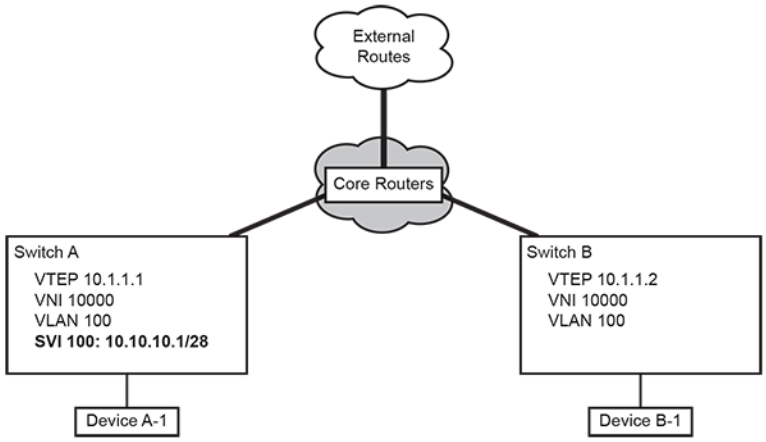
- Implementing VXLAN Routing
- Configuring Direct VXLAN Routing
- Configuring VXLAN VTEP counters
- VXLAN Auto Flood-List Construction

17.3.3.1 Implementing VXLAN Routing

VXLAN routing is enabled by creating a VLAN Interface (SVI) on a VLAN that is associated to a VNI. In the figure below, VXLAN routing is enabled on ***Switch A*** by configuring a VLAN interface with an IP address of

10.10.10.1. Packets from Devices ***A-1*** and ***B-2*** that have destinations other than ***10.10.10.0/28*** are VXLAN-bridged to the default gateway (***10.10.10.1***), then routed from ***Switch A***.

Figure 17-3: Implementing VXLAN Routing



Note: For R and R2 series Arista platforms, VXLAN routing must be enabled in hardware with the **`hardware tcam profile vxlan-routing`** command. This command will cause a brief data-plane interruption. It should be run while the switch is in maintenance mode, or in an interval when a brief data-plane interruption is acceptable.

```
switch(config)#hardware tcam profile vxlan-routing
switch(config)#
```

Note: For Trident2 and some Tomahawk platforms, VXLAN routing requires that recirculation channels be configured with the command **`channel-group recirculation`**.

```
switch(config)#channel-group recirculation 1
switch(config)#
```

Example

These commands configure ***Switch A*** to perform VXLAN routing. The example includes OSPF routing that is used for underlay routing.

```
switch-A(config)# route-map vxlanvlan permit 10
switch-A(config-route-map-vxlanvlan)# match interface loopb5
switch-A(config-route-map-vxlanvlan)# exit
switch-A(config)# route-map vxlanvlan permit 20
switch-A(config-route-map-vxlanvlan)# match interface vlan 100
switch-A(config-route-map-vxlanvlan)# exit
switch-A(config)# router ospf 1
switch-A(config-router-ospf)# redistribute connected route-map vxlanvlan
switch-A(config-router-ospf)# exit
switch-A(config)# interface loopback 5
switch-A(config-if-Lo5)# ip address 10.25.25.3/24
switch-A(config-if-Lo5)# exit
switch-A(config)# interface vxlan 1
switch-A(config-if-Vx1)# vxlan source-interface loopback 5
switch-A(config-if-Vx1)# vxlan vlan 100 vni 10000
switch-A(config)# interface vlan 100
switch-A(config-if-Vl100)# ip address 10.10.10.1/28
switch-A(config-if-Vl100)# exit
```

17.3.3.2 Configuring Direct VXLAN Routing

Figure **Implementing VXLAN Routing** , VXLAN routing is enabled on ***Switch A*** only; ***Switch B*** supports VXLAN bridging. Traffic from ***Switch B*** devices to the external routes must go through the core route twice: once as they are bridged to is VXLAN gateway and once when routed to its next hop device.

Direct VXLAN routing with VXLAN enabled addresses this issue by configuring each VTEP with all VLANs. This allows packets to be VXLAN-bridged to a local VTEP and routed to remote VTEPs. Indirect routing scales well but is complex to engineer efficiently, and naked routing provides the same scalability to indirect routing. Direct routing leads to the most efficient traffic flows, with the number of virtual subnets or virtual machines increasing at scale, and is thereby optimal from a data plane viewpoint.

The following sections describe conventions required to implement Direct VXLAN Routing, then presents a direct VXLAN routing implementation.

Configuring VARP addresses

For direct routing, an anycast IP address is used as the gateway address on the SVI for a VLAN on all hardware VTEPs associated with that VLAN.

Examples

- These commands configure an IP virtual-router and virtual MAC address.

```
switch(config)# interface Vlan2417
switch(config-if-Vl2417)# ip address 1.0.4.50/24
switch(config-if-Vl2417)# ip virtual-router address 1.0.4.1
switch(config-if-Vl2417)# ip virtual-router mac-address 00:00:11:11:22:22
switch(config)#
```

- These commands configure an IP virtual address (instead of IP virtual-router address) for the VLAN SVI, and a secondary address on the loopback interface for the virtual VTEP IP. The virtual VTEP IP is the logical VTEP hosting the virtual MAC address.

```
switch(config)# interface Vlan2417
switch(config-if-Vl2417)# ip address virtual 1.0.4.1/24
switch(config-if-Vl2417)# exit
switch(config)# interface Loopback0
switch(config-if-Lo0)# ip address 1.0.1.1/32
switch(config-if-Lo0)# ip address 1.0.1.2/32 secondary
switch(config-if-Lo0)# ip virtual-router mac-address 00:00:11:11:22:22
switch(config)#
```

Virtual IP and MAC Addresses

Virtual-router IP addresses can be configured on VLAN interfaces in addition to a primary address. All VTEPs in a direct VXLAN network can be configured with the same virtual router address. This allows devices to use a common IP address as their VXLAN gateway.

The **ip address virtual** command configures a specified address as the primary IPv4 address and as a virtual IP address for the configuration mode VLAN interface. This results in the virtual MAC address (**ip virtual-router mac-address**) assignment to the VLAN interface. In large VXLAN networks, using distinct primary IP addresses for each VTEP limits the number addresses on its subnet for connected hosts. Defining a common virtual IP address for all VTEPs and using that their primary addresses conserves subnet addresses

Example

These commands specify a virtual router address of ***00:00:00:00:00:48*** for the switch and, for ***vlan 100***, a primary address of ***10.10.10.10/28*** and a virtual IP address of ***10.10.10.10***.

```
switch(config)# ip virtual-router mac-address 00:00:00:00:00:48
switch(config)# interface vlan 100
```

```
switch(config-if-Vl100)# ip address virtual 10.10.10.10/28
switch(config-if-Vl100)# show active
 interface Vlan100
   ip address virtual 10.10.10.10/28
switch(config-if-Vl100)#
```

Virtual VTEP Configuration

A virtual VTEP address is specified by configuring a secondary address on the loopback interface designated as the VXLAN's source interface. All VTEPs in the direct routing topology share the same virtual VTEP address.

You must also configure the secondary VTEP IP on the flood-list of the downstream VXLAN VTEPS as shown below.

Example

These commands specify a primary (***10.1.1.1***) and virtual VTEP address (***10.2.2.2***).

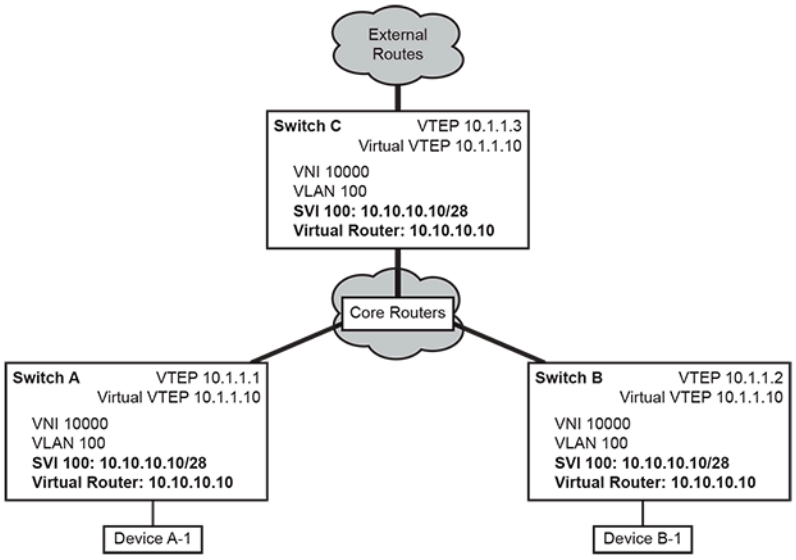
```
switch1
switch(config)# interface loopback 5
switch(config-if-Lo5)# ip address 10.1.1.1/24
switch(config-if-Lo5)# ip address 10.2.2.2/24 secondary
switch(config-if-Lo5)# show active
 interface Loopback5
   ip address 10.1.1.1/24
   ip address 10.2.2.2/24 secondary
switch(config-if-Lo5)# exit
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan source-interface loopback 5
switch(config-if-Vx1)# show active
 interface Vxlan1
   vxlan source-interface Loopback5
   vxlan udp-port 4789
   vxlan vlan 100 vni 10000
switch(config-if-Vx1)#

switch2
switch(config)# interface vxlan1
switch(config-if-Vx1)# vxlan flood vtep 10.1.1.1
switch(config-if-Vx1)# vxlan flood vtep 10.2.2.2
```

Direct VXLAN Topology

The following figure displays a direct VXLAN topology, where each VTEP is configured with the same set of VNIs, VLAN interfaces, and virtual VTEP address.

Figure 17-4: Direct VXLAN Routing



Example

These commands configure VXLAN parameters for ***Switch-A***.

```
switch-A(config)# route-map vxlanvlan permit 10
switch-A(config-route-map-vxlanvlan)# match interface loopb5
switch-A(config-route-map-vxlanvlan)# exit
switch-A(config)# route-map vxlanvlan permit 20
switch-A(config-route-map-vxlanvlan)# match interface vlan 100
switch-A(config-route-map-vxlanvlan)# exit
switch-A(config)# router ospf 1
switch-A(config-router-ospf)# redistribute connected route-map vxlanvlan
switch-A(config-router-ospf)# exit
switch-A(config)# ip virtual-router mac-address 00:00:00:00:00:48
switch-A(config)# interface loopback 5
switch-A(config-if-Lo5)# ip address 10.1.1.3/24
switch-A(config-if-Lo5)# ip address 10.1.1.10/24 secondary
switch-A(config-if-Lo5)# exit
switch-A(config)# interface vxlan 1
switch-A(config-if-Vx1)# vxlan source-interface loopback 5
switch-A(config-if-Vx1)# vxlan vlan 100 vni 10000
switch-A(config)# interface vlan 100
switch-A(config-if-Vl100)# ip address virtual 10.10.10.10/28
switch-A(config-if-Vl100)# exit
```

17.3.3.3 Configuring VXLAN VTEP Counters

The VXLAN VTEP counters feature enables a device to count VXLAN packets received and sent by the device on a per VTEP basis. Specifically, it enables the device to count bytes and packets that are getting encapsulated and decapsulated as they are passing through.

The counters are logically split up in the two VXLAN directions. Encapsulated on the device and directed to the core, “encap” counters count packets coming from the edge. Decapsulated on the device and heading towards the edge, “decap” counters count packets coming from the core.

To be able to count VXLAN packets the device has to support VXLAN and have a VXLAN interface correctly configured.

Examples

- This command configures the enabling of VXLAN VTEP counters for encap.

```
switch(config)# hardware counter feature vtep encap
switch(config)#
```

- This command configures the disabling of VXLAN VTEP counters for encap.

```
switch(config)# no hardware counter feature vtep encap
switch(config)#
```

- This commands configures the enabling of VXLAN VTEP counters for decap.

```
switch(config)# hardware counter feature vtep decap
switch(config)#
```

- This commands configures the disabling of VXLAN VTEP counters for decap.

```
switch(config)# no hardware counter feature vtep decap
switch(config)#
```

17.3.3.4 VXLAN Auto Flood-List Construction

With the introduction of wireless Access Points (APs), VXLAN flood-lists learned from the data-plane is added to or removed from the flood-lists created in the control-plane. When a VXLAN packet is received on a new VNI from a VTEP, it is added to the dynamic flood-list for that VNI and the flood-list is merged with flood-lists from other sources. When all MACs behind a remote VTEP have been removed through aging, for example, the remote VTEP is removed from all dynamic VXLAN flood-lists.

To restrict VTEPs from being added to dynamic flood-lists, when VXLAN traffic is received from untrusted sources, use the **vxlan learn-restrict** command. MAC learning is disabled from the specified IP ranges. The learning restrictions is placed on all platforms including APs.

VXLAN Configuration for Learning Data-plane Flood-lists

The following example is applicable to all platforms.

These commands enable VXLAN flood-lists learning from data-plane.

```
switch(config)# interface Vxlan1
switch(config-if-Vx1)# vxlan flood vtep learned data-plane
```

The following example restricts learning from VTEPs not in a prefix range.

```
switch(config-if-Vx1)# vxlan learn-restrict vtep <prefixes>
```

The following example restricts learning to VTEPs with IP in range.

```
switch(config-if-Vx1)# vxlan learn-restrict vtep 1.1.1.1/24
```

The following command shows the VXLAN flood-lists programmed in hardware.

```
switch(config)# switch(config)#show vxlan flood vtep
```

The following command shows the dynamic VXLAN flood-lists.

```
switch(config)# switch(config)#show l2Rib input vxlan-dynamic
```

The following command shows the VXLAN flood-lists sent to platform.

```
switch(config)# switch(config)#show l2Rib output floodset
```

The following command shows the VXLAN learning restrictions for all VLANs.

```
switch(config)# switch(config)#show vxlan learn-restrict vtep
```

The following command shows the VXLAN learning counters for all VLANs.

```
switch(config)# switch(config)#show vxlan counters learn-restrict all
```

17.3.4 Configuring VXLAN Routing with Overlay VRFs

VXLAN SVIs configured in non-default VRFs are supported with VXLAN routing using overlay VRFs. Overlay SVIs are configured in non-default VRFs but underlay SVIs, which provide IP connectivity between VTEPs, must remain in the default VRF. VXLAN routing is deployable by allowing users to configure separate overlay routing domains using VRFs per tenant, thereby allowing support for overlapping IP addresses in the overlay. This provides separation between overlay and underlay traffic, including simpler and cleaner protocol configuration, without using complicated route-maps to control distribution of prefixes to peers in the overlay VRFs and underlay SVIs. IPv4 based VXLAN routing is currently supported.

17.3.5 Configuring VXLAN over MLAG

VTI configuration must be identical on each MLAG peer for them to act as a single VTEP.

The following VTI elements must be configured identically on both MLAG peers:

VLAN-VNI Mappings

Configure identical VLAN to VNI mappings on both MLAG peers using the **vxlan vlan vni** command.

Example

These commands associate ***vlan 100*** to ***vni 100*** and ***vlan 200*** to ***vni 10.10.200***.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan vlan 100 vni 100
switch(config-if-Vx1)# vxlan vlan 200 vni 10.10.200
switch(config-if-Vx1)#
```

VTEP IP Address of the Source Loopback Interface

Configure the same VTEP IP address for the source loopback interface on both MLAG peers using the **vxlan source-interface** command.

Example

These commands configure a primary VTEP address.

```
switch(config)# interface loopback 5
switch(config-if-Lo5)# ip address 10.1.1.1/24
switch(config-if-Lo5)# exit
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan source-interface loopback 5
```

```
switch(config-if-Vx1)#
```

Flood VTEP List

Configure the same VTEP flood list on both MLAG peers using the **vxlan flood vtep** command.

Example

These commands create a default VXLAN head-end replication flood list.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan flood vtep 10.1.1.1 10.1.1.2
switch(config-if-Vx1)#
```

OSPF Configuration

If OSPF is in use, configure the OSPF router ID using the **router-id (OSPFv2)** command to prevent the switch from using the common VTEP IP address as the router ID.

Example

These commands assign ***10.0.0.1*** as the OSPFv2 router ID.

```
switch(config)# router ospf 100
switch(config-router-ospf)# router-id 10.0.0.1
switch(config-router-ospf)#
```

17.3.6 Configuring VXLAN Control Service

The VXLAN Control Service (VCS) provides a mechanism by which hardware VTEPs share states between each other in order to establish VXLAN tunnels, without the need for a multicast control plane. This feature enables the use of a VCS client.

Examples

- These commands connect a switch to the VCS running on CVX. The server host IP address is the management IP address of the CVX controller or the IP address that CVX is listening on for client connections.

  ```
  switch(config)# management cvx
  switch(config-mgmt-cvx)# server host 172.27.6.248
  switch(config-mgmt-cvx)# no shutdown
  switch(config-mgmt-cvx)#
  ```

- These commands configure the VXLAN interface, except for the multicast group configuration, in order to learn from the controller.

  ```
  switch(config)# interface vxlan 1
  switch(config-if-Vx1)# vxlan controller-client
  switch(config-if-Vx1)#
  ```

17.3.7 Configuring VXLAN Multicast Decapsulation

VXLAN multicast decapsulation enables VTEPs that support Head End Replication (HER). Multicast encapsulated Broadcast/Unknown/Multicast (BUM) packets terminate VTEPs from remote VTEPs that do not support HER.

Examples

- These commands enable VXLAN multicast decapsulation.

```
switch(config)# interface vxlan 1
switch(config-config-if-Vx1)# vxlan multicast-group decap 230.1.1.1
switch(config-config-if-Vx1)#
```

- These commands disable VXLAN multicast decapsulation.

```
switch(config)# interface vxlan 1
switch(config-config-if-Vx1)# no vxlan multicast-group decap 230.1.1.1
switch(config-config-if-Vx1)#
```

17.3.8 VXLAN Rules Support for Mirror ACLs Configuration

VXLAN rules support for mirror ACLs configuration permit VXLAN deep inspection rules to be specified in the mirroring ACLs when the switch is operating in normal mode.

Examples

The following are examples of VXLAN rules specified in mirroring ACLs.

- These commands permit all VXLAN traffic (udp protocol and destination ***port 4789***).

```
switch(config)# ip access-list miracl
switch(config-acl-miracl)# permit vxlan any any
switch(config-acl-miracl)#
```

- These commands permit VXLAN traffic with ***vni 1001*** only.

```
switch(config)# ip access-list miracl
switch(config-acl-miracl)# permit vxlan any any vni 1001 0x000000
switch(config-acl-miracl)#
```

- These commands deny VXLAN traffic with ***vni 0x1000*** through ***0x100f***.

```
switch(config)# ip access-list miracl
switch(config-acl-miracl)# permit vxlan any any vni 0x1000 0x100f
switch(config-acl-miracl)#
```

17.3.9 Configuring EVPN VXLAN

Supported Configurations

- Static EVPN VXLAN Configuration
- VXLAN Bridging and Routing Configuration
- EVPN VXLAN All Active Multihoming
- EVPN VXLAN Single-Active Multihoming
- VARP and Virtual VTEP with VXLAN Routing
- Overlay Multicast using VXLAN Underlay Multicast Tree
- Bridging Over EVPN IPv6 VXLAN Underlay

17.3.9.1 Static EVPN VXLAN Configuration

```
switch(config)# service routing protocols model multi-agent
switch(config)# interface Loopback0
switch(config-if-Lo0)# ip address 172.16.1.1/32
!
switch(config)# interface Vxlan1
switch(config-if-Vx1)# vxlan source-interface Loopback0
```

```
switch(config-if-Vx1)# vxlan udp-port 4789
switch(config-if-Vx1)# vxlan vrf test vni 12345
!
switch(config)# ip routing vrf test
switch(config)# Ipv6 unicast-routing vrf test
!
switch(config)# ip route vrf test 192.168.1.0/24 vtep 10.1.1.2 vni 20000
 router-mac-address 00:00:78:01:00:00
switch(config)# ipv6 route vrf test 1:0:5::0/64 vtep 10.1.1.2 vni 30000 router-
mac-address 00:00:80:01:00:00
```

17.3.9.2 VXLAN Bridging and Routing Configuration

```
switch(config)# interface Loopback0
switch(config-if-Lo0)# ip address 172.16.1.1/32
!
switch(config)# ip virtual-router mac-address 00:02:03:04:05:06
!

switch(config)# ip routing
!
switch(config)# interface Vxlan1
switch(config-if-Vx1)# vxlan source-interface Loopback0
switch(config-if-Vx1)# vxlan udp-port 65330
switch(config-if-Vx1)# vxlan vlan 300 vni 945438
switch(config-if-Vx1)# vxlan vlan 200 vni 654677
switch(config-if-Vx1)# vxlan flood vtep 172.16.1.2 172.16.1.3 172.16.1.1
```

17.3.9.3 EVPN VXLAN All Active Multihoming

Multi-homing is activated in an EVPN environment by assigning an ethernet segment identifier to the participating Ethernet or Port-Channel interfaces.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# evpn ethernet-segment
switch(config-evpn-es)# identifier 00aa:bbbb:cccc:dddd:eeee
switch(config-evpn-es)# route-target import 12:23:34:45:56:67
```

The optional **`designated-forwarder election hold-time`** command can configure a wait time before selecting the designated forwarder and allow potential forwarders a chance to advertise their EVPN ethernet segment (type 4) routes. The default hold time is three (***3***) seconds, as specified in section 8.5 of ***RFC7432*** [1].

The route target configured here is the ES import route target described in section 7.6 of ***RFC7432*** [1]. It can be set to any MAC address, but for each Ethernet segment every participating interface in the network must use the same ES import route target. A suggested value is the MAC address of the CE connected to the multi-homing PEs via this interface.

17.3.9.4 EVPN VXLAN Single-Active Multihoming

Multi-homing allows in an EVPN environment by assigning an ethernet segment identifier or a single Customer Edge (CE) to the participating multiple Provider Edge (PE). The default mode of operation is All-active. Introduced in the ***EOS 4.26.0F*** for VxLAN, singe-active is another mode of operation in which only one PE per VLAN accepts traffic for that ethernet segment.

Single-active multihoming is useful for:

- Manually controlled traffic flows
- Prioritizing links over others

- Connecting separate CE devices to a single ethernet segment
- Connecting a CE that does not support link aggregation to multiple PEs.

To configure single-active multi-homing, use the **redundancy single-active** command on a physical ethernet or aggregate Port-channel interface.

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# evpn ethernet-segment
switch(config-evpn-es)# identifier 0123:0123:0123:0123:0123
switch(config-evpn-es)# route-target import 12:34:12:34:12:34
switch(config-evpn-es)# redundancy single-active
```

When **don't preempt** mode is enabled, a flag bit is included with preference value. Each VLAN specifies ***high/low*** rule with preference-based DF election. The default election rule is ***high*** and the default preference is ***32767*** from ***0*** to ***65535***.

```
interface Port-Channel1
   switchport mode trunk
   switchport trunk allowed vlan 100-200
   evpn ethernet-segment
      identifier 0123:0123:0123:0123:0123
      route-target import 12:34:12:34:12:34
      redundancy single-active
      designated-forwarder election algorithm preference 10000 [dont-preempt]

router bgp 10
   vlan 100
      designated-forwarder election preference rule low
      ...
   vlan-aware-bundle red
      designated-forwarder election preference rule low
      vlan 120-140
      ...
```

Show commands

show bgp evpn instance command takes the name of a configured EVPN instance to limit the output for that instance.

```
switch# show bgp evpn instance vlan 10
EVPN instance: VLAN 10
  Route distinguisher: 10.255.0.0:10
  Route target import: Route-Target-AS:64500:10
  Route target export: Route-Target-AS:64500:10
  Service interface: VLAN-based
  Local IP address: 10.255.0.0
  Encapsulation type: VXLAN
  Local ethernet segment:
    ESI: 0011:1111:1111:1111:1111
      Interface: Ethernet6
      Mode: single-active
      State: up
      ES-Import RT: 00:01:00:01:00:01
      DF election algorithm: preference
      Designated forwarder: 10.255.0.0
      Non-Designated forwarder: 10.255.0.1
```

Each ethernet segment shows the modes, single-active or all-active, the DF election algorithm, the elected designated forwarder and all other candidate forwarders.

When a port/VLAN is inactive, it is not shown by **show vlan** command. It is possible to see configured but inactive VLANs in **show vlan configured** command.

```
switch# show vlan configured
VLAN  Name                             Status Ports
----- -------------------------------- --------- -------------------------
-----
1  default                          active Et1, Et2, Et4, Et5, Et6
10 VLAN0010                         active Et6, Vx1
11 VLAN0011                         active Et6#, Vx1

# indicates a port on which traffic is currently being blocked
```

show bgp evpn detail command shows the EVPN routes contributing to the multihoming state of a device with **route-type ethernet-segment** and any other appropriate filters.

```
switch# show bgp evpn route-type ethernet-segment esi 0011:1111:1111:1111:1111
 detail
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
BGP routing table entry for ethernet-segment 0011:1111:1111:1111:1111
 10.255.0.0, Route Distinguisher: 10.255.0.0:1
 Paths: 1 available
  Local
 - from - (0.0.0.0)
   Origin IGP, metric -, localpref -, weight 0, valid, local, best
   Extended Community: TunnelEncap:tunnelTypeVxlan EvpnEsImportRt:00:01:00:01
:00:01
DF Election: Preference 200
BGP routing table entry for ethernet-segment 0011:1111:1111:1111:1111
 10.255.0.1, Route Distinguisher: 10.255.0.1:1
 Paths: 1 available
  303 301
 10.255.0.1 from 10.0.0.2 (0.0.1.1)
   Origin IGP, metric -, localpref 100, weight 0, valid, external, best
   Extended Community: TunnelEncap:tunnelTypeVxlan EvpnEsImportRt:00:01:00:01
:00:01
DF Election: Preference 100
```

Limitations

- Single-active multihoming with MPLS is not supported.
- Single-active redundancy is currently only supported on trunk ports. Access ports will not drop traffic when inactive.
- Designated forwarder can not be reset in non-revertive mode.

17.3.9.5 VARP and Virtual VTEP with VXLAN Routing

```
interface Loopback0
   ip address 172.16.1.1/32
   ip address 20.0.0.1/32 secondary
!
ip virtual-router mac-address 00:02:03:04:05:06
!
ip routing
!
interface Vlan200
   ipv6 address 2000:0:0:41::2/64
   ip address virtual 1.0.7.1/24
   ipv6 virtual-router address 2000:0:0:41::1
```

```
!
interface Vxlan1
   vxlan source-interface Loopback0
   vxlan udp-port 65330
   vxlan vlan 300 vni 945438
   vxlan vlan 200 vni 654677
   vxlan flood vtep 172.16.1.2 172.16.1.3 172.16.1.1 20.0.0.1
```

17.3.9.6 Overlay Multicast using VXLAN Underlay Multicast Tree

To inject a source route, configure the **ip multicast source route export** command on the incoming interface.

```
switch(config)# interface Vlan10
switch(config-Vl10)# ip pim sparse-mode
switch(config-Vl10)# ip multicast source route export
```

To redistribute the source routes in the MRIB via BGP while running multi-agent protocol model, configure the **redistribute attached-host** command for the IPv4 multicast address-family. Activate the neighbor to establish a BGP connection.

```
switch(config-router-bgp)# address-family ipv4 multicast
switch(config-router-bgp-af)# neighbor 3.0.0.2 activate
switch(config-router-bgp-af)# redistribute attached-host
```

To redistribute the source routes in the URIB via BGP while running ribd protocol model, configure the **redistribute attached-host** command under the ***router bgp*** mode.

```
switch(config-router-bgp)# redistribute attached-host
```

This following is a sample configuration for a VTEP for the setup above using multi-agent protocol model.

```
switch(config)# service routing protocol model multi-agent

switch(config)# ip pim rp-address 15.15.15.15 225.1.1.1/32

switch(config)# interface Loopback0
switch(config-if-Lo0)# ip address 1.1.1.1/32

switch(config)# interface vxlan1
switch(config-if-Vx1)# vxlan source-interface Loopback0
switch(config-if-Vx1)# vxlan vlan10 vni 10000

! Interface to the underlay
switch(config)# interface Ethernet1
switch(config-if-Et1)# ip address 3.0.0.1/24
switch(config-if-Et1)# ip pim sparse-mode

switch(config)# interface vlan10
switch(config-if-Vl10)# ip address 10.1.1.1/24
switch(config-if-Vl10)# ip pim sparse-mode
switch(config-if-Vl10)# ip multicast source route export

switch(config)# router bgp 10
switch(config-router-bgp)# router-id 0.0.0.2

switch(config-router-bgp)# address-family ipv4 multicast
switch(config-router-bgp-af)# neighbor 3.0.0.2 activate
switch(config-router-bgp-af)# redistribute attached-host
```

This following is a sample configuration for a VTEP for the setup above using the ribd protocol model.

```
switch(config)# service routing protocol model ribd

switch(config)# ip pim rp-address 15.15.15.15 225.1.1.1/32

switch(config)# interface Loopback0
switch(config-if-Lo0)# ip address 1.1.1.1/32

switch(config)# interface vxlan1
switch(config-if-Vx1)# vxlan source-interface Loopback0
switch(config-if-Vx1)# vxlan vlan10 vni 10000

! Interface to the underlay
switch(config)# interface Ethernet1
switch(config-if-Et1)# ip address 3.0.0.1/24

switch(config-if-Et1)# ip pim sparse-mode

switch(config)# interface vlan10
switch(config-if-Vl1)# ip address 10.1.1.1/24
switch(config-if-Vl1)# ip pim sparse-mode
switch(config-if-Vl1)# ip multicast source route export

switch(config)# router bgp 10
switch(config-router-bgp)# router-id 0.0.0.2
switch(config-router-bgp)# redistribute attached-host
```

17.3.9.7 Bridging Over EVPN IPv6 VXLAN Underlay

The following example configuration is for VXLAN bridging over EVPN IPv6 VXLAN underlay.

```
switch(config)# interface loopback 0
switch(config-if-Lo0)# ip address 20001::100/128
!
switch(config)# vlan 10
switch(config-vlan-10)#
switch(config)# vlan 20
switch(config-vlan-20)#
!
switch(config)# hardware tcam
switch(config-tcam)# system profile vxlan-v6-underlay
!
switch(config)# interface Ethernet1
switch(config-if-Et1)# switchport access vlan 10
switch(config)# interface Ethernet2
switch(config-if-Et2) #switchport access vlan 20
!
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan source-interface loopback 0
switch(config-if-Vx1)# vxlan encapsulation ipv6
switch(config-if-Vx1)# vxlan vlan 10 vni 10
switch(config-if-Vx1)# vxlan vlan 20 vni 20
!
```

17.3.10 Displaying VXLAN Configuration

The following section describes the commands that control the display format of VNIs and the commands that list VXLAN configuration and transmission information.

Configuring VNI Display Format

The **vxlan vni notation dotted** command configures the switch to display VNIs in dotted decimal notation. VNI values range from ***1*** to ***16777215*** in decimal notation and from ***0.0.1*** to ***255.255.255*** in dotted decimal notation.

The command affects the VNI number display in all **show** commands, including **show running-config**. Commands that include VNI as a parameter may use decimal or dotted decimal notion regardless of the setting of this command. By default, show commands display VNI number in decimal notation.

Examples

- These commands configure the switch to display vni numbers in dotted decimal notation, then displays a configuration that includes a VNI setting.

```
switch(config)# vxlan vni notation dotted
switch(config)# interface vxlan 1
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 333 vni 3.4.5
switch(config-if-Vx1)#
```

- These commands configure the switch to display vni numbers in decimal notation, then displays a configuration that includes a VNI setting.

```
switch(config)# no vxlan vni notation dotted
switch(config)# interface vxlan 1
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 333 vni 197637
switch(config-if-Vx1)#
```

MAC Address Table

The MAC address table indicates a MAC address from a device on a remote host by indicating Vx interface as the port that corresponds to the address.

Example

The **show mac address-table** command displays a MAC address table that includes entries of devices from remote hosts by specifying Vx1 as the corresponding port.

```
switch> show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   1    0050.5682.6725    DYNAMIC     Et16       1       0:02:01 ago
   1    0050.568e.58e9    DYNAMIC     Et23       2       0:08:53 ago
   1    0050.56a0.474a    DYNAMIC     Et16       1       0:18:04 ago
  51    0000.0051.0004    DYNAMIC     Et5        1       12 days, 1:02:44 ago
  51    0000.0051.0005    DYNAMIC     Et5        1       12 days, 1:02:44 ago
  51    0000.0051.0101    DYNAMIC     Vx1        1       12 days, 0:17:30 ago
  51    0000.0051.0102    DYNAMIC     Vx1        1       12 days, 0:17:30 ago
  61    0000.0061.0005    DYNAMIC     Et5        1       12 days, 1:02:44 ago
Total Mac Addresses for this criterion: 8

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0 switch>
```

VXLAN MAC Address Table

VXLAN MAC address table entries correlate MAC addresses accessible through remote VTEPs with the local VLAN and the IP address of the VTEP through which the addressed device is accessed. The VTI uses this table when constructing the VXLAN encapsulation to specify the destination IP address of the recipient VTEP and the VNI segment through which the device's remote VLAN is accessed.

The **show vxlan address-table** command displays the VXLAN MAC address table.

Example

This command displays the VXLAN address table.

```
switch> show vxlan address-table
          Vxlan Mac Address Table
----------------------------------------------------------------------

Vlan  Mac Address     Type      Prt  Vtep             Moves   Last Move
----  -----------     ----      ---  ----             -----   ---------
  51  0000.0051.0101  DYNAMIC   Vx1  10.25.2.12       1       4 days, 0:37:14 ago
  51  0000.0051.0102  DYNAMIC   Vx1  10.25.2.12       1       4 days, 0:37:14 ago
  51  0000.0051.0103  DYNAMIC   Vx1  10.25.2.12       1       4 days, 0:37:14 ago
  51  0000.0051.0104  DYNAMIC   Vx1  10.25.2.12       1       4 days, 0:37:14 ago
  51  0000.0051.0105  DYNAMIC   Vx1  10.25.2.12       1       4 days, 0:37:14 ago
  61  0000.0061.0103  DYNAMIC   Vx1  10.25.2.12       1       4 days, 0:37:14 ago
  61  0000.0061.0104  DYNAMIC   Vx1  10.25.2.12       1       4 days, 0:37:14 ago
  61  0000.0061.0105  DYNAMIC   Vx1  10.25.2.12       1       4 days, 0:37:14 ago switch>
```

VXLAN MAC Address Table

The **show vxlan vtep** command displays information about remote VTEPs that the configured VTI has discovered and with whom it has exchanged packets.

Example

These commands display the VTEPs that have exchanged data with the configured VTI.

```
switch> show vxlan vtep
Remote vteps for Vxlan1:
10.52.2.12
Total number of remote vteps:  1 switch>
```

VXLAN Counters

The **clear vxlan counters** command resets the VXLAN counters. The **show vxlan counters** command displays the VXLAN counters.

Example

This command displays the VXLAN counters

```
switch> show vxlan counters software
encap_bytes:3452284
encap_pkts:27841
encap_read_err:1
encap_discard_runt:0
encap_discard_vlan_range:0
encap_discard_vlan_map:0
encap_send_err:0
encap_timeout:1427
decap_bytes_total:382412426
decap_pkts_total:2259858
decap_bytes:0
decap_pkts:0
decap_runt:0
decap_pkt_filter:45128
```

```
decap_bytes_filter:5908326
decap_discard_vxhdr:0
decap_discard_vlan_map:2214730
decap_timeout:0
decap_sock_err:1 switch>
```

17.3.11 Displaying VXLAN Bridging and Routing Support

All show commands applicable to prior VXLAN implementations on R2 series are also available on R3 series for VXLAN debugging.

The **show interfaces vxlan** command displays operational status and configuration information of the specified VXLAN.

```
switch(config)# show interfaces vxlan 1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback0 and is active with 172.16.1.1
  Replication/Flood Mode is headend with Flood List Source: CLI
  Remote MAC learning via Datapath
  VNI mapping to VLANs
  Static VLAN to VNI mapping is
    [100, 100]
  Note: All Dynamic VLANs used by VCS are internal VLANs.
        Use 'show vxlan vni' for details.
  Static VRF to VNI mapping is not configured
  Headend replication flood vtep list is:
   100 172.16.1.2 10.1.1.1
  MLAG Shared Router MAC is 0000.0000.0000
  VTEP address mask is Non
```

The **show arp** command displays all ARP tables on the configured VXLAN.

```
switch(config)# show arp interface vxlan 1
Address         Age (sec)  Hardware Addr   Interface
192.168.10.1              -  0000.abab.abab  Vlan100, Vxlan1
```

The **show arp interface summary** command displays a summary of all ARP tables on the configured VXLAN.

```
switch(config)# show arp interface vxlan 1 summary
Total: 1
Static: 1
Dynamic: 0
Not learned: 0
```

The **show vxlan counters software** command displays the VXLAN software counters.

```
switch(config)# show vxlan counters software
Rx bytes for encapsulation                          :  0
Rx pkts for encapsulation                           :  0
Rx high priority bytes for encapsulation            :  0
Rx high priority pkts for encapsulation             :  0
Rx low priority bytes for encapsulation             :  0
Rx low priority pkts for encapsulation              :  0
…..
```

```
switch(config)# show vxlan vni
VNI to VLAN Mapping for Vxlan1
VNI       VLAN       Source       Interface         802.1Q Tag
```

```
--------- ---------- ------------ ----------------- ----------
100       100        static       Ethernet2/1       untagged
                                  Vxlan1            100

Note: * indicates a Dynamic VLAN
```

The **show vxlan vtep** command displays information about remote VTEPs that the configured VTI has discovered and with whom it has exchanged packets.

```
switch(config)# show vxlan vtep
Remote VTEPS for Vxlan1:
10.1.1.1
Total number of remote VTEPS:  1
```

```
switch(config)# show platform fap vxlan vtep encapsulation
Tunnel Type: R(Vxlan-Routing), B(Vxlan-Bridging)
D - ECMP is divergent across switching chips
 ------------------------------------------------------------------------------------------------------------
|                                                VTEP Table                                                  |
|------------------------------------------------------------------------------------------------------------|
|                      FEC                              |                      EEDB                        |
|------------------------------------------------------------------------------------------------------------|
|  Destination  | Ecmp|  Fec|Tunnel|Tunnel|  Arp|SIP|TTL| Cmd |     Destination    | VID |   MAC / CPU Code  |
|               |Index|Index| Index| Type |Index|Idx|   |     |                    |     |                   |
|------------------------------------------------------------------------------------------------------------|
|       10.1.1.1|  -  |353900| 16382|     B|65536|  0| 64|ROUTE| Et1/1              |1006 | 00:00:aa:aa:aa:aa |
|       10.1.1.1|  -  |353901| 16383|     R|65536|  0| 64|ROUTE| Et1/1              |1006 | 00:00:aa:aa:aa:aa |
```

```
switch(config)# show cpu counters queue | grep Vxlan
CoppSystemVxlanEncap                0           0           0           0
CoppSystemVxlanVtepLearn            0           0           0           0
CoppSystemVxlanEncap                0           0           0           0
CoppSystemVxlanVtepLearn            0           0           0           0
```

```
switch(config)#show platform fap vxlan mapping vni
    VNI      | VSI
-------------+------
         100 |  100
```

```
switch# show platform pkt | egrep -i "vxlan|vni"
rxpacllog 0 rxracllog 0 rxvteplearn 0 rxvxlan_encap 0
rx_vxlanbfd 0 rxcfm 0
rxvteprestore_drop 0 rxvxlan_encap_drop 0 rxmpc_nodev 0 rx_vxlanbfderr 0
 rx_nonvxlan_arp_drop 0
fab.rxvxlan_decaperr 0 rx_macsecproxyerr 0 rx_macsecproxy_prune 0
CpuCodeVxlanVtepLearn:  0
CpuCodeVxlanEncapRequired:      0
CpuCodeVxlanArp:        0
CpuCodeVxlanUnknownVtepArp:     0
     vxlan :  sys_port   -1 traffic_class 0 fdma - fapid 0 sflow_cookie 0 mark4
 0000 mark6 0000 D
vxlan vni hashtable:
h: 201, i: 0, vni: 100, vlanid: 100
vxlan enabled vlans: 100,
```

use the

```
switch# show cpu counters vxlan l2 ecmp
VTEP Group     Member VTEP IP     ECMP    ECMP      Member    Next Level
ID                           Size    FEC ID    FEC ID    FEC ID
-------------------------------------------------------------------------
```

```
1          172.16.1.2          2        1     91752     353907
                10.1.1.1                          91753     353908
```

17.4 VXLAN Commands

VXLAN Global Configuration Commands

- interface vxlan
- ip address virtual
- vxlan vni notation dotted

VXLAN Interface Configuration Commands

- vxlan flood vtep
- vxlan multicast-group decap
- vxlan source-interface
- vxlan udp-port
- vxlan vlan vni

VXLAN Bridging and Routing Commands

- designated-forwarder election hold-time
- redistribute attached-host

VXLAN Display and Clear Commands

- clear vxlan counters
- show arp
- show interfaces vxlan
- show service vxlan
- show vxlan address-table
- show vxlan counters
- show vxlan flood vtep
- show vxlan vtep

17.4.1 clear vxlan counters

The **`clear vxlan counters`** command resets the VXLAN counters.

Command Mode

Privileged EXEC

Command Syntax

`clear vxlan counters` **ROUTE_TYPE**

Parameters

ROUTE_TYPE Specifies the type of VXLAN counter reset by the command.

- **software** Command resets software counters.
- **varp** Command resets virtual-ARP counters.

Related Command

show vxlan counters displays the VXLAN counters.

Example

This command resets the VXLAN counters

```
switch# clear vxlan counters software
switch# show vxlan counters software
encap_bytes:0
encap_pkts:0
encap_read_err:0
encap_discard_runt:0
encap_discard_vlan_range:0
encap_discard_vlan_map:0
encap_send_err:0
encap_timeout:0
decap_bytes_total:0
decap_pkts_total:0
decap_bytes:0
decap_pkts:0
decap_runt:0
decap_pkt_filter:0
decap_bytes_filter:0
decap_discard_vxhdr:0
decap_discard_vlan_map:0
decap_timeout:0
decap_sock_err:0 switch#
```

17.4.2 designated-forwarder election hold-time

The optional **designated-forwarder election hold-time** command can configure a wait time before selecting the designated forwarder and allow potential forwarders a chance to advertise their EVPN ethernet segment (type 4) routes. The **no** and **default** forms of the command removes the election hold time.

Command Mode

EVPN Ethernet segment identifier configuration mode

Command Syntax

designated-forwarder election hold-time ***sec***

no designated-forwarder election hold-time ***sec***

default designated-forwarder election hold-time ***sec***

Parameter

sec Number of seconds for the timer waiting to receive updates from other PE. Range ***0-1800***.

Example

```
switch(config)# interface Ethernet1
switch(config-if-Et1)# evpn ethernet-segment
switch(config-evpn-es)# designated-forwarder election hold-time 20
```

17.4.3 interface vxlan

The **interface vxlan** command places the switch in VXLAN-interface configuration mode for modifying the specified VXLAN Tunnel Interface (VTI). The command also instantiates the interface if it was not previously created.

VXLAN interface configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The **no interface vxlan** deletes the specified VTI interface, including its configuration statements, from ***running-config***. The **default interface vxlan** command removes all configuration statements for the specified VTI from ***running-config*** without deleting the interfaces.

Command Mode

Global Configuration

Command Syntax

interface vxlan ***vx_range***

no interface vxlan ***vx_range***

default interface vxlan ***vx_range***

Parameter

vx_range VXLAN interface number. The only permitted value is ***1***.

Commands Available in link-flap Configuration Mode

- **vxlan source-interface**
- **vxlan udp-port**
- **vxlan vlan vni**

Examples

- These commands create ***interface vxlan 1***, place the switch in ***vxlan-interface*** configuration mode, then display parameters of the new VTI.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- This command exits ***vxlan-interface*** configuration mode, placing the switch in ***global*** configuration mode.

```
switch(config-if-Vx1)# exit
switch(config)#
```

17.4.4 ip address virtual

The **ip address virtual** command configures a specified address as the primary IPv4 address and as a virtual IP address for the configuration mode VLAN interface. The address resolves to the virtual MAC address configured through the **ip virtual-router mac-address** command. The command includes a subnet designation that is required in primary IP address assignments.

This command is typically used in VXLAN routing configurations as an alternative to assigning a unique IP address to each VTEP. All existing IPv4 addresses must be removed from the interface before executing this command.

The **no ip address virtual** and **default ip address virtual** commands remove the IPv4 address and virtual IP assignment from the configuration mode interface by deleting the **ip address virtual** command from ***running-config***.

Removing the IPv4 address assignments from an interface disables IPv4 processing on that port.

Command Mode

Interface-VLAN Configuration

Command Syntax

ip address virtual ***ipv4_subnet*** **| secondary**

no ip address virtual

default ip address virtual

Parameters

- ***ipv4_subnet*** IPv4 and subnet address (CIDR or address-mask notation).
- **secondary** Configures a secondary address on the loopback interface designated as the VXLAN's source interface.

Related Commands

- **ip address**
- **ip virtual-router mac-address**

Examples

This command configures ***10.10.10.1*** as the IPv4 address and virtual address for ***vlan 100***.

```
switch(config-if-Vl100)# show active
 interface Vlan100
   ip address virtual 10.10.10.1/28
switch(config-if-Vl100)#
```

These commands configure ***10.1.1.1*** as the primary address and ***10.2.2.2*** as the virtual VTEP address.

```
switch1
switch(config)# interface loopback 5
switch(config-if-Lo5)# ip address 10.1.1.1/24
switch(config-if-Lo5)# ip address 10.2.2.2/24 secondary
switch(config-if-Lo5)# show active
  interface Loopback5
    ip address 10.1.1.1/24
    ip address 10.2.2.2/24 secondary
switch(config-if-Lo5)# exit
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan source-interface loopback 5
switch(config-if-Vx1)# show active
  interface Vxlan1
    vxlan source-interface Loopback5
    vxlan udp-port 4789
    vxlan vlan 100 vni 10000
switch(config-if-Vx1)#

switch2
switch(config)# interface vxlan1
switch(config-if-Vx1)# vxlan flood vtep 10.1.1.1
switch(config-if-Vx1)# vxlan flood vtep 10.2.2.2
```

17.4.5 redistribute attached-host

Use the **redistribute attached-host** to redistribute the source routes in the MRIB via BGP while running multi-agent protocol model for the IPv4 multicast address-family. Thius activates the neighbor to establish a BGP connection.

Command Mode

BGP router address-family configuration mode

Command Syntax

redistribute attached-host route-map ***name***

no redistribute attached-host route-map ***name***

default redistribute attached-host route-map ***name***

Parameter

route-map *name* Name of the route map.

Example

```
switch(config-router-bgp)# address-family ipv4 multicast
switch(config-router-bgp-af)# neighbor 3.0.0.2 activate
switch(config-router-bgp-af)# redistribute attached-host
```

17.4.6 show arp

Use the **show arp** command to display and modify entries in the Address Resolution Protocol (ARP) cache.

Command Mode

EXEC

Command Syntax

show arp [agent [ipv4 | ipv6] | host ***name*** | ***A.B.C.D*** [host | interface | mac-address | summary] | interface [Ethernet | Fabric | Loopback | Management | Port-Channel | Switch | Tunnel | Vlan | Vxlan] | mac-address ***H.H.H*** | monitor [summary | vrf] | remote vlan | resolve [host | interface | mac-address | ***A.B.C.D***] | summary total | vrf [***word*** | all]]

Parameters

- **agent** ARP Agent information.
  - **ipv4** Details related to IPv4.
  - **ipv6** Details related to IPv6.
- **host *name*** Hostname filter name.
- ***A.B.C.D*** IP address filter.
  - **host** Hostname filter.
  - **interface** Interface selector.
  - **mac-address** MAC address filter.
  - **summary** Displays a summary of ARP entries.
- **interface** Interface selector.
  - **Ethernet** Ethernet interface.
  - **Fabric** Fabric interface.
  - **Loopback** Hardware interface used in looping packets.
  - **Management** Management interface.
  - **Port-Channel** Port-Channel sub interface.
  - **Switch** Switch interface.
  - **Tunnel** Tunnel interface.
  - **Vlan** Logical interface into a VLAN.
  - **Vxlan** VXLAN tunnel interface.
- **mac-address *H.H.H*** MAC address filter Ethernet address.
- **monitor** Monitored IP/IPv6 addresses.
  - **summary** Displays a summary of monitored IP/IPv6 addresses.
  - **vrf** Displays monitored IP/IPv6 addresses in a VRF.
- **remote vlan** Remote host bindings displaying information about the specified VLAN.
- **resolve** Resolves host names.
  - **host** Hostname filter.

    - **interface** Interface selector.
    - **mac-address** MAC address filter.
    - ***A.B.C.D*** IP address filter.
- **summary** Displays a summary of the ARP entries.
    - **total** Displays a count of the total ARP entries.
- **vrf** Displays ARP entries in a VRF.
    - ***word*** The VRF name.
    - **all** All virtual routing and forwarding instances.

Examples

- The **show interfaces vxlan** command displays operational status and configuration information of the specified VXLAN.

```
switch(config)# show interfaces vxlan 1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback0 and is active with 172.16.1.1
  Replication/Flood Mode is headend with Flood List Source: CLI
  Remote MAC learning via Datapath
  VNI mapping to VLANs
  Static VLAN to VNI mapping is
    [100, 100]
  Note: All Dynamic VLANs used by VCS are internal VLANs.
        Use 'show vxlan vni' for details.
  Static VRF to VNI mapping is not configured
  Headend replication flood vtep list is:
   100 172.16.1.2 10.1.1.1
  MLAG Shared Router MAC is 0000.0000.0000
  VTEP address mask is Non
```

- The **show arp** command displays all ARP tables on the configured VXLAN.

```
switch(config)# show arp interface vxlan 1
Address         Age (sec)  Hardware Addr   Interface
192.168.10.1              -  0000.abab.abab  Vlan100, Vxlan1
```

- The **show arp interface summary** command displays a summary of all ARP tables on the configured VXLAN.

```
switch(config)# show arp interface vxlan 1 summary
Total: 1
Static: 1
Dynamic: 0
Not learned: 0
```

17.4.7 show interfaces vxlan

Use the **show interfaces vxlan** command to display the operational status and configuration information of the specified VXLAN.

Command Mode

EXEC

Command Syntax

show interfaces vxlan ***num***

Parameter

num VXLAN tunnel interface number. Range ***1***-***1***.

Example

```
switch(config)# show interfaces vxlan 1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback0 and is active with 172.16.1.1
  Replication/Flood Mode is headend with Flood List Source: CLI
  Remote MAC learning via Datapath
  VNI mapping to VLANs
  Static VLAN to VNI mapping is
    [100, 100]
  Note: All Dynamic VLANs used by VCS are internal VLANs.
        Use 'show vxlan vni' for details.
  Static VRF to VNI mapping is not configured
  Headend replication flood vtep list is:
   100 172.16.1.2 10.1.1.1
  MLAG Shared Router MAC is 0000.0000.0000
  VTEP address mask is Non
```

17.4.8 show service vxlan

The **show service vxlan** command displays the status of the Vxlan Control Service (VCS) and the received (from all connected VTEPs) and advertised (to all connected VTEPs) MAC address reachability information.

Command Mode

EXEC

Command Syntax

show service vxlan [status | switch [**SWITCH_TYPE**] | vni [**VNI_INFO**]]

Parameters

- **SWITCH_TYPE** displayed by switch type. Options include:
  - ***word*** hostname, IP address, or ID of the switch.
  - **all** all switches.
- **VNI_INFO** displayed with VNI information. Options include:
  - **advertised** advertised MAC addresses.
  - **received** received MAC addresses.

Example

This command displays the status of the VCS.

```
switch(config)# show service vxlan status
Vxlan Controller Service is   : stopped
Mac learning                  : Control plane
Resync period                 : 300 seconds
Resync in progress            : No
Capability                    : VXLAN v4 overlay routing
                                VXLAN v4 overlay indirect routing
fm319(config-if-Vx1)#show service vxlan status
Vxlan Controller Service is   : stopped
Mac learning                  : Control plane
Resync period                 : 300 seconds
Resync in progress            : No
Capability                    : VXLAN v4 overlay routing
                                VXLAN v4 overlay indirect routing
```

```
switch(config)#
```

17.4.9 show vxlan address-table

The **show vxlan address-table** command displays the VXLAN address table. Entries are created by extracting information from packets received from remote VTEPs.

The VXLAN address table correlates MAC addresses that are accessible through remote VTEPs with the local VLAN and the IP address of the VTP through which the addressed device is accessible. The VTI uses this table when constructing the VXLAN encapsulation fields to specify the destination IP address of the recipient VTEP and the VNI segment through which the device's remote VLAN is accessed.

Command Mode

EXEC

Command Syntax

show vxlan address-table [**ENTRY_TYPE**] [**MAC_ADDR**] [**VLANS**] [**REMOTE_VTEP**]

Parameters

- **ENTRY_TYPE** command filters display by entry type. Options include:
  - ***no parameter*** all table entries.
  - **configured** static entries; includes unconfigured VLAN entries.
  - **dynamic** entries learned though packet receipts.
  - **static** entries entered by CLI commands.
  - **unicast** entries with unicast MAC address.
- **MAC_ADDR** command uses MAC address to filter displayed entries.
  - ***no parameter*** all MAC addresses table entries.
  - **address *mac_address*** displays entries with specified address (dotted hex notation – H.H.H).
- **VLANS** command filters display by VLAN.
  - ***no parameter*** all VLANs.
  - **vlan *v_num*** VLAN specified by ***v_num***.
- **REMOTE_VTEP** Filters entries by IP address of the remote VTEPs. Options include:
  - ***no parameter*** all items.
  - **vtep *ipaddr_1*** [***ipaddr_2***...***ipaddr_n***] Identifies VTEPs by their IP address.

Example

This command displays the VXLAN address table.

```
switch> show vxlan address-table
          Vxlan Mac Address Table
----------------------------------------------------------------------

Vlan  Mac Address     Type     Prt  Vtep            Moves   Last Move
----  -----------     ----     ---  ----            -----   ---------
  51  0000.0051.0101  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0102  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0103  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0104  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0105  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0102  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0103  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0104  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0105  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago switch>
```

17.4.10 show vxlan counters

The **show vxlan counters** command displays the VXLAN counters.

Command Mode

EXEC

Command Syntax

show vxlan counters ROUTE_TYPE

Parameters

- **ROUTE_TYPE** Specifies the type of VXLAN counter displayed by the command.
  - **software** Command displays software routers.
  - **varp** Command displays virtual-ARP counters.
- **vtep** Command displays counters for VTEPs which are identified by their IP address. An optional keyword allows the user to view a single direction of the counters:
  - **encap** "encap" counters count packets coming from the edge, encapsulated on the device and directed to the core.
  - **decap** "decap" counters count packets coming from the core, decapsulated on the device and heading towards the edge.

Related Command

clear vxlan counters resets the VXLAN counters.

Examples

- This command displays the VXLAN counters for software routers.

```
switch> show vxlan counters software
encap_bytes:3452284
encap_pkts:27841
encap_read_err:1
encap_discard_runt:0
encap_discard_vlan_range:0
encap_discard_vlan_map:0
encap_send_err:0
encap_timeout:1427
decap_bytes_total:382412426
decap_pkts_total:2259858
decap_bytes:0
decap_pkts:0
decap_runt:0
decap_pkt_filter:45128
decap_bytes_filter:5908326
decap_discard_vxhdr:0
decap_discard_vlan_map:2214730
decap_timeout:0
decap_sock_err:1 switch>
```

- This command displays the VXLAN counters for VTEPs.

```
switch> show vxlan counters vtep
                                                                  Decap Drop or
                                  Decap Known           Decap BUM      Exception
VTEP         Decap Bytes      Unicast Packets           Packets          Packets
-------- ---------------- -------------------- -------------------- -------------
1.0.14.1     62526968000            312632701            312636979             2
1.0.16.1             800                    2                    6     312279633
1.0.23.1             800                    2                    6             2
unlearnt               0                    0                    0             0
                                        Encap Drop or
                                            Exception
```

```
VTEP         Encap Bytes        Encap Packets       Packets
-------- --------------- -------------------- -------------
1.0.14.1     30579308814            268239551             2
1.0.16.1            1140                   10             2
1.0.23.1               0                    0             0 switch>
```

17.4.11 show vxlan flood vtep

The **show vxlan flood vtep** command displays the flood list that the switch is using to perform head-end replication. Head-end replication is a data distribution method that supports Broadcast, Unknown unicast, and Multicast (BUM) traffic over VXLANs by replicating BUM data locally for transmission to the set of remote VTEPs that a flood list specifies. The command displays the VLAN ID that references the configured VNIs (**vxlan vlan vni**).

The flood list is determined by the **vxlan flood vtep** command.

Command Mode

EXEC

Command Syntax

show vxlan flood vtep [**VLANS**]

Parameters

VLANS command filters display by the reference VLAN.

- ***no parameter*** all VLANs.
- **vlan *v_range*** VLANs specified by ***v_range***.

  Valid ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Guidelines

The command displays flood list contents only when the VLAN line protocol status is **up**.

Related Command

vxlan flood vtep configures the flood list.

Example

These commands display the VTEPs that have exchanged data with the configured VTI.

```
switch> show vxlan flood vtep vlan 100-102

          Vxlan Flood Vtep Table
------------------------------------------------------

Vlan   Ip Address
----   ------------------------------------------------
100    3.3.3.3
101    11.1.1.1         11.1.1.2         11.1.1.3
102    11.1.1.1         11.1.1.2         11.1.1.3
       12.1.1.1 switch>
```

17.4.12 show vxlan vtep

The **show vxlan vtep** command displays information about remote VTEPs that the configured VTI has discovered and with whom it has exchanged packets.

Command Mode

EXEC

Command Syntax

show vxlan vtep

Example

These commands display the VTEPs that have exchanged data with the configured VTI.

```
switch> show vxlan vtep
Remote vteps for Vxlan1:
10.52.2.12
Total number of remote vteps:  1 switch>
```

17.4.13 vxlan flood vtep

The **vxlan flood vtep** command supports VXLAN head-end replication by creating or modifying a list that specifies remote VTEPs to which the switch bridges replicated traffic. Head-end replication is a data distribution method that supports Broadcast, Unknown unicast, and Multicast (BUM) traffic over VXLANs by replicating BUM data locally for transmission to the set of remote VTEPs that a flood list specifies. This data flooding facilitates remote MAC address learning through the forwarding of data with unknown MACs.

Each **vxlan flood vtep** statement in ***running-config*** associates a set of VTEP addresses to an access VNI. A default flood list is also configurable that applies to all VNIs for which a flood list is not configured. The **vxlan flood vtep** command is available in the following formats to create or modify corresponding ***running-config*** statements:

- **vxlan flood vtep** creates a statement for a specified VNI and replaces existing statements for that VNI.
- **vxlan flood vtep add** modifies an existing flood statement by adding the specified VTEPs. This statement creates a list if it references a VNI that has no flood statement.
- **vxlan flood vtep remove** modifies an existing flood statement by deleting the specified VTEPs. This statement has no effect if it references a VNI that has no flood statement.

The **vxlan flood vtep** command specifies a VNI by referencing its associated VLAN ID (**vxlan vlan vni**). The command provides these options for specifying the reference VLANs:

- **a single VLAN**: creates or modifies a single statement referenced by the command.
- **a range of VLANs**: creates or modifies all statements referenced by the VLAN range.
- **no VLAN**: creates or modifies the default list.

The **no vxlan flood vtep** and **default vxlan flood vtep** commands remove the specified flood list by deleting the corresponding **vxlan flood vtep** statements from ***running-config***. Commands that specify a VLAN range remove all corresponding statements.

Command Mode

Interface-VXLAN Configuration

Command Syntax

vxlan [**ACCESS_VNI**] **flood vtep**] **MODIFY**] **VTEP_1** [**VTEP_2**]...[**VTEP_N**]

no vxlan [**ACCESS_VNI**] **flood vtep**

default vxlan [**ACCESS_VNI**] **flood vtep**

Parameters

- **ACCESS_VNI** VLAN ID associated to the flood list's target VNI. Value ranges from ***1*** to ***4094***.
  - ***no parameter*** default list.
  - **vlan *vlan_range*** List of VLANs. (Number, range, comma-delimited list of numbers and ranges). Numbers range from ***1*** to ***4094***.
- **MODIFY** Statement modification method. Options include:
  - ***no parameter*** creates new list for specified VLANs. Current list is overwritten.

- **add** specified VTEPs are added to existing list.
- **remove** specified VTEPs are deleted from existing list.

- **VTEP_X** IPv4 address of VTEPs that are added or removed from the list.

Examples

- These commands create a default VXLAN head-end replication flood list.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan flood vtep 10.1.1.1 10.1.1.2
switch(config-if-Vx1)# show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- These commands create VXLAN head-end replication flood lists for the VNIs accessed through ***vlan101*** and ***vlan 102***.

```
switch(config-if-Vx1)# vxlan vlan 101-102 flood vtep 11.1.1.1 11.1.1.2
 11.1.1.3
switch(config-if-Vx1)# show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan vlan 101 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan vlan 102 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- These commands add two VTEPs for the VNI access through ***vlan 102***.

```
switch(config-if-Vx1)# vxlan vlan 102 flood vtep add 12.1.1.1
switch(config-if-Vx1)# show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan vlan 101 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan vlan 102 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3 12.1.1.1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

17.4.14 vxlan multicast-group decap

The **vxlan multicast-group decap** command enables VXLAN multicast decapsulation.

VTEPs are enabled by VXLAN multicast decapsulation, supporting Head End Replication (HER). Multicast encapsulated Broadcast/Unknown/Multicast (BUM) packets terminate VTEPs from remote VTEPs that do not support HER.

The **no vxlan multicast-group decap** and **default vxlan multicast-group decap** commands disable VXLAN multicast decapsulation.

Command Mode

Interface-VXLAN Configuration

Command Syntax

vxlan multicast-group decap ***group_addr***

no vxlan multicast-group decap

default vxlan multicast-group decap

Parameter

group_addr IPv4 address of multicast group. Dotted decimal notation of a valid multicast address.

Examples

- This command enables VXLAN multicast decapsulation.

```
switch(config)# interface vxlan 1
switch(config-config-if-Vx1)# vxlan multicast-group decap 230.1.1.1
switch(config-config-if-Vx1)#
```

- This command disables VXLAN multicast decapsulation.

```
switch(config)# interface vxlan 1
switch(config-config-if-Vx1)# no vxlan multicast-group decap 230.1.1.1
switch(config-config-if-Vx1)#
```

17.4.15 vxlan source-interface

The **`vxlan source-interface`** command specifies the interface from which the configuration mode VXLAN Interface (VTI) derives the source address (IP) that it uses when exchanging VXLAN frames. There is no default source interface assignment.

The **`no vxlan source-interface`** and **`default vxlan source-interface`** commands remove the source interface assignment from the ***interface-vxlan*** configuration mode by deleting the corresponding **`ip vxlan source-interface`** command from ***running-config***.

Command Mode

Interface-VXLAN Configuration

Command Syntax

`vxlan source-interface` **INT_NAME**

`no vxlan source-interface`

`default vxlan source-interface`

Parameters

INT_NAME Interface type and number. Options include:

- **loopback *l_num*** Loopback interface specified by ***l_num***.

Guidelines

A VXLAN interface is inoperable without the source-interface assignment.

Related Command

interface vxlan places the switch in VXLAN interface configuration mode.

Example

These commands configure VTI 1 to use the IP address ***10.25.25.3*** as the source address of outbound VXLAN frames.

```
switch(config)# interface loopback 15
switch(config-if-Lo15)# ip address 10.25.25.3/24
switch(config-if-Lo15)# exit
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan source-interface loopback 15
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan source-interface Loopback15
   vxlan udp-port 4789
```

```
switch(config-if-Vx1)#
```

17.4.16 vxlan udp-port

The **vxlan udp-port** command associates a UDP port with the configuration mode VXLAN Interface (VTI). By default, UDP ***port 4789*** is associated with the VTI.

Packets bridged to the VTI from a VLAN are encapsulated with a VXLAN header that includes the VNI associated with the VLAN and the IP address of the VTEP that connects to the recipient, then sent through the UDP port. Packets that arrive through the UDP port are sent to the bridging domain of the recipient VLAN as determined by the VNI number in the VXLAN header and the interface's VNI-VLAN map.

The **no vxlan udp-port** and **default vxlan udp-port** command restores the default UDP port association (***4789***) on the configuration mode interface by deleting the corresponding **vxlan udp-port** command from ***running-config***.

Command Mode

Interface-VXLAN Configuration

Command Syntax

vxlan udp-port ***port_id***

no vxlan udp-port

default vxlan udp-port

Parameters

port_id UDP port number. Value ranges from ***1024*** to ***65535***.

Guidelines

UDP ***port 4789*** is reserved by convention for VXLAN usage. Under most typical applications, this parameter should be set to the default value.

Related Commands

interface vxlan places the switch in ***interface-vxlan*** configuration mode.

Examples

- This command associates UDP ***port 5500*** with ***interface vxlan 1***.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan udp-port 5500
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 5500
switch(config-if-Vx1)#
```

- This command resets the ***interface vxlan 1*** UDP port association of ***4789***.

```
switch(config-if-Vx1)# no vxlan udp-port
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

17.4.17 vxlan vlan vni

The **vxlan vlan vni** command associates a VLAN ID with a virtual network identifier (VNI). A VNI is a 24-bit number that is assigned to a VLAN to distinguish it from other VLANs that are on a VXLAN Tunnel

Interface (VTI). VNI values range from ***1*** to ***16777215*** in decimal notation and from ***0.0.1*** to ***255.255.255*** in dotted decimal notation.

When a VLAN bridges a packet to the VTI, the packet is encapsulated with a VXLAN header that includes the VNI that is associated with the VLAN. Packets that arrive on the VTI's UDP socket are bridged to the VLAN that is associated with the VNI specified by the VXLAN header that encapsulates the packet.

The VTI requires a one-to-one correspondence between specified VLANs and VNI values. Commands that assign a new VNI to a previously configured VLAN replace the existing VLAN assignment statement in ***running-config***. Commands that attempt to assign a VNI value to a second VLAN generate a CLI error.

The **no vxlan vlan vni** and **default vxlan vlan vni** commands remove the specified VLAN-VNI association from the configuration mode interface by deleting the corresponding **vxlan vlan** command from ***running-config***.

Command Mode

Interface-VXLAN Configuration

Command Syntax

vxlan vlan ***vlan_id*** **vni** [***vni_id***]

no vxlan vlan ***vlan_id*** **vni** [***vni_id***]

default vxlan vlan ***vlan_id*** **vni** [***vni_id***]

Parameters

- ***vlan_id*** number of access VLAN. Value ranges from ***1*** to ***4094***.
- ***vni_id*** VNI number. Valid formats: decimal ***1*** to ***16777215*** or dotted decimal ***0.0.1*** to ***255.255.255***.

Example

These commands associate ***vlan 100*** to ***vni 100*** and ***vlan 200*** to ***vni 10.10.200***.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan vlan 100 vni 100
switch(config-if-Vx1)# vxlan vlan 200 vni 10.10.200
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 200 vni 658120
   vxlan vlan 100 vni 100
switch(config-if-Vx1)# vxlan vni notation dotted
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 200 vni 10.10.200
   vxlan vlan 100 vni 0.0.100
switch(config-if-Vx1)#
```

17.4.18 vxlan vni notation dotted

The **vxlan vni notation dotted** command configures the switch to display VNIs in dotted decimal notation. A Virtual Network Identifier (VNI) is a 24-bit number that is assigned to a VLAN to distinguish it from other VLANs that are on a VXLAN tunnel interface. VNI values range from ***1*** to ***16777215*** in decimal notation and from ***0.0.1*** to ***255.255.255*** in dotted decimal notation.

The command affects the VNI number display in all **show** commands, including **show running-config**. Commands that include VNI as a parameter may use decimal or dotted decimal notion regardless of the setting of this command. By default, show commands display VNI number in decimal notation.

The **no vxlan vni notation dotted** and **default vxlan vni notation dotted** commands restore the default setting of displaying vni numbers in decimal notation by deleting the **vxlan vni notation dotted** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

vxlan vni notation dotted

no vxlan vni notation dotted

default vxlan vni notation dotted

Examples

- These commands configure the switch to display VNI numbers in dotted decimal notation, then displays a configuration that includes a VNI setting.

```
switch(config)# vxlan vni notation dotted
switch(config)# interface vxlan 1
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 333 vni 3.4.5
switch(config-if-Vx1)#
```

- These commands configure the switch to display VNI numbers in decimal notation, then displays a configuration that includes a VNI setting.

```
switch(config)# no vxlan vni notation dotted
switch(config)# interface vxlan 1
switch(config-if-Vx1)# show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 333 vni 197637
switch(config-if-Vx1)#
```

Chapter 18

Ethernet VPN (EVPN)

This chapter describes Arista's EVPN implementation. Sections in this chapter include:

- EVPN Overview
- EVPN Layer 3 Core Operations
- Integrated Routing and Bridging
- VPN MPLS Transport Options
- EVPN Type-5 Routes: IP Prefix Advertisement
- BGP PIC Edge for EVPN VXLAN Routes for Remote VTEP Failures
- VXLAN DSCP Mapping
- EVPN IGP Cost for VTEP Reachability
- EVPN VXLAN Single-Gateway Centralized Routing
- Inter-VRF Local Route Leaking
- Static Inter-VRF Route
- VCS to EVPN Hitless Migration
- Configuring EVPN
- Sharing Equivalence Class entry across multiple VRF
- Sample Configurations
- EVPN and VCS Commands

18.1 EVPN Overview

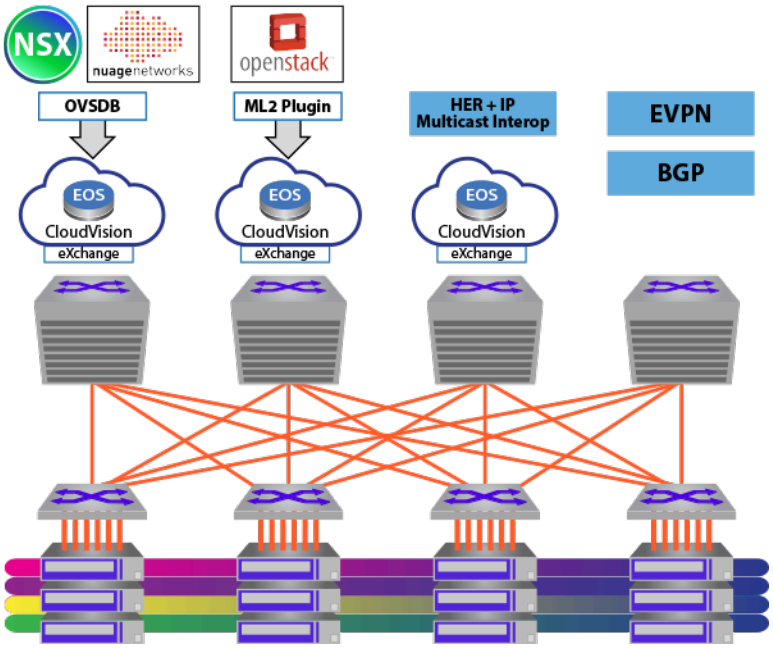
Ethernet VPN (EVPN) is a standards-based BGP control plane to advertise MAC addresses, MAC and IP bindings and IP Prefixes. This document focuses on EVPN and its operation with a VXLAN data plane for building overlay networks in the data center.

A number of control planes exist today for VXLAN, based on specific use cases, whether it be a requirement to integrate with an SDN overlay controller, or operate in a standards based flood and learn control plane model.

Current flood and learn models operate either with a multicast control plane, or ingress replication, where the operator manually configures the remote VTEPs in the flood list. Both of these are data-plane driven, that is, MAC's are learned via flooding. In the IP multicast model MAC's are learned in the underlay via flooding to an IP multicast group, while ingress replication (HER) floods to configured VTEP endpoints and no IP Multicast is required in the underlay.

The controller based solution with Cloud Vision eXchange (CVX), locally learned MAC's are published to a centralized controller and these MAC's are then programed to all participating VTEPs.

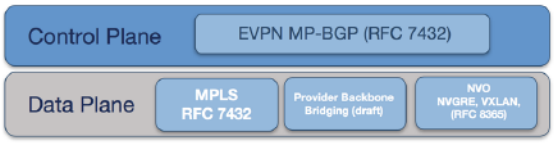
Figure 18-1: Different VXLAN Control Planes



A controller-less BGP EVPN MAC learning is a standards-based control-plane (MP-BGP) is used to discover remote VTEPs and advertise MAC address and MAC/IP bindings in the VXLAN overlay, thus eliminating the flood and learn paradigms of the previously mentioned (multicast or HER) controller-less approaches. As a standards-based approach, the discovery and therefore the advertisement of the EVPN service models can inter-operate amongst multiple vendors.

This highlights an important and powerful advantage of BGP EVPN; that being, it is a single control plane for multiple data-plane encapsulations and defines both Layer 2 and Layer 3 VPN services. As network operators drive toward simplicity and automation, having one control plane protocol and address family for all data-planes and VPN services will prove extremely powerful.

Figure 18-2: VXLAN Control Plane and Data-plane Definitions



The initial EVPN standard is ***RFC 7432*** defined the BGP EVPN control plane and specifies an MPLS data-plane. The control plane with an MPLS data plane was extended to consider additional data plane encapsulations models including VXLAN, NVGRE, and MPLS over GRE.

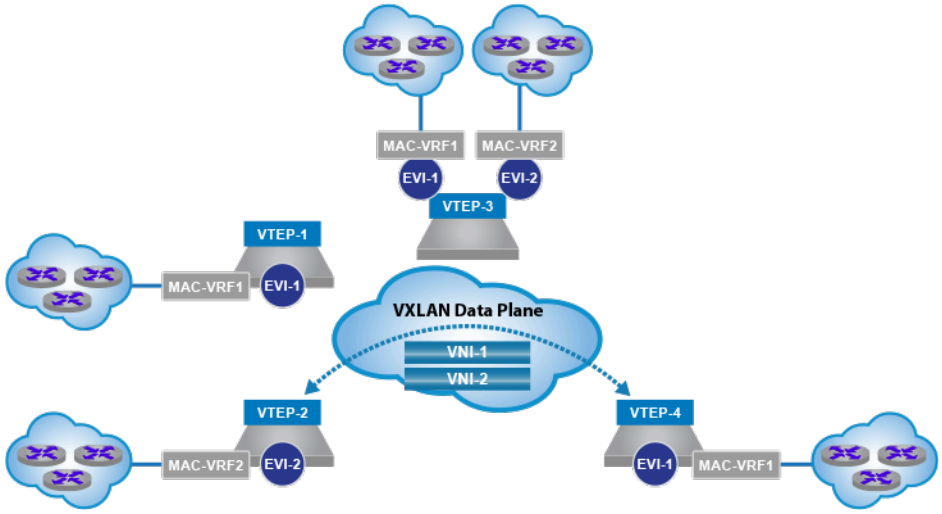
18.1.1 EVPN Terminology

The EVPN standard in the context of an NVO environment, defines the functionality for delivering multi-tenant Layer 2/3 VPN services using either VXLAN, NVGRE or MPLS over GRE encapsulation, across a common physical IP infrastructure. The standard introduces new terminology specific to a NVO environment, which are summarized below in relation to VXLAN encapsulation.

Network Virtualization Overlay (NVO): The overlay network used to deliver the Layer 2 and Layer 3 VPN services. For VXLAN encapsulation, this would define a VXLAN domain, which would include one or more VNIs, for the transportation of tenant traffic over a common IP underlay infrastructure.

- **Network Virtualization End-Point (NVE):** The provider edge node within the NVO environment responsible for the encapsulation of tenant traffic into the overlay network. For a VXLAN data plane, this defines the Virtual Tunnel End-Point (VTEP).
- **Virtual Network Identifier (VNI):** The label identifier within the VXLAN encapsulated frame, defining a Layer 2 domain in the overlay network.
- **EVPN instance (EVI):** A logical switch within the EVPN domain which spans and interconnects multiple VTEPs to provide tenant Layer 2 and Layer 3 connectivity.
- **MAC-VRF:** A Virtual Routing and Forwarding table for storing Media Access Control (MAC) addresses on a VTEP for a specific tenant.

Figure 18-3: EVPN Terminology for a VXLAN Data Plane



The new EVPN Network Layer Reachability Information (NLRI) is carried in BGP using Multi-protocol BGP Extensions with a newly defined Address Family Identifier (AFI) and Subsequent Address Family Identifier (SAFI).

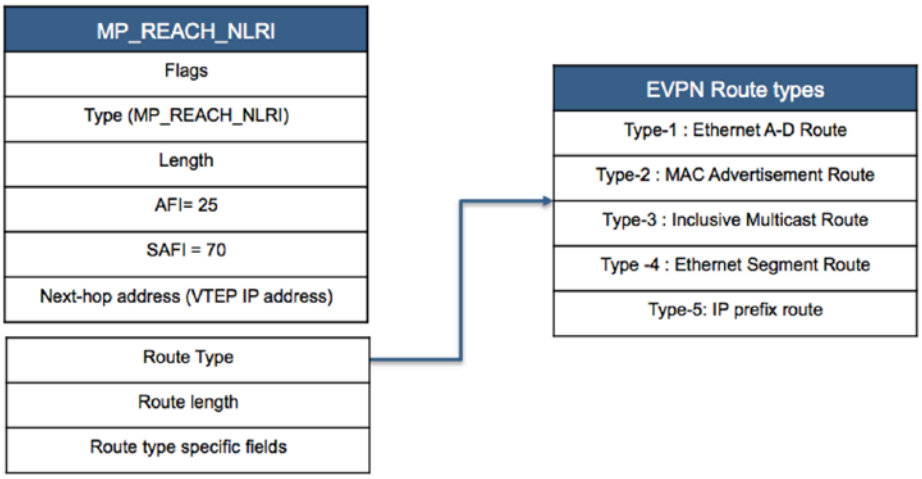
To provide multi-tenancy, the standard uses the above traditional VPN methods to control the import and export of routes and provide support for overlapping IP address between tenants.

Multi-protocol BGP for EVPN: A new AFI and SAFI have been defined for EVPN. These are AFI=25 (Layer 2 VPN) and SAFI = 70 (EVPN).

- **EVPN Layer 2/Layer 3 tenant segmentation:** Similar to standard MPLS VPN configurations Route Distinguisher's (RD's) and Route Targets (RT's) are defined for the VPN.
- **Route Target (RT):** To control the import and export of routes across VRFs, EVPN routes are advertised with Route-Target (RT) (BGP extended communities). The RT can be auto derived to simplify the rule configuration, typically this is based on the AS number and the VNI of the MAC-VRF.
- **Route Distinguisher (RD):** Unique number prepended to the advertised address within the VRF, ensuring support for overlapping IPs and MACs across different tenants.

The format of the MP_REACH_NLRI/MP_UNREACH_NLRI attribute, holding the new EVPN NLRI is illustrated below, where the next-hop address within the NLRI is the IP address of the VTEP advertising the EVPN route.

Figure 18-4: EVPN NLRI Route Format



As illustrated, the original ***MPLS RFC (7348)*** and subsequent IP prefix draft (**EVPN Terminology**), introduce five unique EVPN route types.

Type-1 Route: Ethernet A-D route

Ethernet A-D route per ESI route, announces the reachability of a multi-homed Ethernet Segment. The route type is used for fast convergence (ie: 'mass withdraw') functions, as well as split horizon filtering used for active-active multi-homing.

Ethernet A-D route per EVI route, is used to implement the Aliasing and Backup Path features of EVPN associated with active-active multi-homing.

Type-2 Route: Host advertisement Route

Used to advertise the reachability of a MAC address, or optionally a MAC and IP binding as learned by a specific EVI. With the advertisement of the optional IP address of the host, EVPN provides the ability for VTEPs to perform ARP suppression and ARP proxy to reduce flooding within the Layer 2 VPN.

Type-3 Route: Inclusive Multicast route

The type-3 route is used to advertise the membership of a specific Layer 2 domain (VNI within the VXLAN domain), allowing the dynamic discovery of remote VTEPs in a specific VNI and the population of a VTEP ingress flood list for the forwarding of Broadcast Unknown unicast and Multicast (BUM) traffic.

Type-4 Route: Ethernet Segment Route

The type-4 route is specific to VTEPs supporting the EVPN multi-homing model, for active-active and active-standby forwarding. The route is used to discover VTEPs which are attached to the same shared Ethernet Segment. Additionally, this route type is used in the Designated Forwarder (DF) election process.

Type-5 Route: IP-prefix route advertisement

The type-5 route is used to advertise IP prefixes rather the MAC and IP hosts addresses of the type-2 route. This advertisement of prefixes into the EVPN domain provides the ability to build classic Layer 3 VPN topologies.

A detailed understanding of the function of each of these route types in the operation of EVPN to provide multi-tenant Layer 2 and 3 VPN services, is defined in Section 4 of this document.

While this guide focuses on EVPN with VXLAN data-plane encapsulation, it's important to note that, in addition to the new routes type, a BGP encapsulated extended community is included in all advertisements to determine the data-plane encapsulation.

The Encapsulation extended community is defined in ***RFC 5512***. The different IANA registered tunnel types for an NVO environment are summarized in the table below.

Figure 18-5: Defined Data-Plane Encapsulations

Extended Community Value	Name	Reference
8	VXLAN	draft-ietf-bess-evpn-overlay-08
9	NVGRE	draft-ietf-bess-evpn-overlay-08
10	MPLS	RFC 7342
11	MPLSoGRE	draft-ietf-bess-evpn-overlay-08

18.1.2 EVPN Service Models

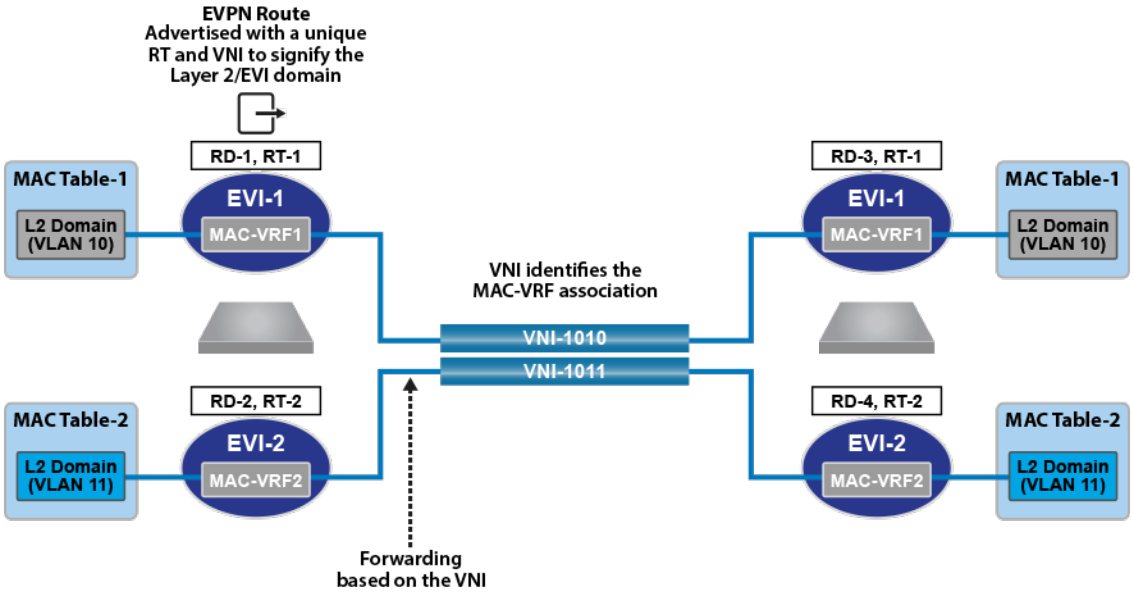
An EVPN Instance (EVI), can contain, one or more Layer 2 broadcast domains (VLANs).

The association of a VLAN-IDs to a specific EVI instance and how a VLAN tag can be transported within the EVI if required, is defined by three EVPN service models: VLAN based, VLAN Bundle, and VLAN aware bundle.

18.1.2.1 VLAN Based Service Interface

In the VLAN based service there is a one-to-one mapping between the VLAN-ID and the MAC-VRF of the EVPN instance. With the MAC-VRF mapping directly to the associated VLAN, there will be a single bridge table within the MAC-VRF. The VLAN tag is not carried in any route update and the VNI label in the route advertisement is used to uniquely identify the bridge domain of the MAC-VRF in the VXLAN forwarding plane.

Figure 18-6: VLAN Based Service Interface



With a one-to-one mapping between the VLAN-ID and the MAC-VRF of EVI instance, the EVI will represent an individual tenant subnet/VLAN in the overlay. The one-to-one mapping also means the route-target associated with the MAC-VRF, uniquely identifies the tenant's subnet/VLAN, providing granular importing of MAC routes on a per VLAN basis on each VTEP.

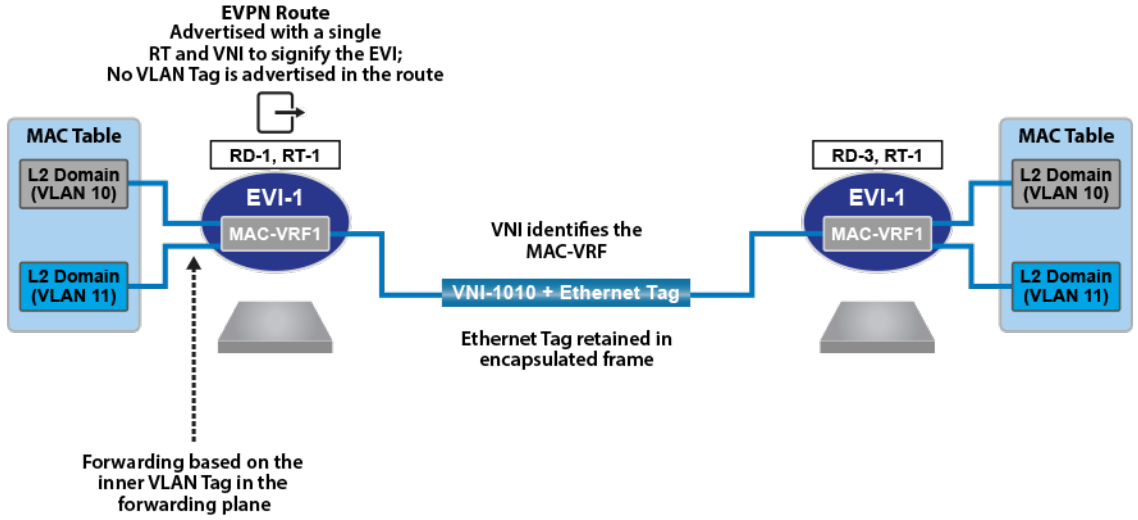
In this service, the associated MAC-VRF table is identified by the Route-Target in the control plane and by the VNI in the data plane and the MAC-VRF table corresponds to a single VLAN bridge domain.

18.1.2.2 VLAN Bundle Service Interface

In the VLAN bundle service, there is a many-to-one mapping between the VLAN-IDs and the MAC-VRF of the EVPN instance. The MAC-VRF however only contains a single Layer 2 bridge table and VNI label, thus MAC addresses must be unique across all associated VLANs.

With the MAC-VRF containing a single Layer 2 bridge table and a single VNI, the original VLAN tag has no significance in the control plane and is not carried in any EVPN route update. The original Ethernet tag and the VNI label are carried in the VXLAN data plane, to allow forwarding to the correct tenant VLAN.

Figure 18-7: VLAN Bundle Service Interface



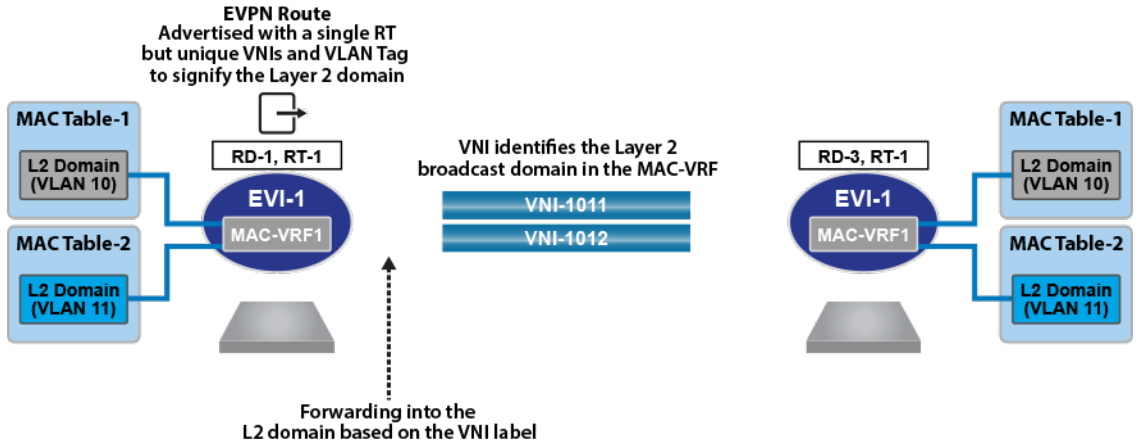
In this service, the Route-Target associated with the MAC-VRF identifies the tenant rather than an individual subnet/VLAN of a tenant. This means all MAC routes for the tenant will be imported on the VTEP regardless of whether or not the specific tenant VLAN exists. The MAC-VRF table is identified by the Route-Target in the control plane and forwarding to the appropriate tenant VLAN is achieved via a combination of the VNI and Ethernet tag in the VXLAN data plane.

18.1.2.3 VLAN Aware Bundle Service Interface

In the VLAN aware bundle service, there is a many-to-one mapping between the VLAN-IDs and the MAC-VRF of the EVPN instance. However, the MAC-VRF contains a unique Layer 2 bridge table for each associated VLAN-ID and a unique VNI label for each bridge domain.

With the MAC-VRF containing multiple Layer 2 bridge tables, the VLAN tag is carried in any EVPN route update to allow mapping to the correct tenant bridge table within the MAC-VRF. Only the unique VNI label is carried in the VXLAN data plane, to allow forwarding to the correct VLAN with the MAC-VRF.

Figure 18-8: VLAN Aware Bundle Service



In this service, the MAC-VRF of the EVI instance represents multiple subnet/VLANs of the tenant. The Layer 2 bridge table of the MAC-VRF is identified by a combination of the Route-Target and the Ethernet tag in the control plane and by the unique VNI and in the VXLAN data plane.

This service type is a common DCI/WAN deployment, where a tenant's VLANs are bundled into single EVI instance, while VLAN "awareness" can be retained in the EVPN service as the VNI tag is advertised in the MAC-IP route (which now identifies the VLAN within the EVI). Bundling into a service like this reduces the number of EVI's that need to be configured, reducing complexity and the control-plane signaling between PE's.

18.1.3 VCS and EVPN in DCI

When VXLAN Control Services (VCS) is enabled on a CloudVision eXchange (CVX) of a Data Center (DC), each VXLAN Tunnel End Point (VTEP) connects to the corresponding CVX for sharing the Layer 2 bridging information of it's attached hosts. In turn, CVX advertises this information to all VTEPs within the DC.

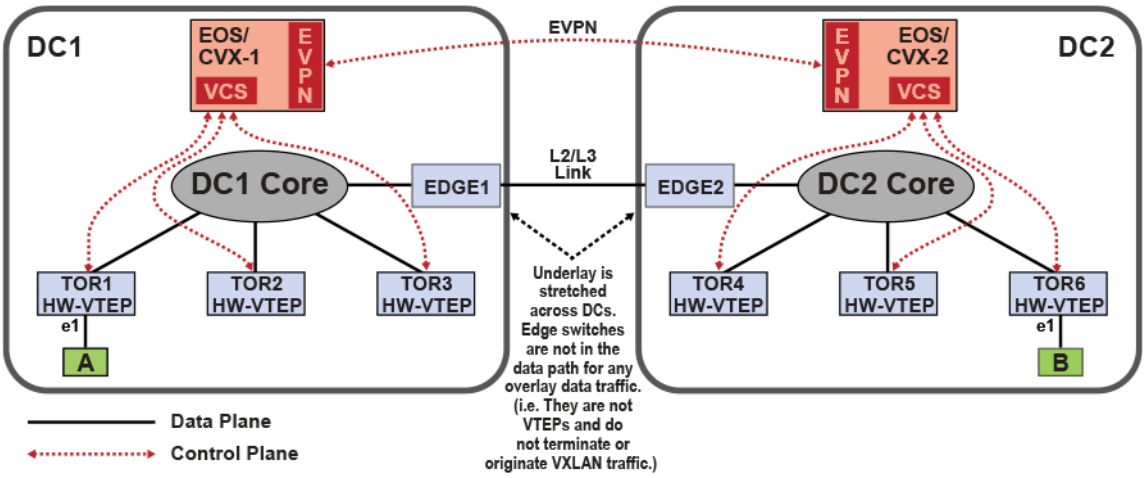
In a topology consisting of multiple DCs where each DC runs its own CVX instance as shown below, a federation of CVXs can be created by using BGP-EVPN. In such Data Center Interconnect (DCI) topologies, CVX in each DC performs the following functions to advertise the Layer 2 bridging information (MAC-VTEP bindings) to all VTEPs in different DCs:

- Receives the local Layer 2 bridging information in CVX control plane format from all VTEPs within the DC; and advertises it to remote CVXs in the BGP-EVPN NLRI format.
- Receives the Layer 2 bridging information in BGP-EVPN NLRI format from remote CVXs; and advertises it to local VTEPs in the CVX control plane format.

Note: The distribution of Layer 2 bridging information as described above allows a Layer 2 overlay network to be stretched across multiple DCs without additional VTEP configurations.

The following graphic illustrates the federation of CVX across multiple DCs.

Figure 18-9: CVX Connected from Multiple DCs



18.1.4 EVPN MPLS LAYER 3 VPN (Type-5 Route)

Ethernet VPN (EVPN) is an extension of the BGP protocol introducing a new address family: Layer 2 VPN (address family number 25) / EVPN (subsequent address family number 70). It is used to exchange overlay MAC and IP address reachability information between BGP peers using type-2 routes. Additionally, EVPN supports the exchange of layer 3 IP overlay routes through the extensions described in (type 5 EVPN routes).

An IP VRF is used on a PE router for each Layer 3 overlay. VRF IP routes are exported into the EVPN BGP table and advertised to remote VTEPs as type 5 routes. The exported EVPN routes carry the Route-Target (RT) extended communities that are configured as export route-targets on the IP VRF from which they were exported.

The RTs carried by the EVPN type 5 routes received by a PE are matched against the VRF import route-target configuration. When a received route carries an RT that is configured as an import route-target on an IP VRF, the route is imported into the IP table for that VRF.

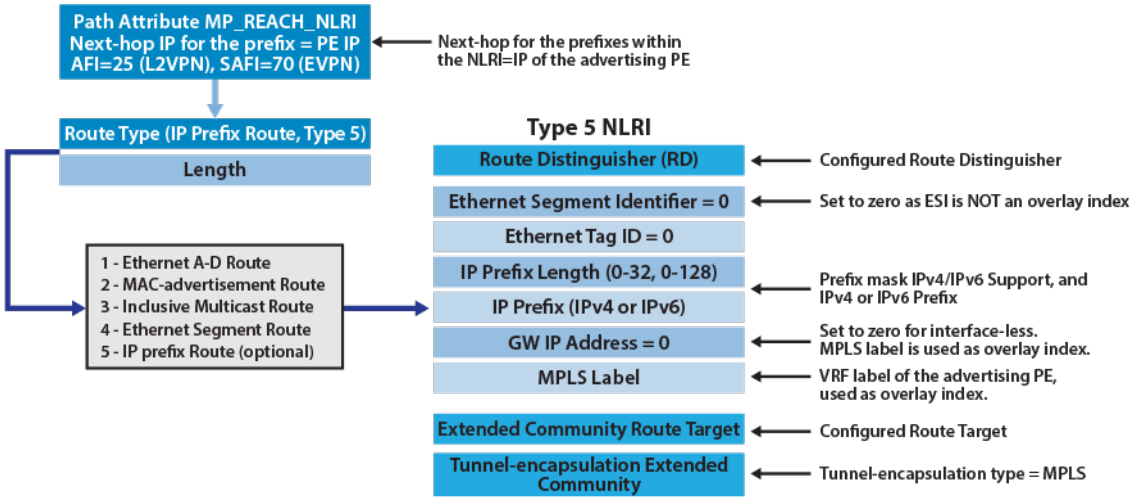
PE routers allocate per-VRF and address family Labels that are advertised as part of the Layer 3 (type 5) EVPN route NLRI. Forwarding of overlay packets between PEs across the underlay requires underlay MPLS connectivity provided by an IP backbone.

The type-5 routes provide the ability to decouple the advertisement of an IP prefix from any specific MAC address, providing the ability to support floating IP address, optimized the mechanism for advertising external IP prefixes, and reduce the churn when withdrawing IP prefixes.

The format of the new type-5 IP-prefix route is illustrated in the figure below. Unlike when VXLAN is used as a transport, BGP route update for MPLS does not specify the router-mac extended community and sets the

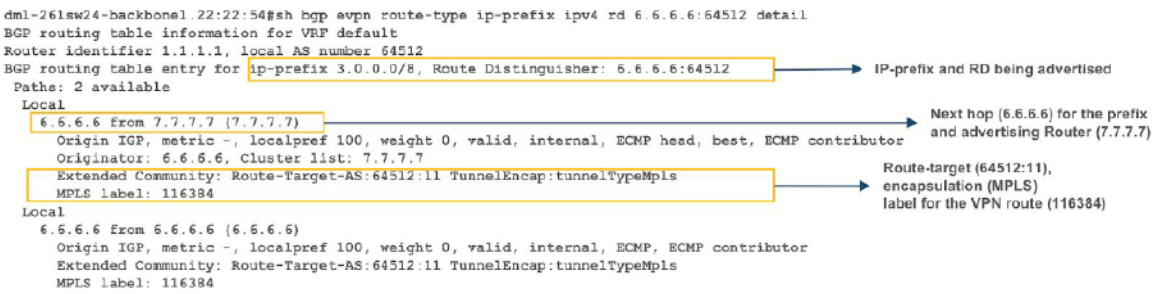
tunnel encapsulation to MPLS. Unlike with VXLAN encapsulation, which uses the VNI as the overlay index, the MPLS Type-5 route uses the MPLS label.

Figure 18-10: EVPN Route Type-5, for Advertisement of IP-Prefixes over MPLS



The following example offers a more detailed view of the route as displayed on a PE router.

Figure 18-11: EVPN Route Type-5 as Shown on PE



As shown in the example, the route contains the VPN route (prefix and RD), the next-hop for the route and the advertising router ID, along with the extended communities of tunnel type (MPLS), MPLS Label value and route-target.

Note: You require ***Release EOS 4.21.1F*** and later versions with Jericho/Jericho+ platforms.

18.1.5 EVPN VxLAN IPV6 Overlay

Beginning with ***EOS Release 4.22.0F***, the EVPN VXLAN L3 Gateway using EVPN IRB supports routing traffic from one IPV6 host to another IPV6 host on a stretched VXLAN VLAN.

18.2 EVPN Layer 3 Core Operations

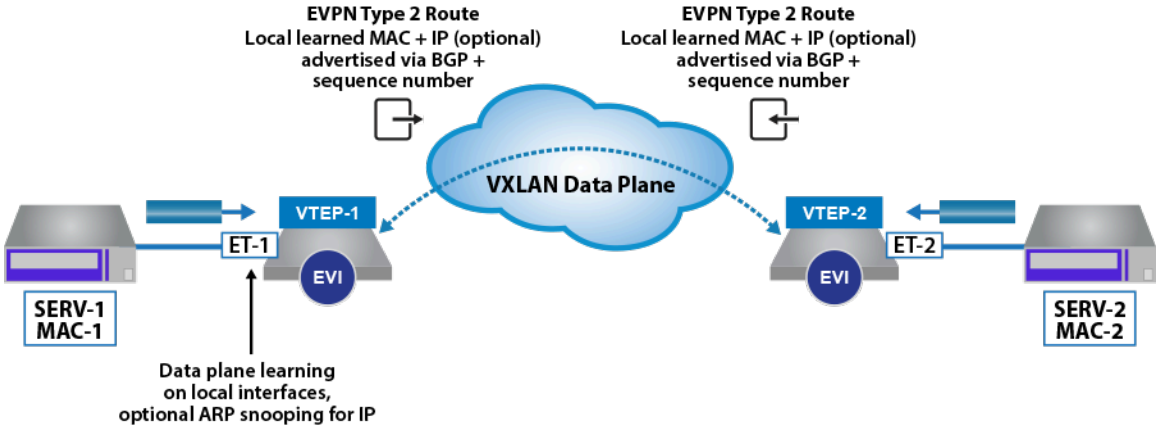
The EVPN standard defines a number of operations and functionality to allow the dynamic learning of MAC and IP bindings, management of MAC moves (VM/host mobility), ARP suppression, automated discovery of remote VTEPs and multi-homing to support active-active topologies.

18.2.1 MAC Address Learning

MAC address learning on the local interface of a VTEP is flow-based learning, however once the MAC addresses are learned locally they are advertised to BGP peers within the EVI via an EVPN route update. The

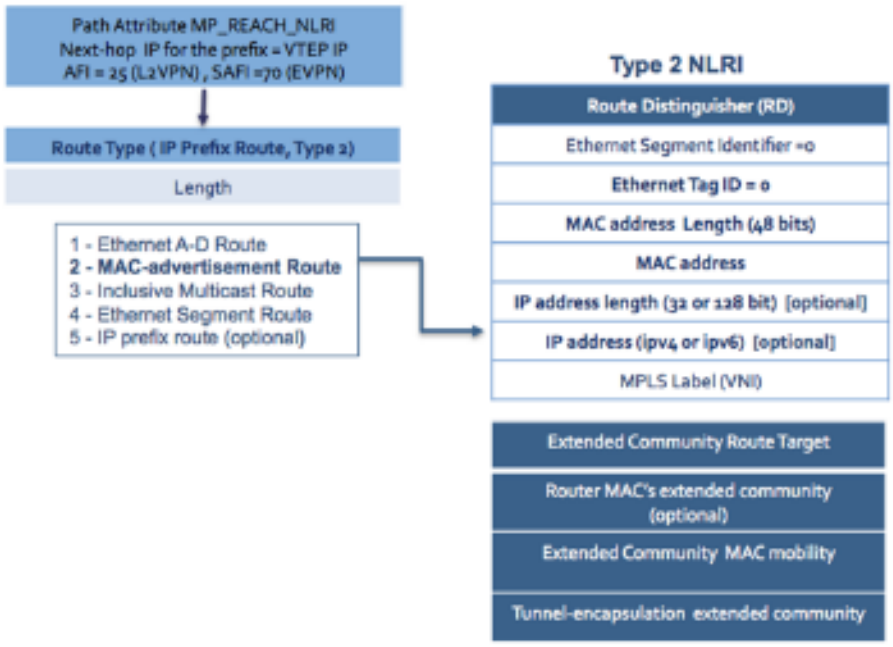
next hop of the update is set to IP of the advertising VTEP. In the case of EVPN VXLAN the label advertised in the update is the VNI, which identifies the MAC-VRF in the case of a VLAN Based service, or the EVI for a VLAN aware bundle service.

Figure 18-12: EVPN Type 2 Route Announcement



The route advertisements are EVPN type-2 routes, which can advertise just the MAC address of the host, or optionally the MAC and IP address of the host. The format of the type-2 route is illustrated in the figure below, along with the mandatory and optional extended community attached to the route.

Figure 18-13: EVPN Type 2 MAC and IP Route Format



Notable fields:

- Multi-protocol Reachable NLRI (MP_REACH_NLRI) attribute of the route is used to carry the next-hop hop for the advertised route. In the context of a VXLAN forwarding plane, this will be the source address (VTI) of the advertising VTEP.
- Route Distinguisher of the advertising node's MAC-VRF.

- Ethernet Segment Identifier (ESI), this field is populated when the VTEP participating in a multi-homed topology. This is discussed in the following sections.
- Ethernet tag ID that will be 0 for VLAN-based service, and the customer VLAN ID in a VLAN-aware bundle service.
- IP address of the host which is associated with advertised MAC address. The advertisement of the Host's IP address is optional.
- Label in the context of a VXLAN forwarding plane is the VNI associated with the MAC-VRF/Layer 2 domain the advertised MAC address has been learned on.
- Route Target associated with the MAC-VRF advertised with route to allow the control of the import and export of routes.

The MAC mobility extended community, as discussed in the following section is used during MAC moves to update all VTEPs of the new location of the host.

18.2.2 ARP Suppression

Providing the option to advertise the MAC and IP binding in the type-2 route, ARP suppression can be supported on the remote VTEPs. The MAC to IP binding can be learned locally, via ARP snooping or DHCP traffic on the VTEP. Once the MAC and IP binding has been learned, it is advertised to the remote VTEPs as a type-2 route. This allows remote VTEPs to respond to any ARP requests for the host locally, thus reducing the amount of ARP traffic across the EVI.

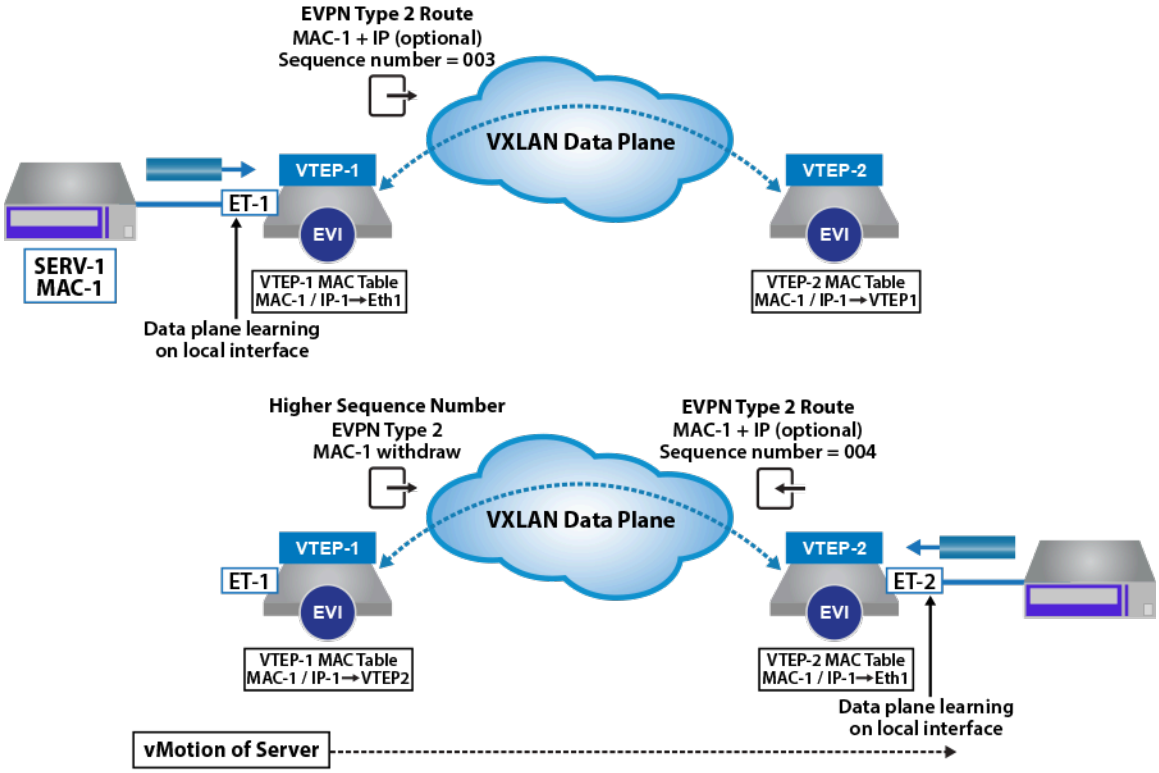
Importantly, the optional MAC and IP route can be advertised separately from the MAC only type-2 route. This is done so that if the MAC and IP route is cleared, i.e. ARP flushed, or the ARP timeout is set to less than the MAC timeout, then the MAC only route will still exist.

18.2.3 MAC Mobility

A common scenario in a data center environment is Virtual Machines (VMs) moving between physical servers, for maintenance or performance reasons, this will result in the MAC of the VM being learned and advertised by a new VTEP.

To cater for this situation a sequence number is attached to the new MAC advertisement ensuring an EVI wide refresh of the MAC table, with VTEPs updating their forwarding tables to point to the advertising VTEP as the new next-hop for MAC address.

Figure 18-14: EVPN type-2 MAC Mobility Behavior



When a MAC address is learned and advertised for the first time, it is advertised without a sequence number and the receiving VTEP assume the sequence to be zero. On detection of a MAC move, such as a MAC is learned locally when the same MAC route is active via a type-2 advertisement, then the sequence number is incremented by one, and the MAC route is advertised to the remote peers. The original advertising VTEP, receives the MAC route with a now higher sequence number and withdraws its own local MAC route. All other VTEPs flush the original MAC route, and update their tables with the new higher sequence number route.

18.2.4 MAC Address Damping

In addition to MAC mobility, EVPN defines a protection mechanism to detect and prevent MAC routes flapping between VTEPs, which can occur during network instability or when hosts have been mis-configured with the same (duplicate) MAC address.

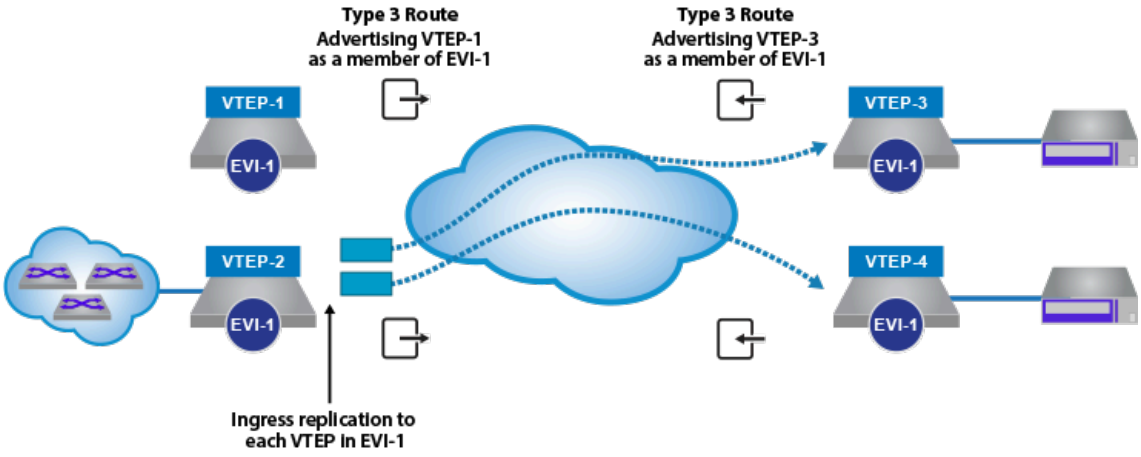
On advertising a locally learned MAC, the VTEP will start a M second counter (default is 180s), if the VTEP detects N MAC moves (default is 5) for the route within the M second window, it will generate a syslog message and stop sending and processing any further updates for the route.

18.2.5 Broadcast and Multicast Traffic

Broadcast, Unknown unicast and Multicast (BUM) traffic is handled within the EVPN forwarding model using ingress replication. Where the BUM frame is replicated on the ingress VTEP to each of the remote VTEPs in the associated EVI/VNI. The VTEP replication list for the EVI, is dynamically populated based on Type-3 route

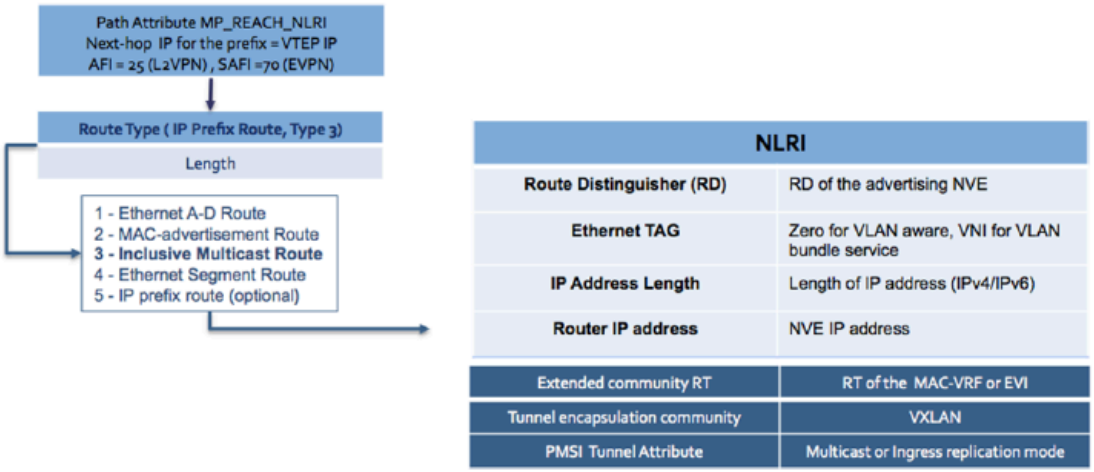
advertisements (Inclusive Multicast Ethernet Tag Route), where VTEPs advertise type-3 routes for each EVI they are members.

Figure 18-15: EVPN type-3 IMET Route Behavior for Ingress Replication



The format of the type-3 route is illustrated below.

Figure 18-16: EVPN Type-3 IMET Route Format



Notable fields of the type-3 route:

- Multi-protocol Reachable NLRI (MP_REACH_NLRI) attribute of the route is used to carry the next-hop hop for the advertised route. In the context of a VXLAN forwarding plane, this will be the source address (VTI) of the advertising VTEP.
- Route Distinguisher of the advertising node's MAC-VRF.
- Ethernet tag that will be 0 for VLAN-based service, and the MAC-VRF VNI for a VLAN-aware bundle service.
- IP address of the VTEP advertising the type 3 route.
- Route Target associated with the MAC-VRF or the EVI in a VLAN-aware bundle service.
- PMSI Tunnel Attribute, to advertise the replication model the VTEP is supporting. The supported options defined within the standard are ingress replication and IP multicast.

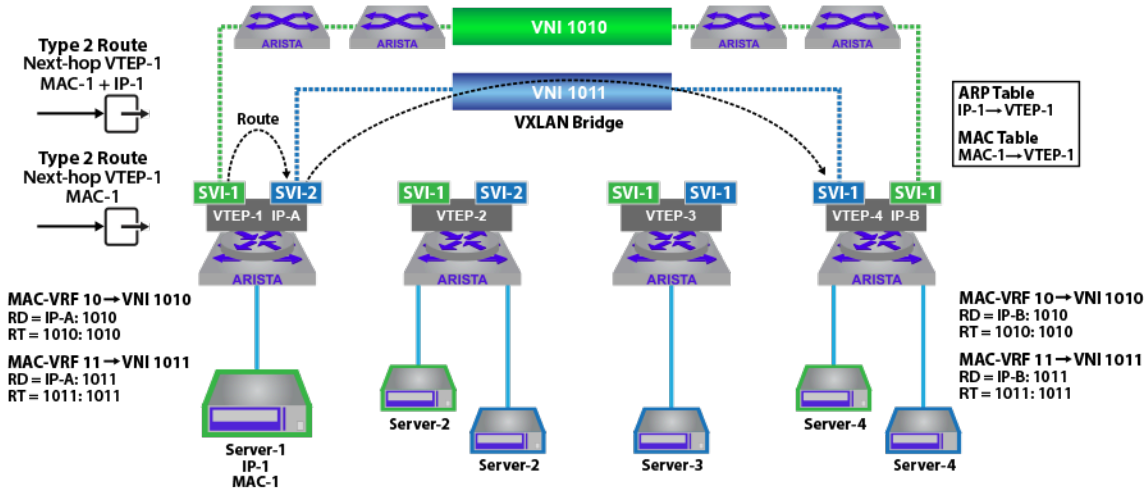
18.3 Integrated Routing and Bridging

In traditional data center design, inter-subnet forwarding is provided by a centralized router, where traffic traverses across the network to a centralized routing node and back again to its final destination. In a large multi-tenant data center environment this operational model can lead to inefficient use of bandwidth and sub-optimal forwarding.

To provide a more optimal forwarding model and avoid traffic tromboning, the IETF draft Integrated Routing and Bridging in EVPN proposes integrating routing and bridging functionality directly onto the VTEP, thereby allowing the routing operation to occur as close to the end host as possible. The draft proposes two forwarding models for the Integrated Routing and Bridging (IRB) functionality, which are termed asymmetric IRB and symmetric IRB. These two models are described in the following sections.

In the asymmetric IRB model, the inter-subnet routing functionality is performed by the ingress VTEP, with the packet after the routing action being VXLAN bridged to the destination VTEP. The egress VTEP only then needs to remove the VXLAN header and forward the packet onto the local Layer 2 domain based on the VNI to VLAN mapping. In the return path, the routing functionality is reversed with the destination VTEP now performing the ingress routing and VXLAN bridging operation, hence the term asymmetric IRB.

Figure 18-17: EVPN Asymmetric IRB



To provide inter-subnet routing on all VTEPs for all subnets, an anycast IP address is utilized for each subnet and configured on each VTEP. The anycast IP acts as the default gateway for the hosts, therefore regardless of where the host resides the directly attached VTEPs can act as the host's default gateway. The host MAC and MAC to IP bindings are learned by each VTEP based on a combination of local learning/ARP snooping and type-2 route advertisement from remote VTEPs.

In a typical implementation, the optional MAC and IP, type-2 route is advertised separately from the MAC only type-2 route. This is done so that if the MAC and IP route is cleared, for example the ARP flushed, or the ARP timeout is set to less than the MAC timeout, then the MAC only route will still exist.

The format of the two advertised type-2 routes for Server-1 are illustrated below, where the ***RD IP-A:1010*** and route-target ***1010:1010*** are used to distinguish the uniqueness of the route and allow the route to be imported into the correct remote MAC-VRF based on the route-target import policy of the VTEP.

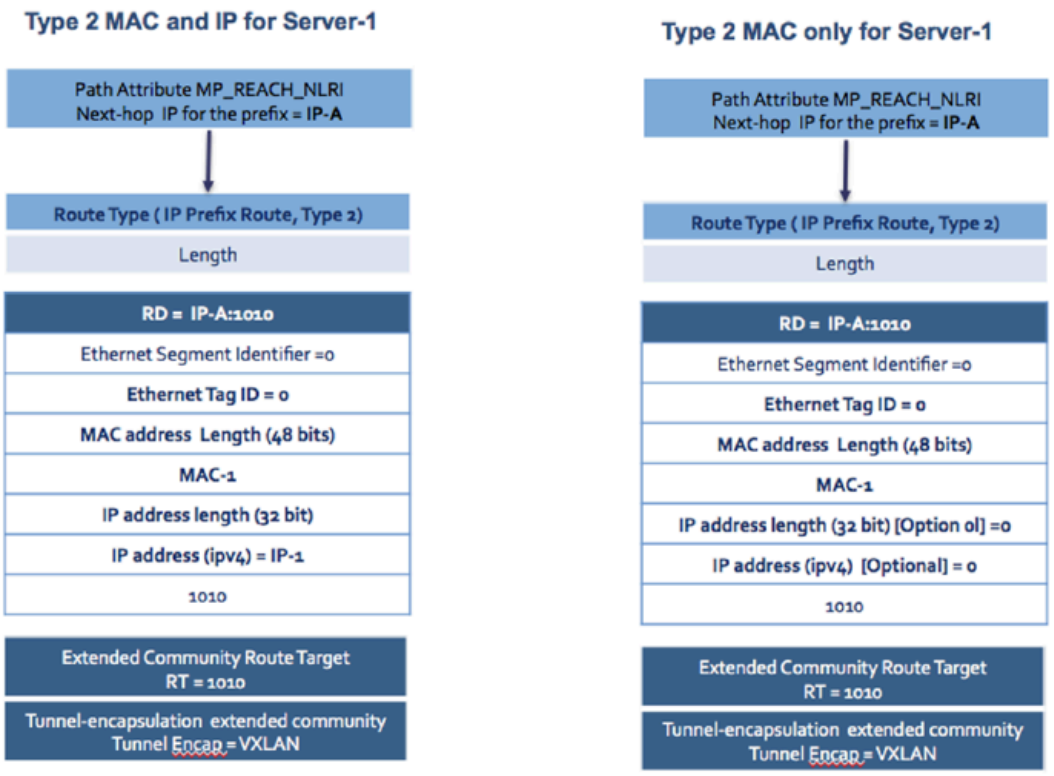
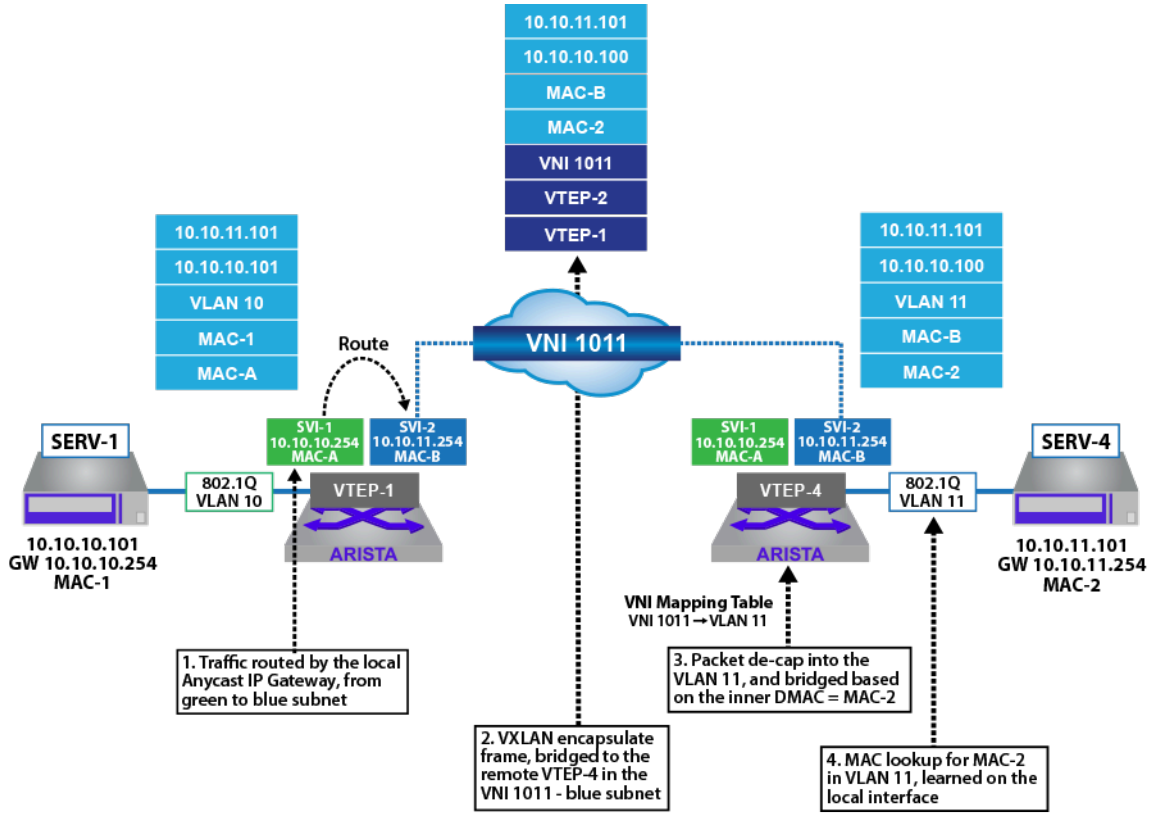


Figure 18-18: EVPN Comparison of MAC & MAC+IP Type 2 Route in Asymmetric IRB

For the traffic flow between ***Server-1*** in ***subnet-10*** and ***Server-4*** in ***subnet-11***, the ingress VTEP (***VTEP-1***) locally routes the packet into ***subnet-11/VNI 1011*** and then VXLAN bridges the frame, inserting the ***VNI 1011*** into the VXLAN header with an inner DMAC equal to the destination host, ***Server-4***. This requires the

receiving VTEP, (***VTEP-4***) to only perform a local Layer 2 lookup, based on the VNI to VLAN mapping, for the DMAC of ***Server-4***.

Figure 18-19: EVPN Asymmetric IRB VxLAN Data-plane Forwarding Detail



For the asymmetric model to operate the sending VTEP needs the information for all the tenant's hosts (MAC and MAC to IP binding), to route and bridge the packet. This means the VTEP needs to be member of all the tenant's subnets/VNI and have an associated SVI with anycast IP for all the subnets, and this will be required on all VTEPs participating in the routing functionality for the tenant. This introduces scaling issues on multiple fronts.

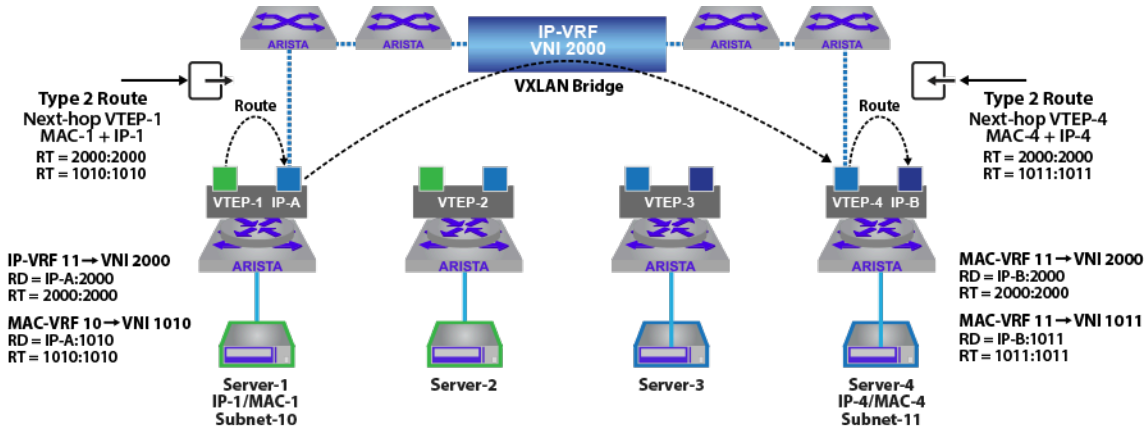
- **VNI Scaling**: The number of VNIs supported on a hardware VTEP will be finite, so not all VNIs can reside on all VTEPs. This is especially true in data-center deployments, where the TOR's have traditionally been more resource constrained than chassis-based edge systems.
- **Forwarding memory scaling**: The VTEPs needs to store all host MACs and ARP entries for all subnets in the network, on leaf switch this is hardware resource which again will be a finite resource defined by the specific hardware platform deployed at the leaf.

Symmetric IRB

To address the scale issues of the asymmetric model, in the symmetric model the VTEP is only configured with the subnets that are present on the directly attached hosts. Connectivity to non-local subnets on a remote VTEP is achieved through an intermediate IP-VRF. The subsequent forwarding model for symmetric IRB is illustrated in the figure below, for traffic between ***Server-1*** on ***subnet-10*** (Green) and ***Server-4*** on the remote ***subnet-11*** (Blue). In this model, the ingress VTEP routes the traffic between the local ***subnet-10***) and the IP-VRF, which both VTEPs are a member of, the egress VTEP then routes the frame from the IP-VRF to

the destination subnet. The forwarding model results in both VTEPs performing a routing function, hence the term symmetric IRB.

Figure 18-20: EVPN Symmetric IRB



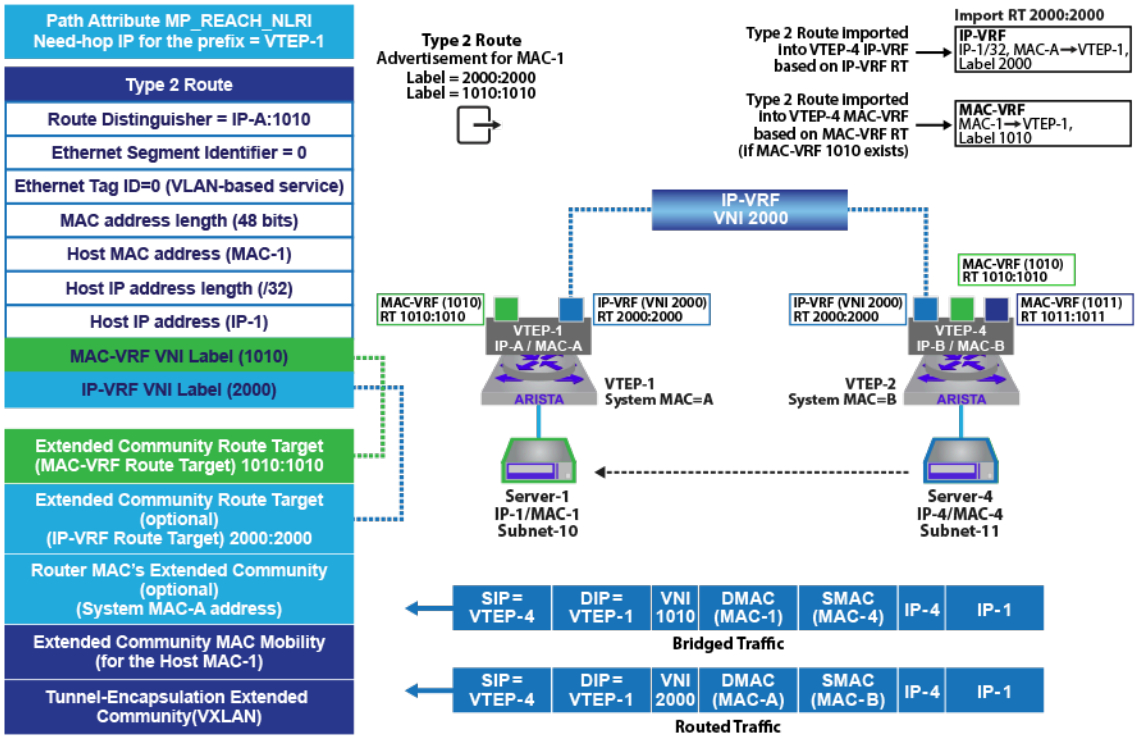
To provide the inter-subnet routing, when the subnet is stretched across multiple VTEPs, an anycast IP address is utilized for each subnet, but only configured on the VTEP's where the subnet exists. The host MAC and MAC to IP bindings are learned by each VTEP based on a combination of local learning/ARP snooping and type-2 route advertisements.

For the symmetric IRB model the type-2 (MAC and IP) route is advertised with two labels and two route-targets corresponding to the MAC-VRF the MAC address is learned on and the IP-VRF. Remote VTEP's receiving the route, import the IP host route into the corresponding IP-VRF based on the IP-VRF route-target and if the corresponding MAC-VRF exists on the VTEP the MAC address is imported into the local MAC-VRF based on the MAC-VRF's Route-Target. The import behavior for the type-2 route is illustrated in the diagrams below for the host Server-1.

If the MAC-VRF exists locally on the receiving router, both the IP host route will be installed in the IP-VRF, and the MAC address will be installed in the MAC-VRF. With both a MAC route in the MAC-VRF and an IP

host route in the IP-VRF, the VNI used in the data-path will depend on whether the traffic is being VXLAN bridged between hosts in the same VNI (***1010***) or VXLAN routed (***VNI 2000***).

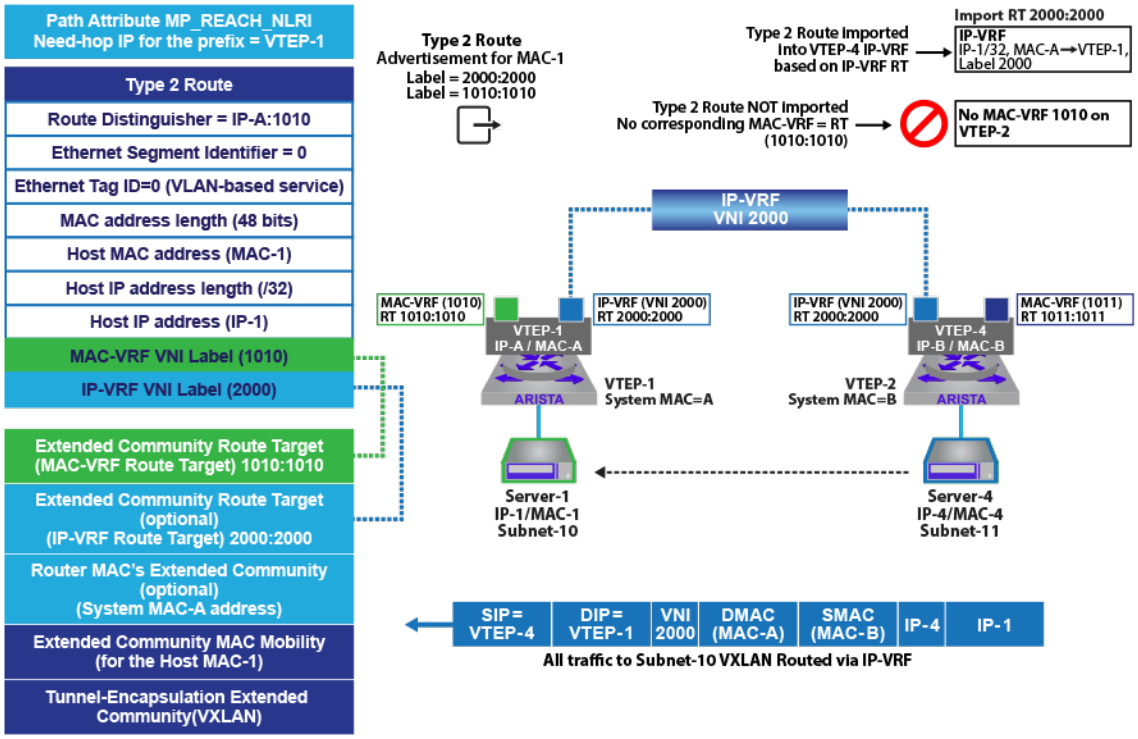
Figure 18-21: EVPN Type 2 Route in Symmetric IRB - MAC-VRF on Both VTEPs



Compare this to the figure below, where the MAC-VRF does not exist on the receiving VTEP (***VTEP-2***). In this case the MAC route is not installed and ignored, as there is no corresponding Route Target on the VTEP.

In this scenario, only the IP-VRF host route is installed on ***VTEP-2***. Traffic from ***VTEP-2*** destined to hosts on ***subnet-10***, are therefore always VXLAN routed via the IP-VRF, ***VNI 2000***.

Figure 18-22: EVPN Type 2 Route in Symmetric IRB - MAC-VRF Only Exists on Sending VTEP



The symmetric IRB type-2 route contains a number of additional extended community attributes over the asymmetric IRB type-2 route, the salient fields of the route are summarized below.

- Multi-protocol Reachable NLRI (MP_REACH_NLRI) attribute is used to carry the next-hop hop for the advertised route. In the context of a VXLAN forwarding plane, this will be the source address of the advertising VTEP.
- Route Distinguisher of the advertising node's MAC-VRF. For ***Server-1*** in the example above this would be ***IPA:1010***.
- MAC address field contains the 48-bit MAC address of the host being advertised. For ***Server-1*** in the example above this would be ***MAC-1***.
- IP address and length field contain the IP address and 32-bit mask for the host being advertised. For ***Server-1*** in the example above this would be ***IP-1***.
- MAC-VRF label, this contains the VNI number (label) corresponding to the local Layer 2 domain/MAC-VRF the host MAC was learned on. For ***Server-1*** in the example above this would be ***VNI 1010***.
- IP-VRF label, this contains the VNI number (label) corresponding to the MAC-VRF's associated IP-VRF. For ***MAC-VRF 10*** in the example above this would be ***IP-VRF 2000***.
- Extended community Route Target for the IP-VRF. This contains the route-target of the IP-VRF associated with the learned MAC address.
- Extended community Router MAC. This field advertises the system MAC of the advertising VTEP and is used as the DMAC for any packet sent to the VTEP via the IP-VRF.
- Extended community Route Target for the MAC-VRF. This contains the route-target of the MAC-VRF associated with the learned MAC address.

18.3.1 IP VPN

RFC 4364 allows Service Providers and Enterprises to use their backbone infrastructure to provide the services to multiple customers, or internal departments; while performing the following functions:

- Maintaining privacy.
- Allowing for IP address overlap amongst customers.
- Constraining route distribution - so that only the service provider routers which need the routes have them.

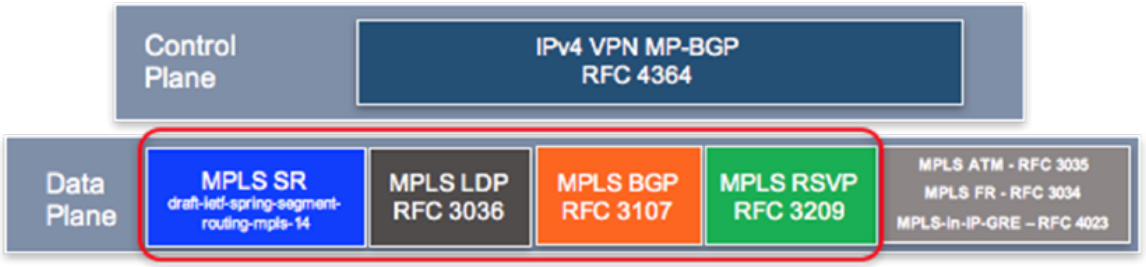
This is achieved through the usage of VRFs, Route Distinguishers and Route-Targets

The IPv4/IPv6 VPN Standard RFC 4364 does the following:

- Specifics an BGP IPv4 VPN control plane with a MPLS data plane.
- BGP control plane, new address family to advertise IP VPN prefixes.
- This RFC obsoleted the original ***RFC 2547***.
- MPLS data-plane defined in multiple RFCs and drafts.

The RED circle in the figure below highlights the main Drafts and RFCs in use today for an MPLS data-plane.

Figure 18-23: MPLS Data-Plane



IPv4 VPN and IPv6 VPN are an extensions of the BGP protocol introducing new address families: IPv4 (address family number ***1***), IPv6 (address family number ***2***), and a subsequent address family number ***128***: MPLS Layer 3 VPN unicast. It is used to exchange overlay IP prefix reachability information between MP-BGP peers.

Figure 18-24: IPv4 VPN and IPv6 VPN

AFI	Description
1	**IPv4**
2	IPv6
25	L2VPN (MAC addr)

+

SAFI	Description
1	unicast forwarding
2	multicast forwarding
70	EVPN
128	**MPLS L3VPN unicast**
129	MPLS L3VPN multicast
65	VPLS

=

AFI	SAFI	Description
1	1	IPv4 unicast forwarding
2	1	IPv6 unicast forwarding
25	70	BGP EVPN
1	**128**	**IPv4 L3VPN unicast**
2	128	IPv6 L3VPN unicast
25	65	MPLS VPLS

IPv4 VPN defines two route types:

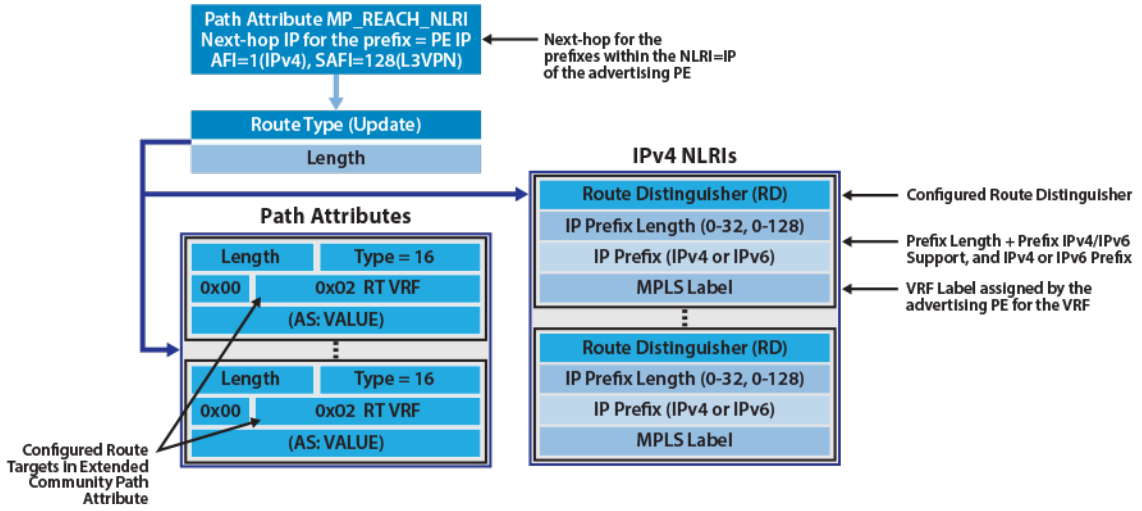
- Update
- Withdrawal

Each route type has its own NLRI prefix format and ach route type advertises its own set of prefixes to update/withdraw.

The format of the IPv4 VPN prefix update route is illustrated in the following figure. As detailed, the update route contains the VPN route (prefix and RD), the next-hop for the route and the advertising router ID, along

with the MPLS Label, along with a number of path attributes (where the RT extended communities are defined), which are associated with these IPv4 NLRIs.

Figure 18-25: IPv4 and IPv6 VPN Update Route Detail



The output in IPv4 VPN route as shown on PE, and the IPv6 VPN route as shown on PE offers a more detailed view of the route as displayed on a PE router.

Figure 18-26: IPv4 VPN Route as Shown on PE

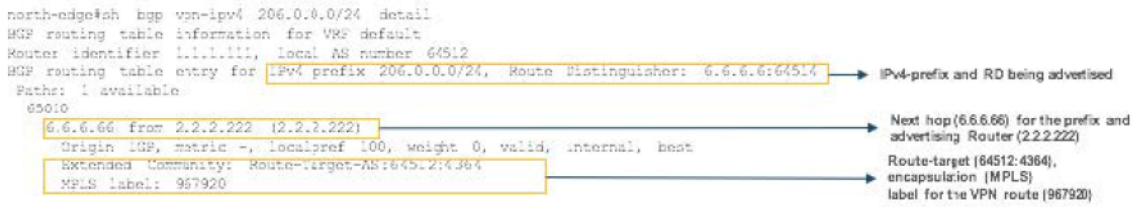
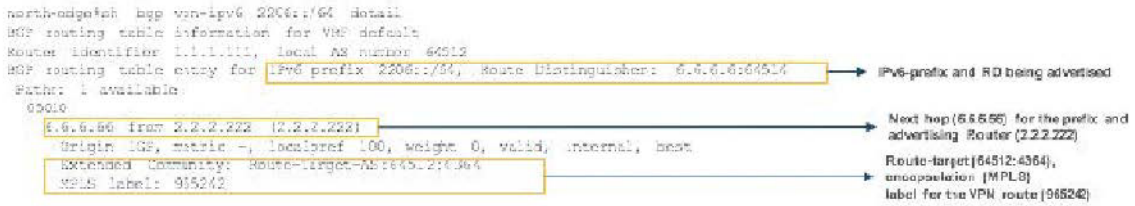
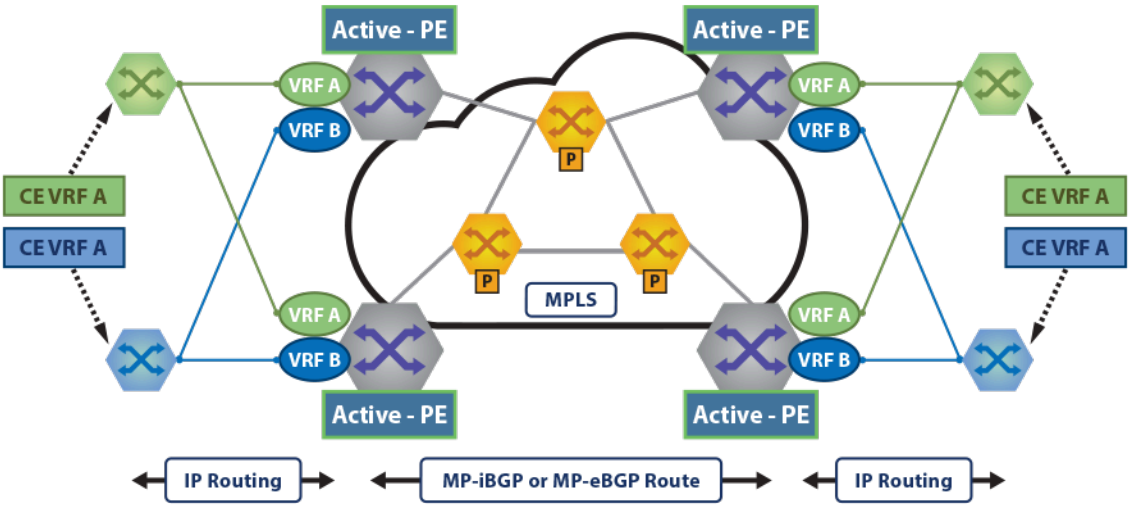


Figure 18-27: IPv6 VPN Route as Shown on PE



The following is an illustration of a basic MPLS Layer 3 VPN topology.

Figure 18-28: MPLS Layer 3 VPN Topology



An IP VRF is used on a PE router for each customer (Layer 3 overlay). VRF IP routes are exported into the MP-BGP table and advertised to remote PEs as VPN routes. The exported VPN routes carry the Route-Target (RT) extended communities that are configured as export route-targets on the IP VRF from which they were exported.

The RTs carried by the VPN routes received by a PE are matched against the VRF import route-target configuration. When a received route carries an RT that is configured as an import route-target on an IP VRF, the route is imported into the IPv4 or IPv6 table for that VRF.

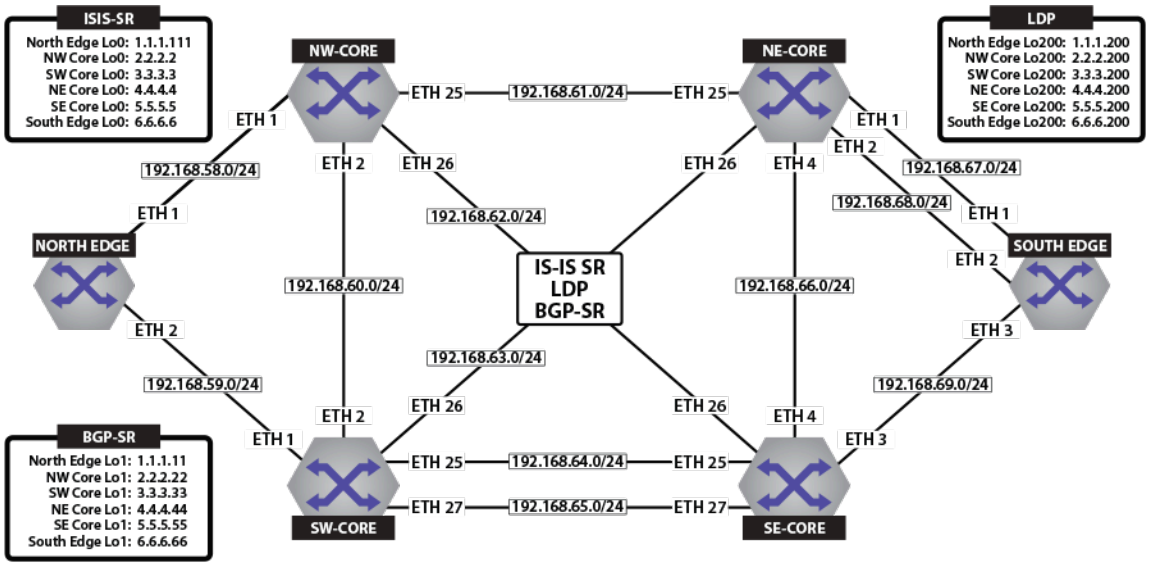
PE routers allocate per-VRF and address family Labels that are advertised as part of the VPN route NLRI. Forwarding of overlay packets between PEs across the underlay requires underlay MPLS connectivity provided by a backbone.

Note: You require Release ***EOS 4.21.1F*** and later versions with Jericho/Jericho+ platforms.

18.4 VPN MPLS Transport Options

EVPN-MPLS and IP-VPN sample topologies illustrate co-existing LDP, BGP-SR, and ISIS-SR on the core.

Figure 18-29: Physical Topology For ISIS-SR, LDP and BGP-SR Transport



LDP, **ISIS-SR**, and **BGP-LU (BGP-SR)** demonstrates the corresponding Label Switched Paths (LSPs) as the MPLS transport LSPs for Layer3 EVPN and IP VPN services.

EVPN Sample Topology

In the figures below **Tenant-A DCI** and **Tenant-B DCI**, the prefixes from each DC are transported over the WAN/DCI domain, maintaining the Layer 3 multi-tenancy in tenant-a and tenant-b.

Figure 18-30: Tenant-A DCI

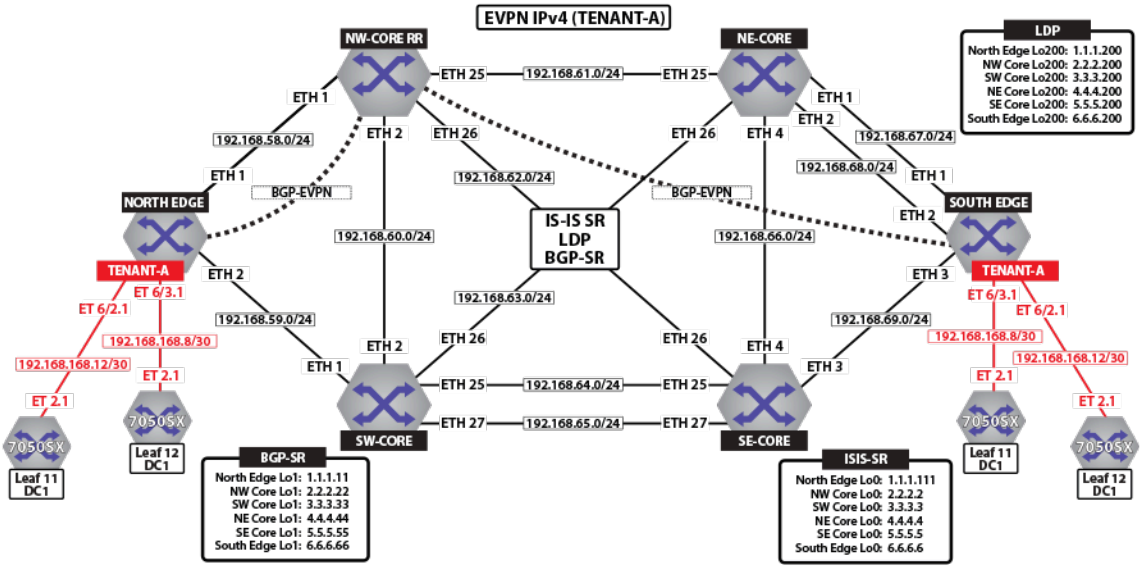
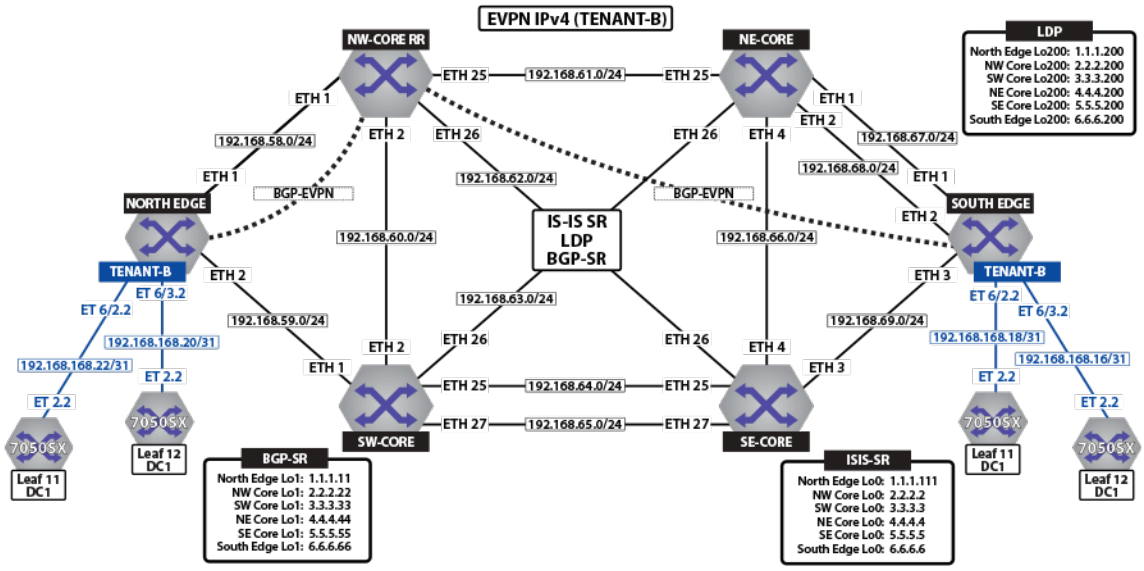


Figure 18-31: Tenant-B DCI



To provide external connectivity from the DC into the MPLS domain, leaf-11 and leaf-12 are eBGP peering via the tenants VRFs with the border routers. Both core routers are advertising external prefixes for Internet and any remote site connectivity (default route and ip-prefixes from the other DC for the tenant). To provide connectivity within the EVPN domain, the leaf switches (leaf-21 and leaf-2) re-advertise the prefixes into the tenant's VRF via a type-5 route advertisement, with a next-hop equal to the advertising PE.

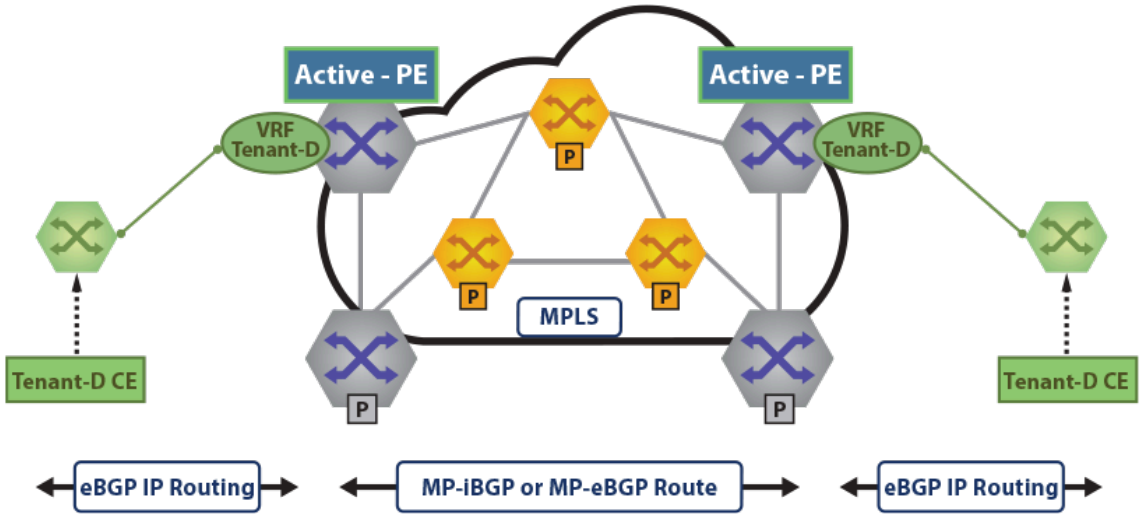
Let us review the concepts of transport labels, advertised to provide the label switched path, or LSP, across the back-bone and the VPN, or tenant label, used by the Provider Edge (PE) routers to identify a particular tenant.

EVPN MPLS Sample Configuration displays BGP route updates and how the tenant VRF is transported over these transport LSPs.

IP VPN Sample Topology

Let us review the concepts of transport labels, advertised to provide the label switched path, or LSP, across the back-bone and the VPN, or tenant label, used by the Provider Edge (PE) routers to identify a particular tenant.

Figure 18-32: IPv4 & IPv6 VPN Sample Topology



In the figures **Tenant-D IPv4 VPN** and **Tenant-D IPv6 VPN**, the prefixes for VRF tenant-d are transported over the MPLS WAN between North Edge and South Edge routers.

Figure 18-33: Tenant-D IPv4 VPN

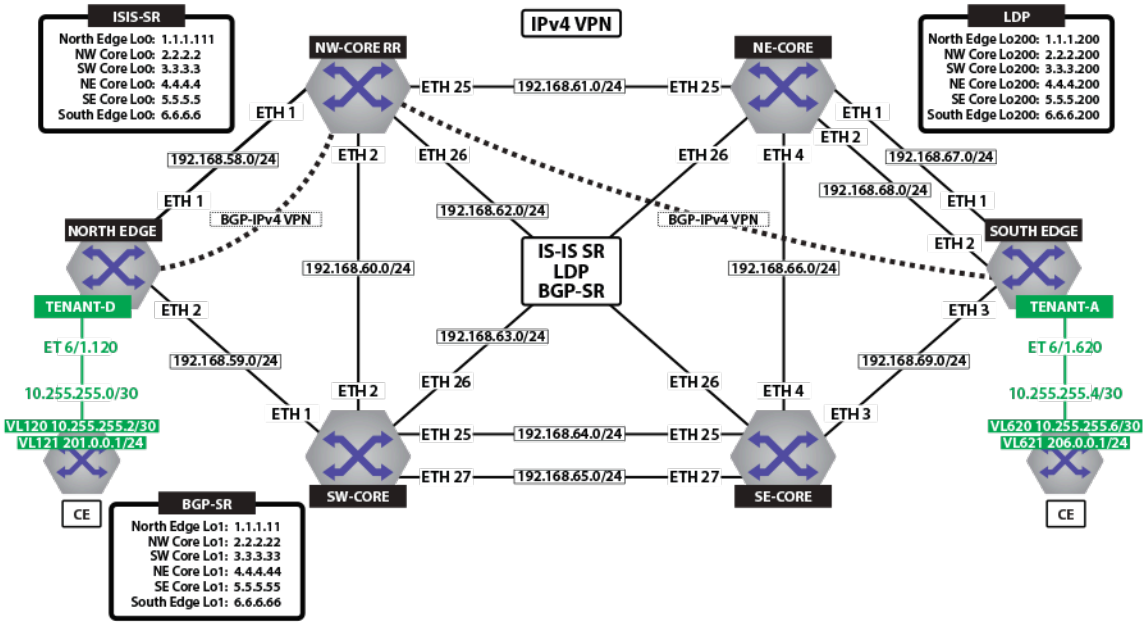
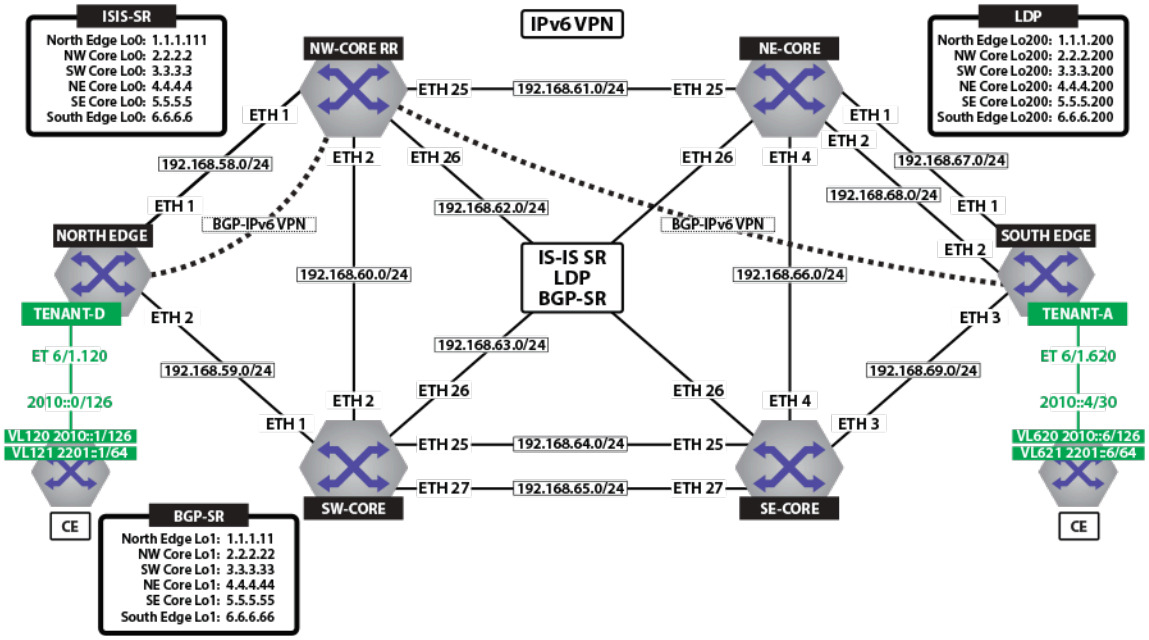


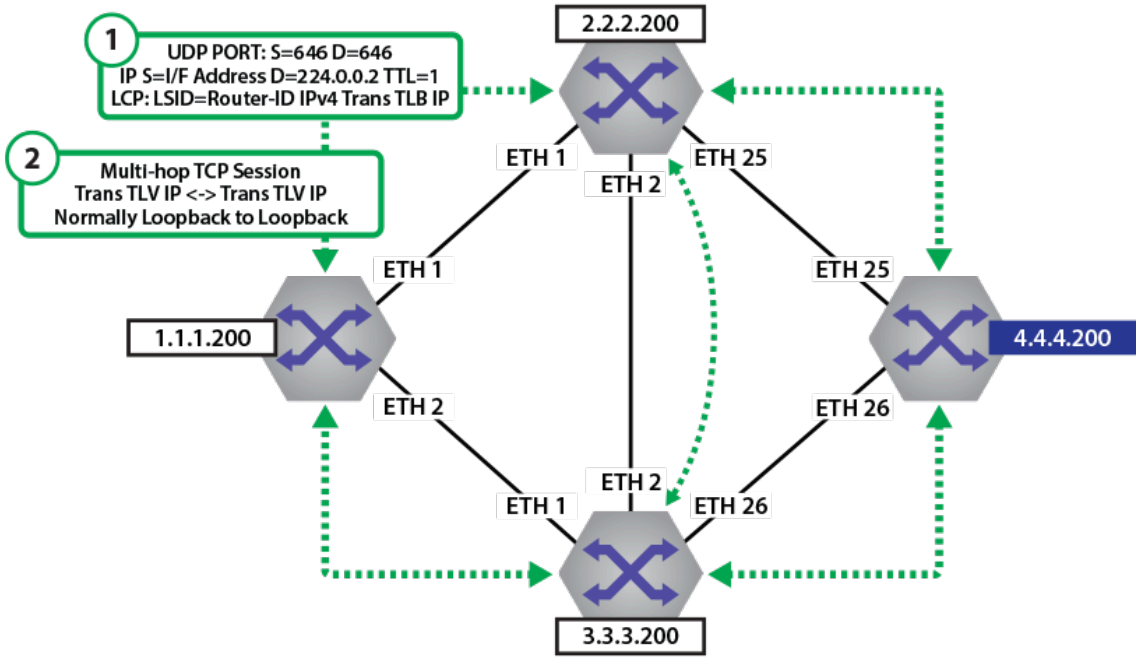
Figure 18-34: Tenant-D IPv6 VPN



18.4.1 LDP

The figure below illustrates how LDP neighbor relationships are built. First each router sends a discovery to a destination multicast address (TTL=1) ***224.0.0.2*** on ***port 646***. This discovery contains the router-id and the transport IPv4 address the router wants to use. The second stage is building the TCP peering session using the transport IP addresses specified. This is normally loopback to loopback.

Figure 18-35: LDP Peering Establishment



Examples

- The **`show mpls ldp neighbor`** command on the North Edge router displays more detail on TCP session establishment, and the local addresses of the LDP neighbor for which it is binding a label.

  Note: All connected interfaces are advertised as bound. However, EOS currently advertised labels for /32 addresses, and FEC filter is configured to install only x.x.x.200/32 prefixes.

```
North Edge.17:51:17# show mpls ldp neighbor
Peer LDP ID: 2.2.2.200:0; Local LDP ID: 1.1.1.200:0
   TCP Connection: 2.2.2.200:38395 - 1.1.1.200:646
   State: oper; Msgs sent/rcvd: 46/46; downstream unsolicited
   Uptime: 0:06:17
   KeepAlive expires in: 20.27 sec
   LDP discovery sources:
      Ethernet1/1
   Addresses bound to peer:
      2.2.2.200          2.2.2.2            192.168.1.177    192.168.62.11
      192.168.1.181      192.168.58.12      192.168.60.11    192.168.61.11
Peer LDP ID: 3.3.3.200:0; Local LDP ID: 1.1.1.200:0
   TCP Connection: 3.3.3.200:38510 - 1.1.1.200:646
   State: oper; Msgs sent/rcvd: 42/42; downstream unsolicited
   Uptime: 0:05:51
   KeepAlive expires in: 20.02 sec
   LDP discovery sources:
      Ethernet2/1
   Addresses bound to peer:
      192.168.65.11      192.168.59.12      3.3.3.200        192.168.60.12
      192.168.63.11      3.3.3.3            192.168.64.11
```

- The **`show mpls lfib route 116384`** command on the North Edge router displays the label POP and swap operations for any traffic traversing North Edge. As can be seen if traffic came in with label ***116384*** it would be swapped to the labels seen in the tunnel table.

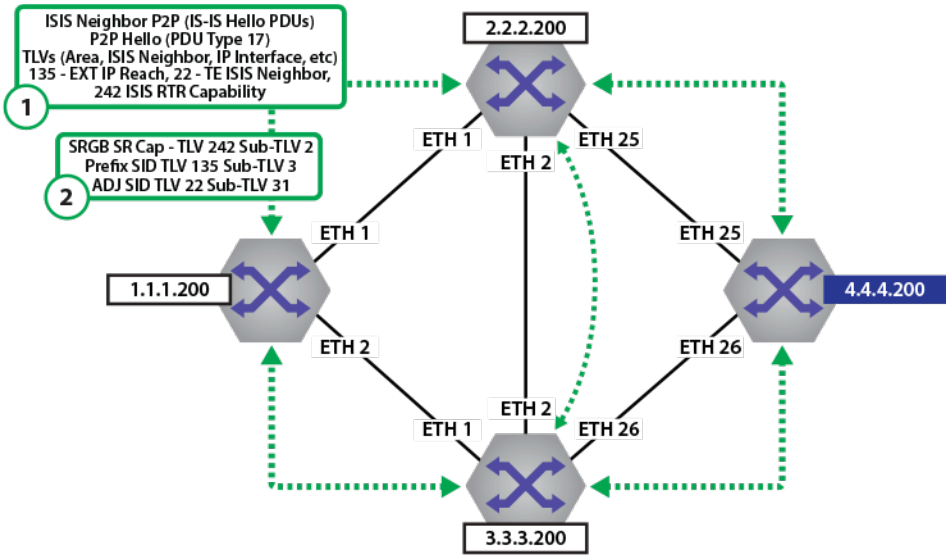
```
North Edge.23:38:28(config)# show mpls lfib route 116384
MPLS forwarding table (Label [metric] Vias) - 1 routes
MPLS next-hop resolution allow default route: False
Via Type Codes:
          M - Mpls Via, P - Pseudowire Via,
          I - IP Lookup Via, V - Vlan Via,
          VA - EVPN Vlan Aware Via, ES - EVPN Ethernet Segment Via,
          VF - EVPN Vlan Flood Via, AF - EVPN Vlan Aware Flood Via
Source Codes:
          S - Static MPLS Route, B2 - BGP L2 EVPN,
          B3 - BGP L3 VPN, P - Pseudowire,
          L - LDP, IP - IS-IS SR Prefix Segment,
          IA - IS-IS SR Adjacency Segment, IL - IS-IS SR Segment to LDP,
          LI - LDP to IS-IS SR Segment, BL - BGP LU,
          DE - Debug LFIB

 L   116384   [1], 6.6.6.200/32
                via M, 192.168.58.12, swap 132768
                    payload autoDecide, ttlMode autoDecide, apply egress-acl
                    interface Ethernet1/1
                via M, 192.168.59.12, swap 100000
                    payload autoDecide, ttlMode autoDecide, apply egress-acl
                    interface Ethernet2/1
```

18.4.2 ISIS-SR

The figure below illustrates how ISIS-SR distributes the SID index information in the ISIS TLVs and sub-TLVs

Figure 18-36: ISIS Neighbor Adj and TLVs



The Prefix SID index, SRGB, and ADJ SID values are populated in the sub-TLVs in the ISIS neighbor updates. Each router then builds its own database of Node (Prefix) segments (Labels) and locally assigned ADJ labels.

Examples

- The **`show isis neighbors detail`** command on the North Edge router displays the detailed information of all ISIS neighbors.

```
north-edge# show isis neighbors detail
Instance  VRF      System Id        Type Interface          SNPA            State Hold time   Circuit Id
sr_instan default  nw-core          L2   Ethernet1/1        P2P             UP    30          1D
  Area Address(es): 49.0001
  SNPA: P2P
  Advertised Hold Time: 30
  State Changed: 6d17h ago
  IPv4 Interface Address: 192.168.58.12
  IPv6 Interface Address: none
  Interface name: Ethernet1/1
  Graceful Restart: Supported
  Segment Routing Enabled
    Router ID: 2.2.2.2
    SRGB Base: 408000 Range: 4096
    Adjacency Label IPv4: 953252
sr_instan default  sw-core          L2   Ethernet2/1        P2P             UP    28          1E
  Area Address(es): 49.0001
  SNPA: P2P
  Advertised Hold Time: 30
  State Changed: 00:06:06 ago
  IPv4 Interface Address: 192.168.59.12
  IPv6 Interface Address: none
  Interface name: Ethernet2/1
  Graceful Restart: Supported
  Segment Routing Enabled
    Router ID: 3.3.3.3
    SRGB Base: 408000 Range: 4096
    Adjacency Label IPv4: 953253
```

- The **`show isis segment-routing adjacency-segments`** command on the North Edge router displays the locally assigned Adjacency Segment Identifier (Adj-SIDs).

```
North Edge# show isis segment-routing adjacency-segments

System ID: north-edge                  Instance: sr_instance
SR supported Data-plane: MPLS                 SR Router ID: 1.1.1.111
Adj-SID allocation mode: SR-adjacencies
Adj-SID allocation pool: Base: 953249     Size: 16384
```

```
Adjacency Segment Count: 5
Flag Descriptions: F: Ipv6 address family, B: Backup, V: Value
                   L: Local, S: Set

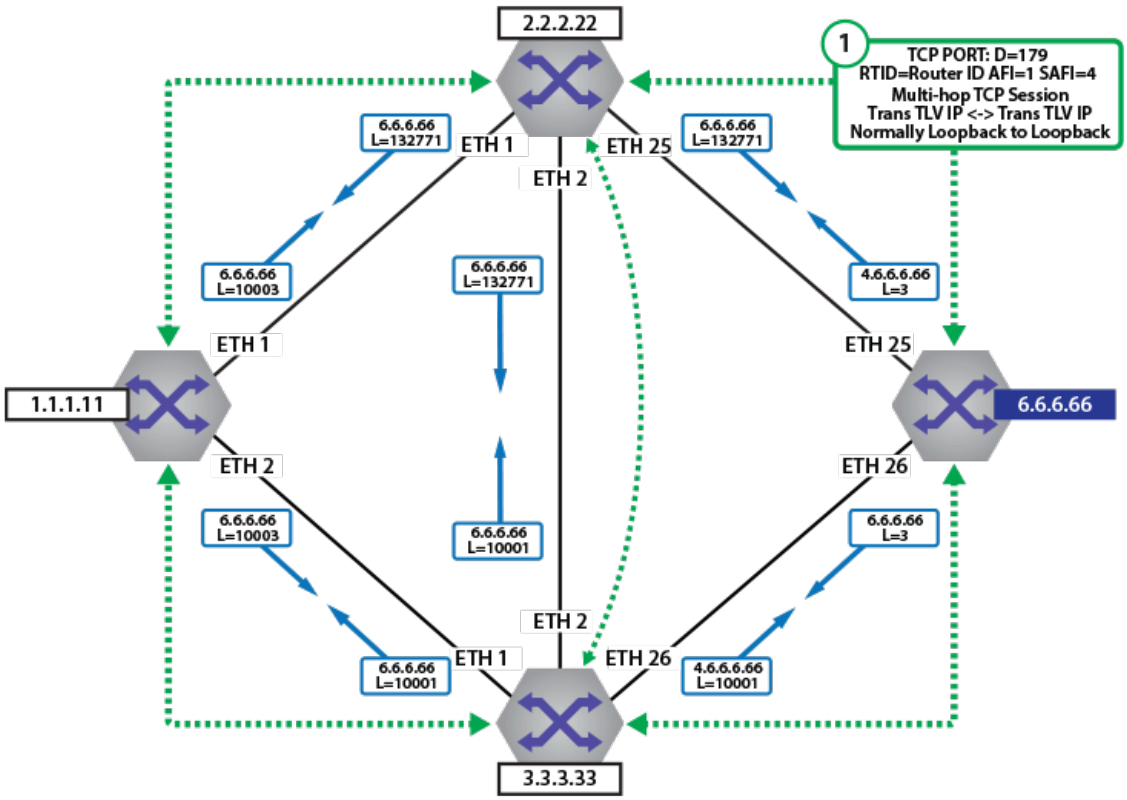
Segment Status codes: L1 - Level-1 adjacency, L2 - Level-2 adjacency, P2P - Point-to-Point adjacency, LAN -
Broadcast adjacency

Locally Originated Adjacency Segments
   Adj IP Address       Local Intf          SID       SID Source                    Flags       Type
------------------- ---------------- ------------ ---------------- ------------------------- ------
    192.168.1.154          Et36/1       953249          Dynamic       F:0 B:0 V:1 L:1 S:0    P2P L2
    192.168.1.174          Et23/1       953250          Dynamic       F:0 B:0 V:1 L:1 S:0    P2P L2
    192.168.58.12           Et1/1       953252          Dynamic       F:0 B:0 V:1 L:1 S:0    P2P L2
    192.168.59.12           Et2/1       953253          Dynamic       F:0 B:0 V:1 L:1 S:0    P2P L2
    192.168.1.165           Et8/1       953254          Dynamic       F:0 B:0 V:1 L:1 S:0    P2P L2
```

18.4.3 BGP-LU (BGP-SR)

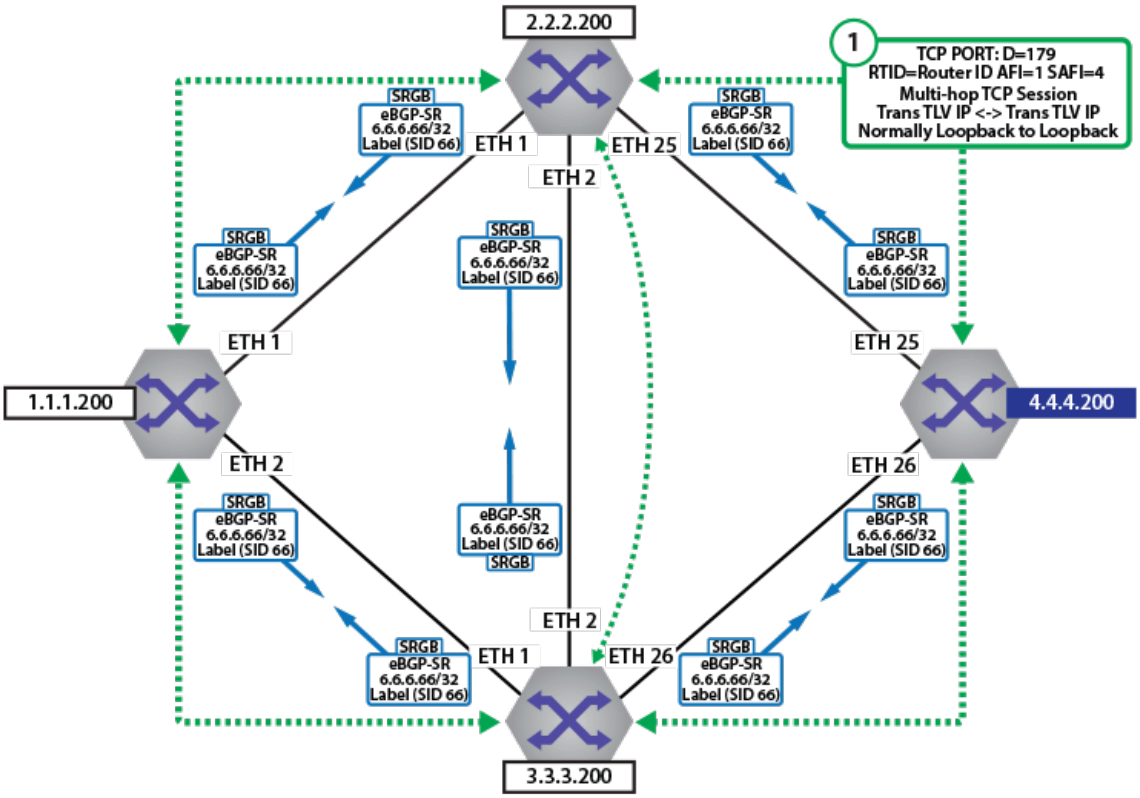
BGP-LU Label Distribution illustrates how BGP-LU distributes the label information in BGP.

Figure 18-37: BGP-LU Label Distribution



BGP-SR Index and SRGB Distribution illustrates how BGP-LU distributes the Label SRGB and SID index information in BGP. This is known as BGP-SR.

Figure 18-38: BGP-SR Index and SRGB Distribution



The Prefix SID index, and SRGB values are populated in the TLVs in the BGP neighbor updates. Each router then builds its own database of Node (Prefix) segments (Labels).

Examples

- The **`show bgp neighbor`** command displays BGP-SR neighbors.

```
north-edge# show bgp neighbor | include BGP neighbor|Multiprotocol IPv4
 MplsLabel

BGP neighbor is 192.168.2.10, remote AS 64512, internal link
    Multiprotocol IPv4 MplsLabel: received
BGP neighbor is 192.168.3.9, remote AS 64512, internal link
    Multiprotocol IPv4 MplsLabel: advertised and received and negotiated
BGP neighbor is 192.168.3.10, remote AS 64512, internal link
    Multiprotocol IPv4 MplsLabel: advertised
BGP neighbor is 192.168.58.12, remote AS 2, external link
    Multiprotocol IPv4 MplsLabel: advertised and received and negotiated
BGP neighbor is 192.168.59.12, remote AS 3, external link
```

- The **`show ip bgp labeled-unicast 6.6.6.66/32 detail`** command displays the detailed information of BGP labeled routes unicast with ***6.6.6.66/32***.

```
north-edge(config-if-Et2/1)# show ip bgp labeled-unicast 6.6.6.66/32 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for 6.6.6.66/32
 Paths: 2 available
  2 4 6
```

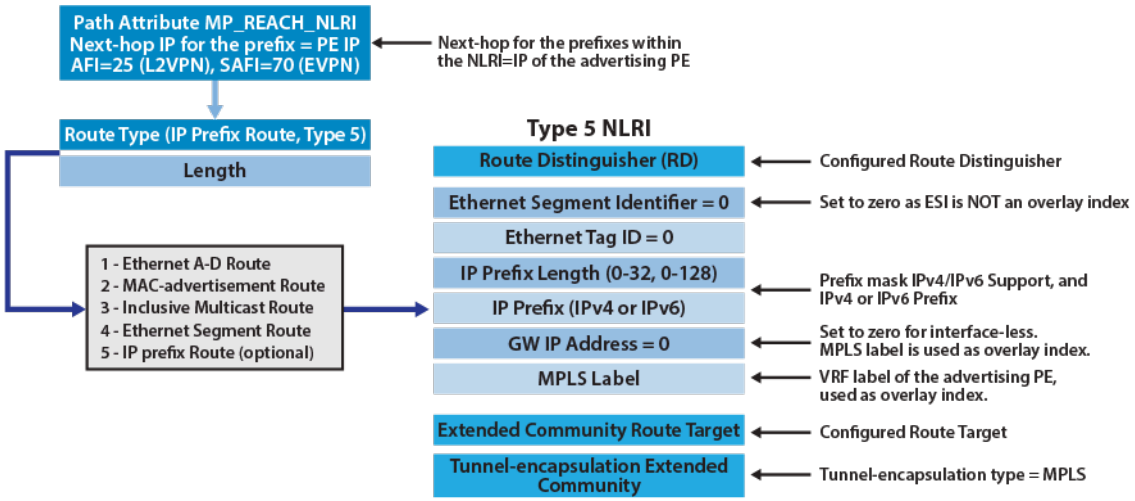
```
    192.168.58.12 labels [ 200066 ] from 192.168.58.12 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP
head, best, ECMP contributor
      Local MPLS label: 200066, SR Label Index: 66
 3 4 6
    192.168.59.12 labels [ 200066 ] from 192.168.59.12 (3.3.3.200)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP,
ECMP contributor
      Not best: ECMP-Fast configured
      Local MPLS label: 200066, SR Label Index: 66
Advertised to 2 peers:
    192.168.3.9       192.168.59.12
```

18.5 EVPN Type-5 Routes: IP Prefix Advertisement

The EVPN type 2 routes can be used to advertise IP prefixes by making use of the optional IP address and IP address length fields in the route, however they are explicitly linked to the MAC address advertised within the route. The EVPN type-5 route defined within the IETF draft, provides the ability to decouple the advertisement of an IP prefix from any specific MAC address, providing the ability to support floating IP address, optimize the mechanism for advertising external IP prefixes, and reduce the churn when withdrawing IP prefixes.

The figure below displays the format of the new type-5 IP-prefix route.

Figure 18-39: EVPN Route Type-5, for Advertisement of IP Prefixes



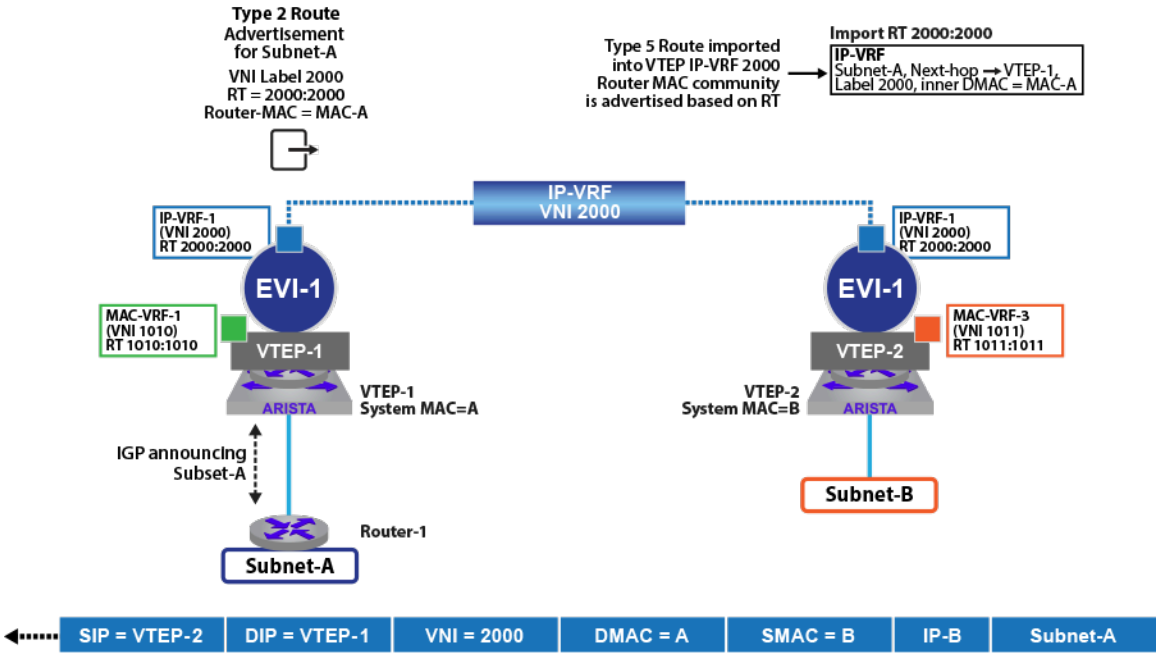
The IP prefix draft defines a number of specific use cases for the type-5 route, which consequently affect the format and content of the fields within the route. The different deployment scenarios and use cases defined within the draft are summarized below.

- Advertising of IP prefixes behind an appliance, when the appliance is not running a routing protocol and only supporting static routes. This could be the typical use case for a Virtual Firewall with a number of local subnets directly attached, but the firewall is only supporting static routes into the associated EVI.
- Support for active-standby deployment of appliances using a shared floating IP model. This is an extension of the previous case where there is now a virtual IP (or VIP) for clustering the appliances, rather than a dedicated physical IP address on the appliance.
- Support for Layer 2 appliances, acting as a “bump in the wire” with no physical IP addresses configured, where instead of the appliances having an IP next-hop there is only a MAC next-hop.
- IP-VRF to IP-VRF model, which is similar to inter-subnet forwarding for host routes (detailed in the symmetric/asymmetric section), except only Type-5 routes and IP prefixes are advertised, allowing announcement of IP prefixes into a tenant’s EVI domain for external connectivity outside the domain.

Interface-less

In interface-less mode, the IP prefixes within the type-5 route, whether they are local or learned from a connected router are advertised to remote peers via the shared IP-VRF, as illustrated in the figure below.The IP-VRF to IP-VRF model, is further divided in the draft into three distinct use cases.

Figure 18-40: EVPN Route Type-5, Interface-less Update

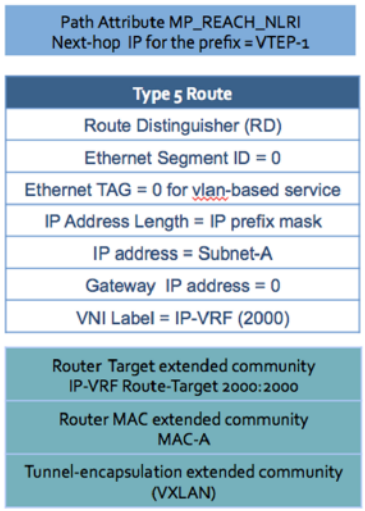


As illustrated above, the IP prefix (***subnet-A***) residing behind the router (***Rtr-1***) is learned via an IGP in EVI-1 on VTEP-1. The prefix is announced and learned by the remote VTEPs residing in the same EVI, via the type-5 route announcement. The type-5 route, is advertised along with the prefix, with a route-target (***2000:2000***) and a VNI label (***2000***) equal to the IP-VRF which interconnects the VTEPs in the EVI, the router-mac extended community of the route is used to define the inner DMAC (equal to system MAC of ***VTEP-1***) for any VXLAN frame destined to advertised IP prefix.

From a forwarding perspective, host residing on ***subnet-B*** communicating with a host on ***subnet-A***, will send traffic to their default gateway which is the IRB interface on ***VTEP-2*** in ***VLAN 11/VNI 1011***. ***VTEP-2*** performs a route lookup for the destination ***subnet-A***), which has been learned in the IP-VRF with a next-hop of ***VTEP-1*** and VNI label of ***2000***. The packet is thus VXLAN encapsulated with VNI label of ***2000*** an inner DMAC of A (***VTEP-1*** system/router MAC), and routed to ***VTEP-1***, which is the next-hop for the prefix. Receiving the frame, ***VTEP-1*** de-encapsulates the packet, with an inner DMAC of the VTEPs router MAC, it performs a local route lookup for the destination ***subnet-A***), which has been learned with a next-hop of ***rtr-1***.

The frame is forwarded directly to ***rtr-1***, which subsequently routes the packet to the local host on subnet-A. The format of the type-5 route in interface-less mode is illustrated in figure below.

Figure 18-41: EVPN Type-5 Route Format for Interface-less Mode



In this model, the VTEPs forming the EVI are interconnected via an IP-VRF, meaning there is no IRB interface (MAC and IP) created for the interconnection on each of the VTEPs, hence the term “interface-less”. With no IRB interface the gateway IP address within the type-5 route is set to ***0***, traffic is routed to the prefix based on the next-hop of the route (VTEP IP) as well as MAC address conveyed within the Router MAC extended community, which represents the inner destination MAC of the VXLAN encapsulated frame.

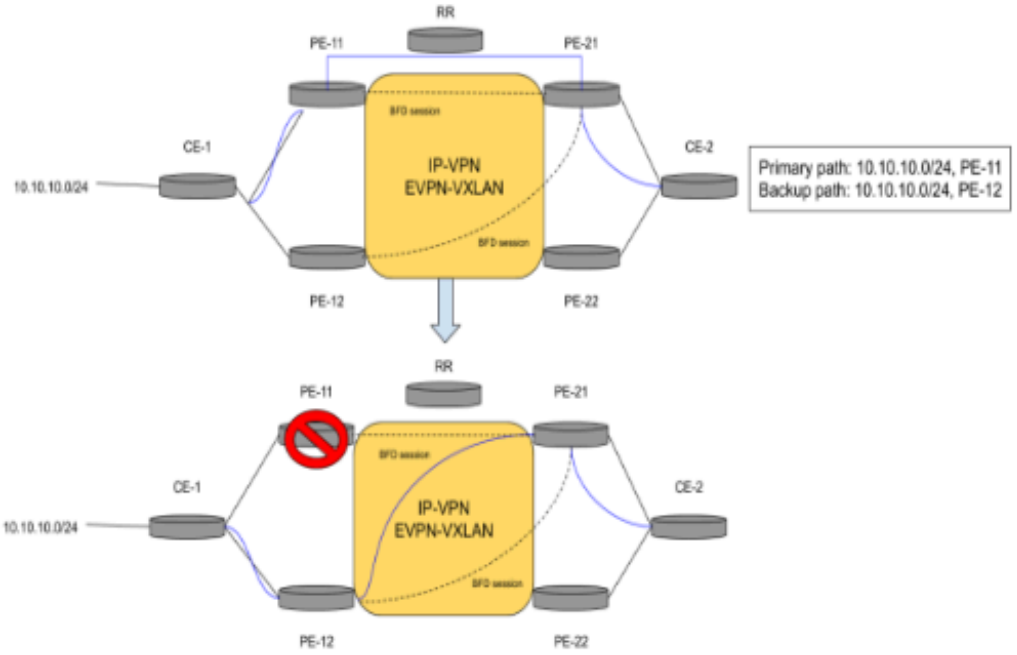
18.6 BGP PIC Edge for EVPN VXLAN Routes for Remote VTEP Failures

When a remote VTEP goes down, this would require action by the IGP and BGP to recompute a new best path traffic destined to affected BGP prefixes originally reachable by the problematic VTEP. Currently, the BGP PIC is restricted to locally identifiable failures such as link failures.

To overcome such VTEP failure issues the above feature introduces support for EVPN learned VTEPs to improve convergence times in these scenarios by tying the liveness detection provided by the BFD sessions into existing BGP PIC support for software fast-failover. Without this feature, until the underlay route providing reachability to the problematic VTEP is removed from the FIB the overlay route could still forward traffic towards this VTEP. Once the underlay route is eventually removed the adjacency would be updated in place to avoid per-prefix updates so that any overlay routes containing the path to this VTEP would be corrected. However with this feature, upon detecting that a BFD session to a remote VTEP has gone down the hardware

forwarding agents will update the affected adjacencies before the corresponding underlay route has been removed from the FIB which can improve convergence times.

Figure 18-42: BGP PIC Edge for VXLAN



The above diagram outlines a scenario in which ***CE-2*** is sending traffic bound for ***10.10.10.0/24*** via ***PE-12*** to ***PE-11*** to ***CE-1***. ***PE-11*** goes down, however we have BFD sessions from ***PE-21*** to the remote VTEPs of ***PE-11*** and ***PE-12*** and therefore detect that it goes down and quickly update the forwarding to send the traffic along the pre-computed backup path via ***PE-12*** to ***CE-1***.

18.6.1 Configuring BGP PIC Edge for EVPN VXLAN Routes for Remote VTEP Failures

Use the **`bfd vtep evpn`** command to configure the BGP PIC Edge for EVPN VXLAN routes for remote VTEPs. This command is configured under the VXLAN Tunnel Interface (VTI).

```
switch# config
switch(config)# interface Vxlan1
switch(config-if-Vx1)# bfd vtep evpn interval <interval> min-rx <min-rx>
 multiplier <multiplier>
```

This configuration uses the specified timer values to initiate BFD sessions for all VTEPs learned through EVPN VXLAN for this VTI.

- interval – Transmit rate in milliseconds
- min-rx – Expected minimum incoming rate in milliseconds
- multiplier – BFD multiplier

Example

```
switch(config-if-Vx1)# bfd vtep evpn interval 100 min-rx 100 multiplier 3
```

In this example (assuming symmetric configuration on other PE devices) any BFD for VXLAN session initiated on the VTI would have a detect time of ***300ms*** (interval of 100ms multiplied by 3).

To utilize these BFD sessions, there must also be an alternate path traffic can take in the event that the session goes down. This would include other paths in an ECMP group or a backup path.

As mentioned, by default the above configuration will initiate BFD sessions for all VTEPs learned through EVPN VXLAN for the VTI.

```
switch# config
switch(config)# interface Vxlan1
switch(config-if-Vx1)# bfd vtep evpn prefix-list <PREFIX-LIST>
```

This command uses a supplied prefix list to filter and select the candidate VTEPs. By default, an empty prefix list will act as a deny-all and not initiate BFD sessions with any learned VTEPs.

18.6.2 Show Commands

The existing **`show interface <VTI>`** command is used to view whether BFD is enabled on the VTI and to see the timers used for any of the BFD sessions or any prefix-list configured for filtering BFD sessions.

```
switch# show interface Vxlan1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback0 and is active with 10.1.1.1
  Replication/Flood Mode is headend with Flood List Source: CLI
  Remote MAC learning is disabled
  VNI mapping to VLANs
  Static VLAN to VNI mapping is
  Dynamic VLAN to VNI mapping for 'evpn' is
    [4092, 30000]     [4093, 20000]
  Dynamic VLAN to VNI mapping for 'vccbfd' is
    [4091, 0]
  Note: All Dynamic VLANs used by VCS are internal VLANs.
        Use 'show vxlan vni' for details.
  Static VRF to VNI mapping is
   [vrf0, 20000]
  MLAG Shared Router MAC is 0000.0000.0000
  BFD is enabled with transmit interval 50, receive interval 50, multiplier 3,
 VTEP prefix list pl-example
```

The existing **`show bfd peers`** command is used to view the state of the BFD for VXLAN sessions.

```
switch# show bfd peers
VRF name: default
-----------------
DstAddr    MyDisc       YourDisc    Interface/Transport    Type        LastUp           LastDown       LastDiag
      State
-------- ----------  ------------ ---------------------- ---------  --------------  ------------
 -------------- -----------
10.1.1.2   1965370229   3607849318                 NA       VXLAN    01/12/21 10:45          NA       No
 Diagnostic       Up
10.1.1.3   1355343148   2407539267                 NA       VXLAN    01/12/21 10:45          NA       No
 Diagnostic       Up
```

The **`show ip route <VRF>`** command can be used to determine which prefixes are eligible for fast-failover.

```
switch# show ip route vrf example-vrf

VRF: example-vrf
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
```

```
       DP - Dynamic Policy Route, L - VRF Leaked,
       RC - Route Cache Route

Gateway of last resort is not set

 C        20.0.2.0/24 is directly connected, Ethernet14/1
 B E      99.99.0.0/24 [200/0] via VTEP 10.1.1.2 VNI 30000 router-mac fc:bd:67:3d:21:fd
                               via VTEP 10.1.1.3 VNI 30000 router-mac ba:ed:43:3f:ca:8e backup
```

In the above example there is a prefix with the primary path using a VXLAN tunnel to VTEP ***10.1.1.2*** and has a backup VXLAN tunnel to VTEP ***10.1.1.3***. Both paths are monitored via BFD.

In other use cases the prefix may have multiple paths with ECMP in which one or multiple of the paths are VXLAN tunnels to remote VTEPs monitored by these BFD for VXLAN sessions.

```
switch# show ip route vrf example-vrf

VRF: example-vrf
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked,
       RC - Route Cache Route

Gateway of last resort is not set

 C        20.0.2.0/24 is directly connected, Ethernet14/1
 B E      99.99.0.0/24 [200/0] via VTEP 10.1.1.2 VNI 30000 router-mac fc:bd:67:3d:24:fe
                               via VTEP 10.1.1.3 VNI 30000 router-mac ba:ed:43:3f:ca:8e
```

MLAG

This feature is applicable only to the scenario when remote prefix is known via two different MLAG VTEP pairs.

Prior to ***4.26.0F*** this feature is only supported on the primary switch of an MLAG pair due to use of Shared VTEP IP within MLAG pair as VXLAN tunnel source/destination. If BFD for VXLAN packets are received on the secondary MLAG switch, they will be forwarded to the primary MLAG switch for processing. Because only the primary MLAG switch will have BFD state for remote VTEPs, if a BFD session to a remote VTEP goes down only the primary MLAG switch will perform the fast-failover while the secondary MLAG switch will retain current behavior. Therefore it is not recommended to use this feature in conjunction with MLAG.

As of ***4.26.0F***, the primary MLAG switch will sync its BFD for VXLAN state to the secondary MLAG switch to allow the secondary to failover to an alternate path as well. To view the synced state, a new show command has been added.

```
switch# show bfd peers protocol vxlan mlag primary
Remote VTEPS for Vxlan1 on MLAG primary:
VTEP         BFD Status
----------  ----------
10.1.1.2       up
10.1.1.3       up
```

However, because this state is synced across devices, the secondary MLAG switch will not be as performant in reacting to the BFD state transitions as the primary MLAG switch which is natively responding to the BFD session.

Another exception is the multi-VTEP MLAG feature, which allows BFD for VXLAN to run on the secondary MLAG switch. When running with multi-VTEP MLAG both the primary and secondary switches will run independent BFD sessions to remote VTEPs and react to BFD state transitions separately. Each switch will use the local VTEP IP of the VTI as the source IP address for the BFD sessions, which must differ from the MLAG VTEP IP.

In summary, it is not recommended to use MLAG with this feature unless configured with the multi-VTEP IP feature referenced above.

18.6.3 Troubleshooting

- Ensure that BFD configuration is present on the relevant VTI and that the VTI status shows BFD as being enabled using the mentioned **`show interface <VTI>`** command.
- As mentioned in the prior section, BFD state transitions are Syslogged and will display if a BFD session to a remote VTEP goes down.
- Upon fast-failover to a separate path, **`show ip route`** will still display FIB state that may display the original path. To view the post failover prefix state, **`show ip hardware ale vrf <VRF> <prefix>`** can be used instead.

18.6.4 Limitations

- Support is limited to EVPN VXLAN.
- IPv6 VXLAN underlay with this feature is not supported.

18.7 VXLAN DSCP Mapping

VXLAN DSCP Mapping allows selecting Differentiated Services Code Point DSCP and Traffic Class TC values for packets at VTEPs (VXLAN Tunnel Endpoint) along VXLAN encapsulation and decapsulation directions respectively.

An incoming packet from an edge port is encapsulated with a new IP and VXLAN header before being sent out to a remote VTEP via a core facing port for encapsulation direction.When this is enabled, the DSCP field of the outgoing packet is set as the value of the DSCP of the incoming packet at the edge port.

An incoming VXLAN packet on a core port is decapsulated and forwarded out via an edge port for decapsulation direction. When this is enabled, the TC value is derived from the value of the DSCP of the incoming VXLAN encapsulated packet at the core port. This is applicable for both VXLAN Bridging and Routing.

Platform Compatibility

The following platforms support VXLAN DSCP mapping:

- DCS-7050SX3-48YC12
- DCS-7050CX3M-32S
- DCS-7050CX3-32S

Configuration

The following command enables the propagation of the DSCP field from the incoming packet to the VXLAN encapsulated packet.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan qos dscp propagation encapsulation
```

The following command enables the assignment of the TC value during decapsulation of a VXLAN packet.

```
switch(config)# interface vxlan 1
switch(config-if-Vx1)# vxlan qos dscp to traffic-class decapsulation
```

Show Commands

The **`show vxlan qos`** command shows the VXLAN DSCP and TC configuration.

```
switch# show vxlan qos
VXLAN ECN Propagation is Disabled.
Outer header DSCP is derived from Inner DSCP.
Traffic Class is derived from Outer DSCP.
```

Limitation

The core ports must be in DSCP Trust mode in order to derive TC on the outer DSCP field of the incoming VXLAN packet.

18.8 EVPN IGP Cost for VTEP Reachability

In EVPN deployment with VXLAN underlay when an EVPN type-5 prefix is imported into an IP VRF, the IGP cost of the underlay VTEP reachability is not considered as part of BGP bestpath selection post import. Therefore, if such a prefix is reachable via more than one VTEPs, the IGP metric step in the BGP best-path selection algorithm will not filter out any paths irrespective of the underlay's IGP metric for the VTEP reachability. If ECMP is enabled in the overlay and multiple paths are found to be otherwise equivalent, such paths would form ECMP regardless of the IGP metric. This is the default behavior.

However, the above mentioned behavior can be overridden by the following configuration command **`encapsulation vxlan layer-3 set next-hop igp-cost`** under ***config-route-bgp-af*** mode for **`address-family evpn`** as shown.

```
switch(config-router-bgp)# address-family evpn
switch(config-router-bgp-af)# [no | default] encapsulation vxlan layer-3 set
 next-hop igp-cost
```

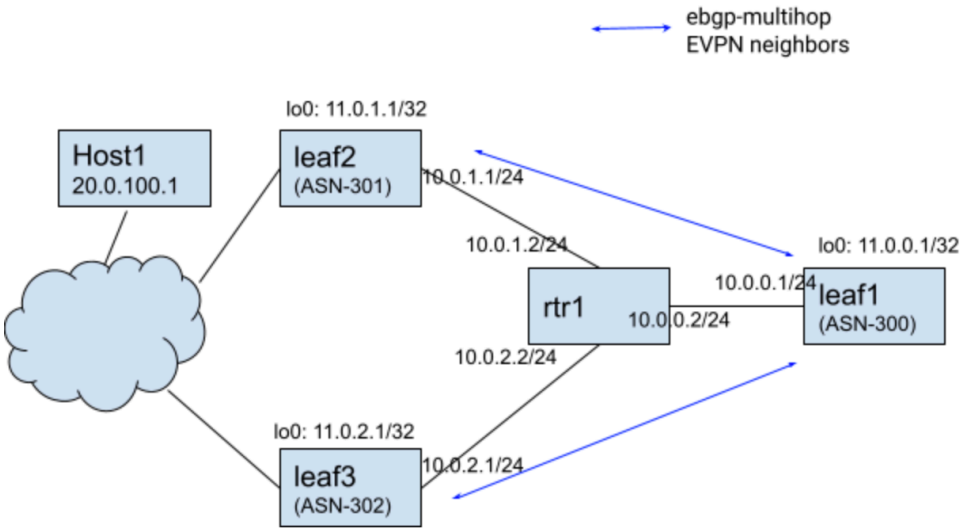
The **`encapsulation vxlan layer-3 set next-hop igp-cost`** command will cause the underlay IGP metric for the VTEP reachability to be considered for BGP best path selection in the IP VRF that is importing the EVPN route. An IGP protocol such as OSPF, ISIS or static configuration could be the source of such a metric value.

Note: This feature is available only with the multi-agent routing protocol model.

18.8.1 Configuration Example

Let us consider a topology of four routers ***leaf1***, ***leaf2***, ***leaf3*** and ***rtr1*** as shown in the figure.

Figure 18-43: EVPN IGP Cost for VTEP



There could be IGP running among the four routers, but for simplicity let's have static routes with IGP metric on ***leaf1*** to ***lo0*** of border ***leaf2*** and ***leaf3*** (and vice-versa for reverse reachability on the respective routers, though that configuration not shown here):

```
leaf1#
ip route 11.0.1.1/32 10.0.0.2 metric 340
ip route 11.0.2.1/32 10.0.0.2 metric 350
```

Following are the IP routes in the default VRF on leaf1.

```
leaf1# show ip route

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked,
       RC - Route Cache Route

Gateway of last resort:
 S        0.0.0.0/0 [1/0] via 10.0.0.2, Ethernet2

 C        10.0.0.0/24 is directly connected, Ethernet2
 C        11.0.0.1/32 is directly connected, Loopback0
 S        11.0.1.1/32 [1/340] via 10.0.0.2, Ethernet2
```

```
 S        11.0.2.1/32 [1/350] via 10.0.0.2, Ethernet2
```

Following are eBGP-multihop EVPN neighbor pairs with VXLAN as underlay:

leaf1 (ASN-300) # ***leaf2*** (ASN-301)

leaf1 (ASN-300) # ***leaf3*** (ASN-302)

Consider an example where a prefix ***20.0.100.1/32*** is reachable behind two VTEPs ***leaf2*** and ***leaf3*** as learnt on leaf1 via eBGP EVPN Type-5 routes.

Following EVPN paths will show for ***20.0.100.1/32*** on ***leaf1***:

```
leaf1(config)# show bgp evpn detail

BGP routing table entry for ip-prefix 20.0.100.1/32, Route Distinguisher:
 11.0.1.1:0
 Paths: 1 available
  301
    11.0.1.1 from 10.0.1.1 (0.0.2.1)
      Origin INCOMPLETE, metric -, localpref 100, weight 0, valid, external,
 best
      Extended Community: Route-Target-AS:64500:20000 TunnelEncap:t
unnelTypeVxlan
      EvpnRouterMac:00:00:78:03:00:00
      VNI: 20000
BGP routing table entry for ip-prefix 20.0.100.1/32, Route Distinguisher:
 11.0.2.1:0
 Paths: 1 available
  302
    11.0.2.1 from 10.0.2.1 (0.0.3.1)
      Origin INCOMPLETE, metric -, localpref 100, weight 0, valid, external,
 best
      Extended Community: Route-Target-AS:64500:20000 TunnelEncap:t
unnelTypeVxlan
      EvpnRouterMac:00:00:78:04:00:00
      VNI: 20000
```

18.8.2 Show Commands

By default, since the underlay IGP cost for the VTEP reachability is not used for best path selection in the imported EVPN Type-5 routes either of the two paths could be selected as the best path. If ECMP is configured in the importing VRF then the two paths will form ECMP. Following example shows an ECMP that is allowed in the importing VRF vrf1.

```
switch1# show ip route vrf vrf1

VRF: vrf1
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked,
       RC - Route Cache Route

 B E      20.0.100.1/32 [200/0] via VTEP 11.0.1.1 VNI 20000 router-mac
 00:00:78:02:00:00
```

```
                                    via VTEP 11.0.2.1 VNI 20000 router-mac
 00:00:78:03:00:00

switch1# show ip bgp 20.0.100.1/32 vrf vrf1
BGP routing table information for VRF vrf1
Router identifier 11.0.0.1, local AS number 300
BGP routing table entry for 20.0.100.1/32
 Paths: 2 available
  302
    11.0.2.1 from 10.0.2.1 (0.0.3.1), imported EVPN route, RD 11.0.2.1:0
      Origin INCOMPLETE, metric 0, localpref 100, IGP metric 350, weight 0, tag
 0
      Received 01:11:00 ago, valid, external, ECMP head, ECMP, best, ECMP
 contributor
      Extended Community: Route-Target-AS:64500:20000 TunnelEncap:t
unnelTypeVxlan
      EvpnRouterMac:00:00:78:04:00:00
      Remote VNI: 20000
      Rx SAFI: Unicast
  301
    11.0.1.1 from 10.0.1.1 (0.0.2.1), imported EVPN route, RD 11.0.1.1:0
      Origin INCOMPLETE, metric 0, localpref 100, IGP metric 340, weight 0, tag
 0
      Received 01:11:00 ago, valid, external, ECMP, ECMP contributor
      Not best: ECMP-Fast configured
      Extended Community: Route-Target-AS:64500:20000 TunnelEncap:t
unnelTypeVxlan
      EvpnRouterMac:00:00:78:03:00:00
      Remote VNI: 20000
      Rx SAFI: Unicast
```

The configuration command **`encapsulation vxlan layer-3 set next-hop igp-cost`** under "address-family evpn" will cause the underlay IGP cost to be taken into account and only VTEP ***11.0.1.1*** with lower IGP cost will be selected as shown below:

```
switch1# show ip route vrf vrf1

VRF: vrf1
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked,
       RC - Route Cache Route

 B E      20.0.100.1/32 [200/0] via VTEP 11.0.1.1 VNI 20000 router-mac
 00:00:78:02:00:00

switch1(config)# show ip bgp 20.0.100.1/32 vrf vrf1
BGP routing table information for VRF vrf1
Router identifier 11.0.0.1, local AS number 300
BGP routing table entry for 20.0.100.1/32
 Paths: 2 available
  301
    11.0.1.1 from 10.0.1.1 (0.0.2.1), imported EVPN route, RD 11.0.1.1:0
      Origin INCOMPLETE, metric 0, localpref 100, IGP metric 340, weight 0, tag
 0
      Received 00:23:35 ago, valid, external, best
```

```
      Extended Community: Route-Target-AS:64500:20000 TunnelEncap:t
unnelTypeVxlan
      EvpnRouterMac:00:00:78:03:00:00
      Remote VNI: 20000
      Rx SAFI: Unicast
  302
    11.0.2.1 from 10.0.2.1 (0.0.3.1), imported EVPN route, RD 11.0.2.1:0
      Origin INCOMPLETE, metric 0, localpref 100, IGP metric 350, weight 0, tag
 0
      Received 00:23:35 ago, valid, external
      Not best: IGP cost
      Extended Community: Route-Target-AS:64500:20000 TunnelEncap:t
unnelTypeVxlan
      EvpnRouterMac:00:00:78:04:00:00
      Remote VNI: 20000
      Rx SAFI: Unicast
```

18.9 EVPN VXLAN Single-Gateway Centralized Routing

In a traditional EVPN VXLAN centralized anycast gateway deployment, multiple L3 VTEPs serve the role of the centralized anycast gateway. In order for the hosts to have a consistant ARP binding for any of the individual centralized gateway VTEPs, each VTEP operating as a centralized gateway is configured with a virtual router MAC (VARP MAC), and a virtual VTEP IP (VARP VTEP IP), that is shared between all of the L3 VTEPs operating as centralized gateways. Each centralized gateway VTEP also advertises an EVPN type-3 route for both its primary VTEP IP and VARP VTEP IP, so both IPs end up in the overlay floodset.

The traditional configuration works fine, but in the specific case of a network with only a single L3 VTEP centralized gateway (or single MLAG pair operating as the L3 VTEP centralized gateway), this leads to unnecessary BUM traffic.When both the physical VTEP IP and the VARP IP end up in the overlay floodset, BUM traffic is duplicated to the centralized gateway, which can overhead workloads that have a lot of broadcast or multicast traffic. There is only a single centralized gateway, so it is not necessary to have a VARP VTEP IP to provide a stable ARP binding for the gateway.

The EVPN VXLAN Single-Gateway Centralized Routing feature enables a single L3 VTEP (or single MLAG pair) operating as an anycast gateway to not configure a VARP VTEP IP, thereby eliminating the duplicate BUM traffic caused by the VARP VTEP IP being in the overlay floodset. It accomplishes this with two changes:

- A change to the default ARP behavior when a host ARPs for the MAC address associated with a virtual IP address (SVI IP).

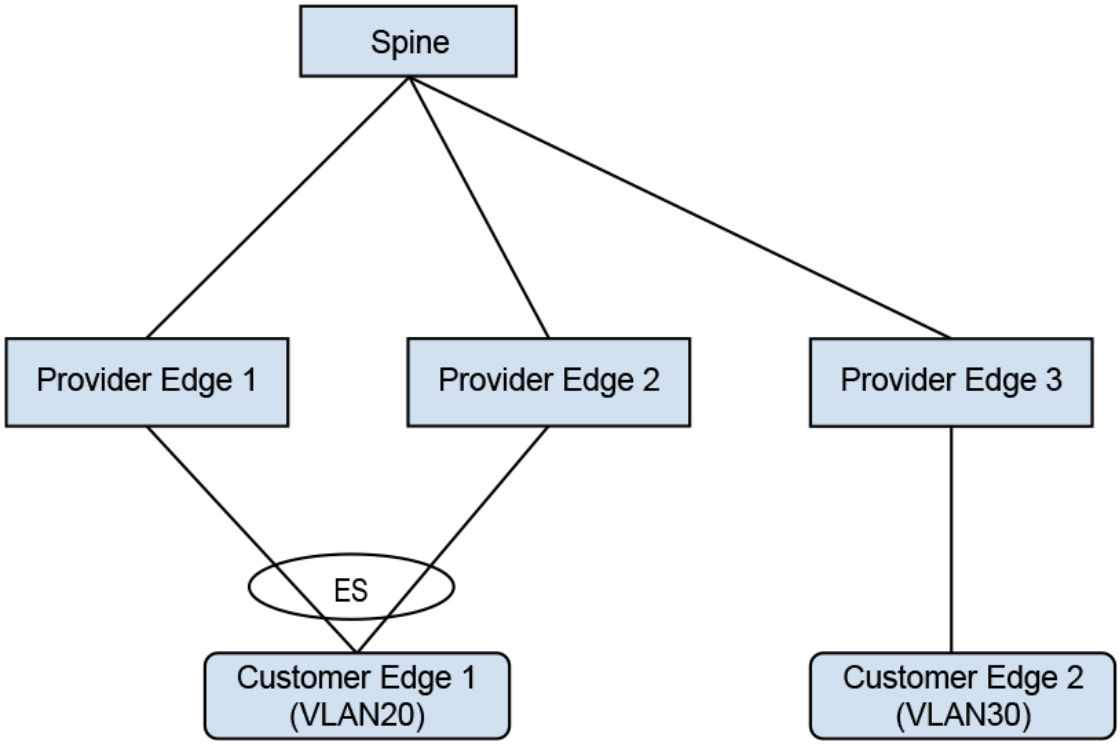
  Previously, ARPing for a virtual IP returned different MAC address bindings, depending on whether or not a VARP VTEP IP was configured. If a VARP VTEP IP was configured, the ARP request returns the configured VARP MAC. If one was not, the ARP request returns the switch router MAC. This feature changes the behavior to always respond with the configured VARP MAC to an ARP request for a virtual IP. This closes an exception to the rule that virtual IPs are always associated with the VARP MAC.
- A new EVPN MAC VRF configuration command that generates an EVPN type-2 route for the VARP MAC with a nexthop of the physical VTEP IP.

In a traditional EVPN centralized anycast gateway development, the presence of a configured VARP VTEP IP advertises an EVPN type-2 route for the VARP MAC with a nexthop of the VARP VTEP IP. This allows TOR switches to learn the ARP binding of the centralized anycast gateway (presumably, their default gateway). With this feature, no VARP VTEP IP is configured, so an alternative method is required to advertise the appropriate EVPN route. This feature adds a new EVPN MAC VRF configuration command, **`redistribute router-mac next-hop vtep primary`** ,which when configured on a MAC VRF advertises an EVPN type-2 route for the VARP MAC with a nexthop of the primary VTEP IP. This allows TOR switches to learn the ARP binding of the centralized anycast gateway the same way they would in a traditional centralized anycast gateway deployment.

18.9.1 Configuration

On the multi-homing PEs, you must configure the Ethernet Segment (ES) to the CE. In addition, the configuration needed for asymmetric IRB or symmetric IRB must be configured on the local and the remote PEs.

Figure 18-44: Asymmetric IRB with IPv4



In the example, ***CE1*** is a multi-homed CE in ***VLAN20***. ***CE2*** is a remote CE in ***VLAN30***. Asymmetric IRB is configured for inter-VLAN traffic.

Configuration on ***PE1***:

```
switch(config)# interface Port-Channel100
switch(config-if-Po100)# switchport access vlan 20
!
switch(config-if-Po100)# evpn ethernet-segment
switch(config-evpn-es)# identifier 0033:3333:3333:3333:3333
switch(config-evpn-es)# route-target import 00:03:00:03:00:03
switch(config-evpn-es)# lacp system-id 1234.5678.0123
!
switch(config)# interface Ethernet1
switch(config-if-Et1)# switchport mode trunk
switch(config-if-Et1)# channel-group 100 mode on
!
switch(config)# interface Loopback0
switch(config-if-Lo0)# ip address 10.255.0.0/32
!
switch(config)# interface Vlan20
switch(config-if-Vl20)# ip address virtual 20.0.20.1/24
!
switch(config)# interface Vlan30
switch(config-if-Vl30)# ip address virtual 20.0.30.1/24
!
switch(config)# interface Vxlan1
```

```
switch(config=if-Vx1)# vxlan source-interface Loopback0
switch(config=if-Vx1)# vxlan udp-port 4789
switch(config=if-Vx1)# vxlan vlan 20 vni 10020
switch(config=if-Vx1)# vxlan vlan 30 vni 10030
!
switch(config)# ip virtual-router mac-address 00:00:80:00:00:00
!
switch(config)# router bgp 300
switch(config-router-bgp)# router-id 0.0.0.1
switch(config-router-bgp)# neighbor 10.0.0.1 remote-as 303
switch(config-router-bgp)# neighbor 10.0.0.1 ebgp-multihop
switch(config-router-bgp)# neighbor 10.0.0.1 send-community extended
switch(config-router-bgp)# neighbor 10.0.0.1 maximum-routes 12000
switch(config-router-bgp)# redistribute static
   !
switch(config-router-bgp)# vlan 20
switch(config-macvrf-20)# rd 10.255.0.0:20
switch(config-macvrf-20)# route-target both 64500:10020
switch(config-macvrf-20)# redistribute learned
   !
switch(config-router-bgp)# vlan 30
switch(config-macvrf-30)# rd 10.255.0.0:30
switch(config-macvrf-30)# route-target both 64500:10030
switch(config-macvrf-30)# redistribute learned
   !
switch(config-macvrf-30)# address-family evpn
switch(config-router-bgp-af)# neighbor 10.0.0.1 activate
```

The Ethernet segment to the multi-homed CE is configured on the port channel interface Port-Channel ***100***. ***SVI 20*** and ***SVI 30*** along with VARP IP are configured for inter-subnet routing. A VARP MAC is configured globally on ***PE1***. The configuration on ***PE2*** is similar to the configuration shown above. On ***PE3***, ***SVI 20*** and ***SVI 30*** are configured along with VARP IP and VARP MAC.

Symmetric IRB with IPv4 example:

Configuration on ***PE1***:

```
!
switch(config)# vrf instance red
switch(config-vrf-red)# rd 10.255.0.0:0
!
switch(config-vrf-red)# interface Port-Channel100
switch(config-if-Po100)# switchport access vlan 20
!
switch(config-if-Po100)# evpn ethernet-segment
switch(config-evpn-es)# identifier 0033:3333:3333:3333:3333
switch(config-evpn-es)# route-target import 00:03:00:03:00:03
switch(config-evpn-es)# lacp system-id 1234.5678.0123
!
switch(config-if-Po100)# interface Ethernet6/6/1
switch(config-if-Et6/6/1)# switchport mode trunk
switch(config-if-Et6/6/1)# channel-group 100 mode on
!
switch(config-if-Et6/6/1)# interface Loopback0
switch(config-if-Lo0)# ip address 10.255.0.0/32
!
switch(config-if-Lo0)# interface Vlan20
switch(config-if-Vl20)# vrf red
switch(config-if-Vl20)# ip address virtual 20.0.20.1/24
!
switch(config-if-Vl20)# interface Vxlan1
switch(config-if-Vx1)# vxlan source-interface Loopback0
```

```
switch(config-if-Vx1)# vxlan udp-port 4789
switch(config-if-Vx1)# vxlan vlan 10 vni 10010
switch(config-if-Vx1)# vxlan vlan 20 vni 10020
switch(config-if-Vx1)# vxlan vrf red vni 20000
!
switch(config-if-Vx1)# ip virtual-router mac-address 00:00:80:00:00:00
!
switch(config)# router bgp 300
switch(config-router-bgp)# router-id 0.0.0.1
switch(config-router-bgp)# maximum-paths 2
switch(config-router-bgp)# neighbor 10.0.0.1 remote-as 303
switch(config-router-bgp)# neighbor 10.0.0.1 ebgp-multihop
switch(config-router-bgp)# neighbor 10.0.0.1 send-community extended
switch(config-router-bgp)# neighbor 10.0.0.1 maximum-routes 12000
switch(config-router-bgp)# redistribute static
!
switch(config-router-bgp)# vlan 20
switch(config-macvrf-20)# rd 10.255.0.0:20
switch(config-macvrf-20)# route-target both 64500:10020
switch(config-macvrf-20)# redistribute learned
!
switch(config-macvrf-20)# address-family evpn
switch(config-router-bgp-af)# neighbor 10.0.0.1 activate
!
switch(config-router-bgp-af)# vrf red
switch(config-router-bgp-vrf-red)# rd 10.255.0.0:0
switch(config-router-bgp-vrf-red)# route-target import evpn 64500:20000
switch(config-router-bgp-vrf-red)# route-target export evpn 64500:20000
switch(config-router-bgp-vrf-red)# router-id 10.255.0.0
!
```

The Ethernet segment to the multi-homed ***CE1*** is configured on the port channel interface Port-Channel ***100***. ***SVI 20*** along with VARP IP and VARP MAC is configured. Also, IP VRF is configured which is needed for symmetric IRB. The configuration on ***PE2*** is similar. On ***PE3***, IP VRF and ***SVI 30*** are configured for symmetric IRB.

VXLAN example

A netwok with 2 VRFs, ***red*** and ***blue*** has VLANs ***10*** and ***20*** in ***red*** and VLANs ***30*** and ***40*** in ***blue***. The spines in these act as a route reflectors. Multicast groups are used to encapsulate traffic arriving in a VRF such that it is delivered to VTEPs that have that VRF provisioned.

```
interface Loopback0
   ip address 10.0.0.20/32
!
vlan 10
vlan 20
vlan 30
vlan 40
!
interface Ethernet1
   switchport access vlan 10
!
interface Ethernet2
   switchport access vlan 20
!
interface Ethernet3
   switchport access vlan 30
!
interface Ethernet4
   switchport access vlan 40
!
```

```
interface Vlan10
   vrf red
   ip address virtual 192.168.1.0/24
   ip igmp
   pim ipv4 local-interface loopback0
!
interface Vlan20
   vrf red
   ip address 192.168.2.0/24
   ip igmp
!
interface Vlan30
   vrf blue
   ip address 192.168.1.0/24
   ip igmp
!
interface Vlan40
   vrf blue
   ip address 192.168.2.0/24
   ip igmp
!
interface Vxlan1
   vxlan source-interface Loopback0
   vxlan vlan 10 vni 10
   vxlan vlan 20 vni 20
   vxlan vlan 30 vni 30
   vxlan vlan 40 vni 40
   vxlan vrf red vni 100
   vxlan vrf blue vni 200
   vxlan vlan 10 flood group 225.1.1.2
   vxlan vlan 20 flood group 225.1.1.3
   vxlan vlan 30 flood group 226.1.1.2
   vxlan vlan 40 flood group 226.1.1.3
   vxlan vrf red multicast group 225.1.1.1
   vxlan vrf blue multicast group 226.1.1.1
!
```

18.9.2 Show Commands

The following examples are based on the sample topology and configuration in the previous sections.

On the remote VTEP, to display the EVPN routes to the multi-homed CE (***20.0.20.2***):

```
switch# show bgp evpn route-type mac-ip 20.0.20.2
BGP routing table information for VRF default
Router identifier 0.0.3.1, local AS number 302
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head,
                    e - ECMP S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

          Network                Next Hop              Metric  LocPref Weight  Path
 * >     RD: 10.255.0.0:20 mac-ip 0000.0101.0000 20.0.20.2
                                 10.255.0.0            -       100     0       303 300 i
 * >     RD: 10.255.0.1:20 mac-ip 0000.0101.0000 20.0.20.2
                                 10.255.0.1            -       100     0       303 301 i
```

As shown above, there are two EVPN MAC-IP routes for the multi-homed CE.

On the remote PE, to display the installed routes to the multi-homed CE:

```
switch# show ip route vrf red 20.0.20.2/32

VRF: red
Codes: C - connected, S - static, K - kernel,
```

```
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

 B E      20.0.20.2/32 [200/0] via VTEP 10.255.0.0 VNI 20000 router-mac 00:00:78:01:00:00
                               via VTEP 10.255.0.1 VNI 20000 router-mac 00:00:78:04:00:00
```

Two routes to the multi-homed CE are installed from the two EVPN MAC-IP routes and they form L3 ECMP.

On the remote PE, to check the details of the two routes in BGP RIB:

```
switch# show ip bgp 20.0.20.2/32 vrf red
BGP routing table information for VRF red
Router identifier 10.255.0.2, local AS number 302
BGP routing table entry for 20.0.20.2/32
 Paths: 2 available
  303 300
    10.255.0.0 from 10.0.2.1 (0.0.1.1), imported EVPN route, RD 10.255.0.0:20
      Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, received 00:14:43 ago,
      valid, external, ECMP head, ECMP, best, ECMP contributor
      Extended Community: Route-Target-AS:64500:10020 Route-Target-AS:64500:20000
      TunnelEncap:tunnelTypeVxlan EvpnMacMobility:1 EvpnRouterMac:00:00:78:01:00:00
      Remote VNI: 20000
      Rx SAFI: Unicast
  303 301
    10.255.0.1 from 10.0.2.1 (0.0.1.1), imported EVPN route, RD 10.255.0.1:20
      Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, received 00:14:43 ago,
      valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:64500:10020 Route-Target-AS:64500:20000
      TunnelEncap:tunnelTypeVxlan EvpnMacMobility:1 EvpnRouterMac:00:00:78:04:00:00
      EvpnNdFlags:pflag
      Remote VNI: 20000
      Rx SAFI: Unicast
```

The second route has **EvpnNdFlags:pflag** to indicate that this is a proxy MAC-IP route.

This command shows information about the SBD instance that is created when **evpn multicast** is configured under an IP VRF:

```
switch# show bgp evpn instance sbd red
EVPN instance: SBD red
  Route distinguisher: 100:1
  Service interface: VLAN-based
  Local IP address: 10.0.0.20
  Encapsulation type: VXLAN
vtep2#show bgp evpn instance sbd blue
EVPN instance: SBD red
  Route distinguisher: 200:1
  Service interface: VLAN-based
  Local IP address: 10.0.0.20
  Encapsulation type: VXLAN
```

The OISM supported capability is set in IMET routes for the SBD when **evpn multicast** is configured for a VRF. The IGMP proxy flag is not set in IMET routes for VLANs in the VRF unless**redistribute igmp** is configured in a VLAN:

```
switch# show bgp evpn route-type imet next-hop 10.0.0.10 detail
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
BGP routing table entry for imet 10.0.0.10, Route Distinguisher: 10:1
 Paths: 1 available
  Local
    10.0.0.10 from 10.0.0.1 (0.0.1.1)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:10:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10
      PMSI Tunnel: PIM-SSM Tree, MPLS Label: 10, Leaf Information Required: false,
      Tunnel ID: 10.0.0.10, 225.1.1.2
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
```

```
BGP routing table entry for imet 10.0.0.10, Route Distinguisher: 100:1
 Paths: 1 available
  Local
    10.0.0.10 from 10.0.0.1 (0.0.1.1)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:100:1 TunnelEncap:tunnelTypeVxlan Multicast
      Flags: IGMP proxy, OISM-supported, SBD
      VNI: 100
      PMSI Tunnel: Ingress Replication, MPLS Label: 100, Leaf Information Required: false,
      Tunnel ID: 10.0.0.10
```

This command shows information about the single SMET route for the group with the RD of ***VRF red***:

```
switch# show bgp evpn route-type smet multicast 228.1.1.1
BGP routing table information for VRF default
Router identifier 0.0.1.1, local AS number 300
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop
          Network                Next Hop              Metric  LocPref Weight  Path
 * >     RD: 100:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.10
                                 10.0.0.10             -       100     0       i
 * >     RD: 100:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD: 100:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD: 200:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD: 200:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.40
                                 10.0.0.40             -       100     0       i
```

This command shows information about the SPMSI route for the group:

```
switch# show bgp evpn route-type spmsi
BGP routing table information for VRF defaultRouter identifier 0.0.0.1, local AS number 300
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop
          Network                Next Hop              Metric  LocPref Weight  Path
 * >     RD:  10:1 spmsi (S, G): (*, *) originating IP: 10.0.0.10
                                 10.0.0.10             -       100     0       i
 * >     RD:  10:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD:  20:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD:  30:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD:  40:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD:  10:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD:  20:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD:  30:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD:  40:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD:  30:1 spmsi (S, G): (*, *) originating IP: 10.0.0.40
                                 10.0.0.40             -       100     0       i
 * >     RD: 100:1 spmsi (S, G): (*, *) originating IP: 10.0.0.10
                                 10.0.0.10             -       100     0       i
 * >     RD: 100:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD: 200:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD: 100:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD: 200:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD: 200:1 spmsi (S, G): (*, *) originating IP: 10.0.0.40
                                 10.0.0.40             -       100     0       i
```

18.9.3 Limitations

The following limitations are included in the EVPN VXLAN all-active multi-homing integrated routing and bridging feature:

- L2 loop-free protocols, such as Spanning Tree Protocol (STP) are not supported between PE-CE with EVPN VXLAN All-Active Multi-homing. Users must ensure their topology is loop-free. STP must be disabled for the Multihomed VLANs on PEs and CEs.
- A limit of two multi-homing destinations are installed at one time for any particular MAC address. Any other valid destinations are ignored in the context of switching unicast traffic until one of the active destinations is removed. The multi-homing PEs themselves operate normally regardless of the number of PEs on the ethernet segment; this limitation only affects the selection of a destination for unicast traffic.
- VXLAN GPE is not currently supported, so packets cannot be flagged as having been broadcast. As a result, unicast packets that are known at the sender and unknown at the receiver are dropped if the receiving PE is not a designated forwarder. Similarly, unicast packets that are unknown at the sender and known at the receiver are duplicated at the receiving CE. This state automatically resolves itself as the BGP network distributes the appropriate type 1 auto-discovery and type 2 MAC/IP advertisement EVPN routes.
- Fast mass withdrawal is not supported, so there may be a delay if an interface harboring a large number of MAC addresses goes down.

18.9.4 Flood Traffic Filtering with EVPN

VXLAN fabric managed by EVPN does not always flood with broadcast, multicast or unknown MAC traffic. Sometimes ARP request broadcast and ND multicast traffic flood the VXLAN fabric as well. There may be other cases where flooding ARP plus other traffic is allowed but not all broadcast traffic into the fabric.

Most of the ARPs are learnt through EVPN and for some cases where the ARP is not learnt through EVPN, it is acceptable to flood such ARP requests. However, it is requested to rate limit the ARP flooding going into the VXLAN fabric.

Configuration

The default command disables flooding of different kinds of traffic into the VXLAN fabric in order to restrict only ARP and ND traffic flooding.

```
switch(config-rtr-l2-vpn)# flooding default disabled
```

The following command enables flooding of ARP packets into the VXLAN fabric again.

```
switch(config-rtr-l2-vpn)# arp flooding
```

The above command starts flooding ARP traffic into the VXLAN fabric while all other traffic including ND traffic is prevented from getting flooded into the fabric. The following command enables flooding of ND traffic.

```
switch(config-rtr-l2-vpn)# nd flooding
```



Note: These commands are in effect only when EVPN is enabled. ARP & ND flooding are enabled by default.

Show Commands

The **show vxlan counters software** command shows the number of ARP & ND packets prevented from getting flooded into the VXLAN fabric.

```
switch# show vxlan counters software
. . . . . .
Tx pkts after IPv6 encapsulation                    :  0
```

```
SW pkts forwarded to remote VTEPs via HW HER         :  4
SW pkts forwarding to remote VTEPs via HW HER failed :  0
Packets suppressed from getting flooded              :  0
```

18.10 Inter-VRF Local Route Leaking

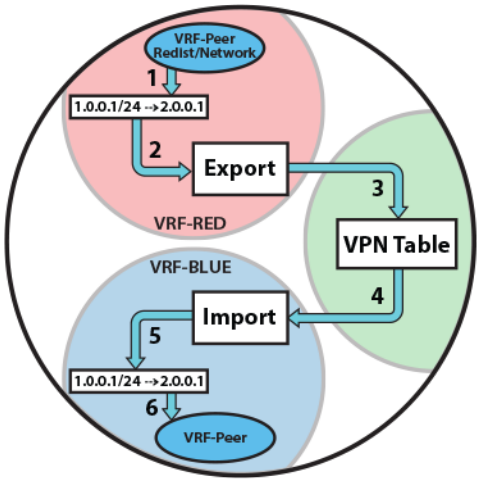
Inter-VRF local route leaking allows the leaking of routes from one VRF (the source VRF) to another VRF (the destination VRF) on the same router. Inter-VRF routes can exist in any VRF (including the default VRF) on the system. Routes can be leaked using the following methods:

- Inter-VRF Local Route Leaking using BGP VPN
- Inter-VRF Local Route Leaking using VRF-leak Agent

18.10.1 Inter-VRF Local Route Leaking using BGP VPN

Inter-VRF local route leaking allows the user to export and import routes from one VRF to another on the same device. This is implemented by exporting routes from a VRF to the local VPN table using route target extended community list and then importing the same route target extended community lists from the local VPN table into the target VRF. VRF route leaking is supported on VPN-IPv4, VPN-IPv6, and EVPN types.

Figure 18-45: Inter-VRF Local Route Leaking using Local VPN Table

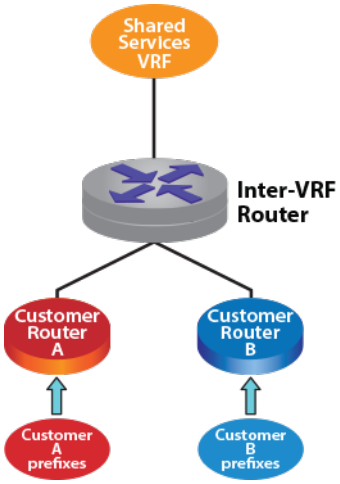


Accessing Shared Resources Across VPNs

To access shared resources across VPNs, all the routes from the shared services VRF must be leaked into each of the VPN VRFs and customer routes must be leaked into the shared services VRF for return traffic. Accessing shared resources allows one to export the route target of the shared services VRF into all customer VRFs, and allows the shared services VRF to import route targets from customers A and B. The figure below shows how to provide customers, corresponding to multiple VPN domains, access to services like DHCP available in the shared VRF.

Route leaking across the VRFs is supported on VPN-IPv4, VPN-IPv6, and EVPN.

Figure 18-46: Accessing Shared Resources Across VPNs



18.10.1.1 Configuring Inter-VRF Local Route Leaking

Inter-VRF local route leaking is configured using VPN-IPv4, VPN-IPv6, and EVPN. Prefixes can be exported and imported using any of the configured VPN types. Ensure that the same VPN type that is exported is used while importing.

Leaking unicast IPv4 or IPv6 prefixes is supported and achieved by exporting prefixes locally to the VPN table and importing locally from the VPN table into the target VRF on the same device as shown in the figure titled **Inter-VRF Local Route Leaking using Local VPN Table** using the **`route-target`** command.

Exporting or importing the routes to or from the EVPN table is accomplished with the following two methods:

- Using VXLAN for encapsulation
- Using MPLS for encapsulation

Using VXLAN for Encapsulation

To use VXLAN encapsulation type, ensure that VRF to VNI mapping is present and the interface status for the VXLAN interface is up. This is the default encapsulation type for EVPN.

Example

The configuration for VXLAN encapsulation type is as follows:

```
switch(config)# router bgp 65001
switch(config-router-bgp)# address-family evpn
switch(config-router-bgp-af)# neighbor default encapsulation vxlan next-hop-
self source-interface Loopback0
switch(config)# hardware tcam
switch(config-hw-tcam)# system profile vxlan-routing
switch(config-hw-tcam)# interface Vxlan1
switch(config-hw-tcam-if-Vx1)# vxlan source-interface Loopback0
switch(config-hw-tcam-if-Vx1)# vxlan udp-port 4789
switch(config-hw-tcam-if-Vx1)# vxlan vrf vrf-blue vni 20001
switch(config-hw-tcam-if-Vx1)# vxlan vrf vrf-red vni 10001
```

Using MPLS for Encapsulation

To use MPLS encapsulation type to export to the EVPN table, MPLS needs to be enabled globally on the device and the encapsulation method needs to be changed from default type, that is VXLAN to MPLS under the EVPN address-family sub-mode.

Example

```
switch(config)# router bgp 65001
switch(config-router-bgp)# address-family evpn
switch(config-router-bgp-af)# neighbor default encapsulation mpls next-hop-self
 source-interface Loopback0
```

18.10.1.2 Route-Distinguisher

Route-Distinguisher (RD) is used to uniquely identify routes from a particular VRF. Route distinguisher is configured for every VRF from which routes are exported from or imported into.

The following commands are used to configure route distinguisher for a VRF.

```
Switch(config-router-bgp)# vrf vrf-services
switch(config-router-bgp-vrf-vrf-services)# rd 1.0.0.1:1

switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 2.0.0.1:2
```

18.10.1.3 Exporting Routes from a VRF

Use the **route-target export** command to export routes from a VRF to the local VPN or EVPN table using the route target extended community list.

Examples

- These commands export routes from ***vrf-red*** to the local VPN table.

```
switch(config)# service routing protocols model multi-agent
switch(config)# mpls ip
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-red
switch(config-router-bgp-vrf-vrf-red)# rd 1:1
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv4 10:10
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv6 10:20
```

- These commands export routes from ***vrf-red*** to the EVPN table.

```
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-red
switch(config-router-bgp-vrf-vrf-red)# rd 1:1
switch(config-router-bgp-vrf-vrf-red)# route-target export evpn 10:1
```

18.10.1.4 Importing Routes into a VRF

Use the **route-target import** command to import the exported routes from the local VPN or EVPN table to the target VRF using the route target extended community list.

Examples

- These commands import routes from the VPN table to ***vrf-blue***.

```
switch(config)# service routing protocols model multi-agent
switch(config)# mpls ip
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 2:2
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv4 10:10
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv6 10:20
```

- These commands import routes from the EVPN table to ***vrf-blue***.

```
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 2:2
switch(config-router-bgp-vrf-vrf-blue)# route-target import evpn 10:1
```

18.10.1.5 Exporting and Importing Routes using Route Map

To manage VRF route leaking, control the prefixes that are exported and imported with route-map export or import commands. The route map is effective only if the VRF paths or the VPN paths are already candidates for export or import. It is mandatory to have the route-target export or import command configured first. Setting BGP attributes using route maps is effective only on the export end.

Note: Prefixes that are leaked are not re-exported to the VPN table from the target VRF.

Examples

- These commands export routes from ***vrf-red*** to the local VPN table.

```
switch(config)# service routing protocols model multi-agent
switch(config)# mpls ip
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-red
switch(config-router-bgp-vrf-vrf-red)# rd 1:1
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv4 10:10
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv6 10:20
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv4 route-
map EXPORT_V4_ROUTES_T0_VPN_TABLE
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv6 route-
map EXPORT_V6_ROUTES_T0_VPN_TABLE
```

- These commands export routes to from ***vrf-red*** to the EVPN table.

```
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-red
switch(config-router-bgp-vrf-vrf-red)# rd 1:1
switch(config-router-bgp-vrf-vrf-red)# route-target export evpn 10:1
switch(config-router-bgp-vrf-vrf-red)# route-target export evpn route-map
 EXPORT_ROUTES_T0_EVPN_TABLE
```

- These commands import routes from the VPN table to ***vrf-blue***.

```
switch(config)# service routing protocols model multi-agent
switch(config)# mpls ip
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 1:1
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv4 10:10
```

```
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv6 10:20
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv4 route-
map IMPORT_V4_ROUTES_VPN_TABLE
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv6 route-
map IMPORT_V6_ROUTES_VPN_TABLE
```

- These commands import routes from the EVPN table to ***vrf-blue***.

```
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 2:2
switch(config-router-bgp-vrf-vrf-blue)# route-target import evpn 10:1
switch(config-router-bgp-vrf-vrf-blue)# route-target import evpn route-map
 IMPORT_ROUTES_FROM_EVPN_TABLE
```

18.10.2 Inter-VRF Local Route Leaking using VRF-leak Agent

Inter-VRF local route leaking allows the leaking of routes from one VRF to another using route map as a VRF-leak agent. VRFs are leaked based on the preferences assigned to each VRF.

18.10.2.1 Configuring Route Maps

To leak routes from one VRF to another using a route map, use the **`router general`** command to enter Router-General Configuration Mode, then enter the VRF submode for the destination VRF, and use the **`leak routes`** command to specify the source VRF and the route map to be used. Routes in the source VRF that match the policy in the route map will then be considered for leaking into the configuration-mode VRF. If two or more policies specify leaking the same prefix to the same destination VRF, then the route with a higher (post-set-clause) distance and preference is chosen.

Example

These commands configure a route map to leak routes from ***VRF1*** to ***VRF2*** using route map ***RM1***.

```
switch(config)# router general
switch(config-router-general)# vrf VRF2
switch(config-router-general-vrf-VRF2)# leak routes source-vrf VRF1 subscribe-
policy RM1
switch(config-router-general-vrf-VRF2)#
```

18.11 Static Inter-VRF Route

The Static Inter-VRF Route feature adds support for static inter-VRF routes. This enables the configuration of routes to destinations in one ingress VRF with an ability to specify a next-hop in a different egress VRF through a static configuration.

You can configure static inter-VRF routes in default and non-default VRFs. A different egress VRF is achieved by “tagging” the **next-hop** or **forwarding via** with a reference to an egress VRF (different from the source VRF) in which that next-hop should be evaluated. Static inter-VRF routes with ECMP next-hop sets in the same egress VRF or heterogenous egress VRFs can be specified.

The Static Inter-VRF Route feature is independent and complementary to other mechanisms that can be used to setup local inter-VRF routes. The other supported mechanisms in EOS and the broader use-cases they support are documented here:

- Local Route leaking using BGP VPN
- Local Route leaking using EOS RouteLeak Agent

18.11.1 Configuration

The configuration to setup static-Inter VRF routes in an ingress (source) VRF to forward IP traffic to a different egress (target) VRF can be done in the following modes:

- This command creates a static route in one ingress VRF that points to a next-hop in a different egress VRF.

  ip|ipv6 route [vrf ***vrf-name destination-prefix*** [egress-vrf***egress-next-hop vrf-name***] ***next-hop***

18.11.2 Show Commands

Use the **show ip route vrf** to display the egress VRF name if it is different from the source VRF.

Example

```
switch# show ip route vrf vrf1

VRF: vrf1
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

Gateway of last resort is not set

 S        1.0.1.0/24 [1/0] via 1.0.0.2, Vlan2180 (egress VRF default)
 S        1.0.7.0/24 [1/0] via 1.0.6.2, Vlan2507 (egress VRF vrf3)
```

18.11.3 Limitations

- For bidirectional traffic to work correctly between a pair of VRFs, static inter-VRF routes in both VRFs must be configured.
- Static Inter-VRF routing is supported in the multi-agent routing protocol mode only.

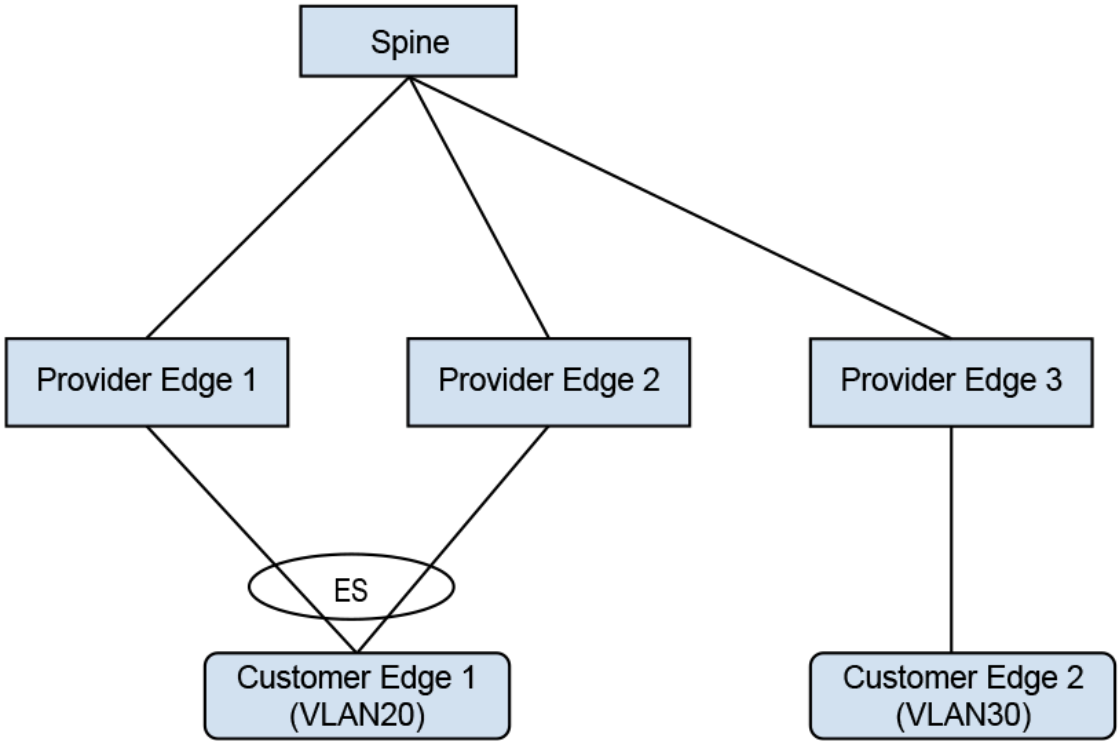
18.12 VCS to EVPN Hitless Migration

VCS to EVPN Hitless Migration enables support for migrating from only using VCS as the control plane to only using EVPN as a control plane in a hitless manner with respect to L2 reachability information.

18.12.1 Configuration

On the multi-homing PEs, you must configure the Ethernet Segment (ES) to the CE. In addition, the configuration needed for asymmetric IRB or symmetric IRB must be configured on the local and the remote PEs.

Figure 18-47: Asymmetric IRB with IPv4



In the example, ***CE1*** is a multi-homed CE in ***VLAN20***. ***CE2*** is a remote CE in ***VLAN30***. Asymmetric IRB is configured for inter-VLAN traffic.

Configuration on ***PE1***:

```
switch(config)# interface Port-Channel100
switch(config-if-Po100)# switchport access vlan 20
!
switch(config-if-Po100)# evpn ethernet-segment
switch(config-evpn-es)# identifier 0033:3333:3333:3333:3333
switch(config-evpn-es)# route-target import 00:03:00:03:00:03
switch(config-evpn-es)# lacp system-id 1234.5678.0123
!
switch(config)# interface Ethernet1
switch(config-if-Et1)# switchport mode trunk
switch(config-if-Et1)# channel-group 100 mode on
!
switch(config)# interface Loopback0
switch(config-if-Lo0)# ip address 10.255.0.0/32
!
switch(config)# interface Vlan20
switch(config-if-Vl20)# ip address virtual 20.0.20.1/24
!
switch(config)# interface Vlan30
switch(config-if-Vl30)# ip address virtual 20.0.30.1/24
!
switch(config)# interface Vxlan1
```

```
switch(config=if-Vx1)# vxlan source-interface Loopback0
switch(config=if-Vx1)# vxlan udp-port 4789
switch(config=if-Vx1)# vxlan vlan 20 vni 10020
switch(config=if-Vx1)# vxlan vlan 30 vni 10030
!
switch(config)# ip virtual-router mac-address 00:00:80:00:00:00
!
switch(config)# router bgp 300
switch(config-router-bgp)# router-id 0.0.0.1
switch(config-router-bgp)# neighbor 10.0.0.1 remote-as 303
switch(config-router-bgp)# neighbor 10.0.0.1 ebgp-multihop
switch(config-router-bgp)# neighbor 10.0.0.1 send-community extended
switch(config-router-bgp)# neighbor 10.0.0.1 maximum-routes 12000
switch(config-router-bgp)# redistribute static
   !
switch(config-router-bgp)# vlan 20
switch(config-macvrf-20)# rd 10.255.0.0:20
switch(config-macvrf-20)# route-target both 64500:10020
switch(config-macvrf-20)# redistribute learned
   !
switch(config-router-bgp)# vlan 30
switch(config-macvrf-30)# rd 10.255.0.0:30
switch(config-macvrf-30)# route-target both 64500:10030
switch(config-macvrf-30)# redistribute learned
   !
switch(config-macvrf-30)# address-family evpn
switch(config-router-bgp-af)# neighbor 10.0.0.1 activate
```

The Ethernet segment to the multi-homed CE is configured on the port channel interface Port-Channel ***100***. ***SVI 20*** and ***SVI 30*** along with VARP IP are configured for inter-subnet routing. A VARP MAC is configured globally on ***PE1***. The configuration on ***PE2*** is similar to the configuration shown above. On ***PE3***, ***SVI 20*** and ***SVI 30*** are configured along with VARP IP and VARP MAC.

Symmetric IRB with IPv4 example:

Configuration on ***PE1***:

```
!
switch(config)# vrf instance red
switch(config-vrf-red)# rd 10.255.0.0:0
!
switch(config-vrf-red)# interface Port-Channel100
switch(config-if-Po100)# switchport access vlan 20
!
switch(config-if-Po100)# evpn ethernet-segment
switch(config-evpn-es)# identifier 0033:3333:3333:3333:3333
switch(config-evpn-es)# route-target import 00:03:00:03:00:03
switch(config-evpn-es)# lacp system-id 1234.5678.0123
!
switch(config-if-Po100)# interface Ethernet6/6/1
switch(config-if-Et6/6/1)# switchport mode trunk
switch(config-if-Et6/6/1)# channel-group 100 mode on
!
switch(config-if-Et6/6/1)# interface Loopback0
switch(config-if-Lo0)# ip address 10.255.0.0/32
!
switch(config-if-Lo0)# interface Vlan20
switch(config-if-Vl20)# vrf red
switch(config-if-Vl20)# ip address virtual 20.0.20.1/24
!
switch(config-if-Vl20)# interface Vxlan1
switch(config-if-Vx1)# vxlan source-interface Loopback0
```

```
switch(config-if-Vx1)# vxlan udp-port 4789
switch(config-if-Vx1)# vxlan vlan 10 vni 10010
switch(config-if-Vx1)# vxlan vlan 20 vni 10020
switch(config-if-Vx1)# vxlan vrf red vni 20000
!
switch(config-if-Vx1)# ip virtual-router mac-address 00:00:80:00:00:00
!
switch(config)# router bgp 300
switch(config-router-bgp)# router-id 0.0.0.1
switch(config-router-bgp)# maximum-paths 2
switch(config-router-bgp)# neighbor 10.0.0.1 remote-as 303
switch(config-router-bgp)# neighbor 10.0.0.1 ebgp-multihop
switch(config-router-bgp)# neighbor 10.0.0.1 send-community extended
switch(config-router-bgp)# neighbor 10.0.0.1 maximum-routes 12000
switch(config-router-bgp)# redistribute static
!
switch(config-router-bgp)# vlan 20
switch(config-macvrf-20)# rd 10.255.0.0:20
switch(config-macvrf-20)# route-target both 64500:10020
switch(config-macvrf-20)# redistribute learned
!
switch(config-macvrf-20)# address-family evpn
switch(config-router-bgp-af)# neighbor 10.0.0.1 activate
!
switch(config-router-bgp-af)# vrf red
switch(config-router-bgp-vrf-red)# rd 10.255.0.0:0
switch(config-router-bgp-vrf-red)# route-target import evpn 64500:20000
switch(config-router-bgp-vrf-red)# route-target export evpn 64500:20000
switch(config-router-bgp-vrf-red)# router-id 10.255.0.0
!
```

The Ethernet segment to the multi-homed ***CE1*** is configured on the port channel interface Port-Channel ***100***. ***SVI 20*** along with VARP IP and VARP MAC is configured. Also, IP VRF is configured which is needed for symmetric IRB. The configuration on ***PE2*** is similar. On ***PE3***, IP VRF and ***SVI 30*** are configured for symmetric IRB.

VXLAN example

A netwok with 2 VRFs, ***red*** and ***blue*** has VLANs ***10*** and ***20*** in ***red*** and VLANs ***30*** and ***40*** in ***blue***. The spines in these act as a route reflectors. Multicast groups are used to encapsulate traffic arriving in a VRF such that it is delivered to VTEPs that have that VRF provisioned.

```
interface Loopback0
   ip address 10.0.0.20/32
!
vlan 10
vlan 20
vlan 30
vlan 40
!
interface Ethernet1
   switchport access vlan 10
!
interface Ethernet2
   switchport access vlan 20
!
interface Ethernet3
   switchport access vlan 30
!
interface Ethernet4
   switchport access vlan 40
!
```

```
interface Vlan10
   vrf red
   ip address virtual 192.168.1.0/24
   ip igmp
   pim ipv4 local-interface loopback0
!
interface Vlan20
   vrf red
   ip address 192.168.2.0/24
   ip igmp
!
interface Vlan30
   vrf blue
   ip address 192.168.1.0/24
   ip igmp
!
interface Vlan40
   vrf blue
   ip address 192.168.2.0/24
   ip igmp
!
interface Vxlan1
   vxlan source-interface Loopback0
   vxlan vlan 10 vni 10
   vxlan vlan 20 vni 20
   vxlan vlan 30 vni 30
   vxlan vlan 40 vni 40
   vxlan vrf red vni 100
   vxlan vrf blue vni 200
   vxlan vlan 10 flood group 225.1.1.2
   vxlan vlan 20 flood group 225.1.1.3
   vxlan vlan 30 flood group 226.1.1.2
   vxlan vlan 40 flood group 226.1.1.3
   vxlan vrf red multicast group 225.1.1.1
   vxlan vrf blue multicast group 226.1.1.1
!
```

18.12.2 Show Commands

The following examples are based on the sample topology and configuration in the previous sections.

On the remote VTEP, to display the EVPN routes to the multi-homed CE (***20.0.20.2***):

```
switch# show bgp evpn route-type mac-ip 20.0.20.2
BGP routing table information for VRF default
Router identifier 0.0.3.1, local AS number 302
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head,
                    e - ECMP S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

          Network                Next Hop              Metric  LocPref Weight  Path
 * >     RD: 10.255.0.0:20 mac-ip 0000.0101.0000 20.0.20.2
                                 10.255.0.0            -       100     0       303 300 i
 * >     RD: 10.255.0.1:20 mac-ip 0000.0101.0000 20.0.20.2
                                 10.255.0.1            -       100     0       303 301 i
```

As shown above, there are two EVPN MAC-IP routes for the multi-homed CE.

On the remote PE, to display the installed routes to the multi-homed CE:

```
switch# show ip route vrf red 20.0.20.2/32

VRF: red
Codes: C - connected, S - static, K - kernel,
```

```
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

 B E      20.0.20.2/32 [200/0] via VTEP 10.255.0.0 VNI 20000 router-mac 00:00:78:01:00:00
                               via VTEP 10.255.0.1 VNI 20000 router-mac 00:00:78:04:00:00
```

Two routes to the multi-homed CE are installed from the two EVPN MAC-IP routes and they form L3 ECMP.

On the remote PE, to check the details of the two routes in BGP RIB:

```
switch# show ip bgp 20.0.20.2/32 vrf red
BGP routing table information for VRF red
Router identifier 10.255.0.2, local AS number 302
BGP routing table entry for 20.0.20.2/32
 Paths: 2 available
  303 300
    10.255.0.0 from 10.0.2.1 (0.0.1.1), imported EVPN route, RD 10.255.0.0:20
      Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, received 00:14:43 ago,
      valid, external, ECMP head, ECMP, best, ECMP contributor
      Extended Community: Route-Target-AS:64500:10020 Route-Target-AS:64500:20000
      TunnelEncap:tunnelTypeVxlan EvpnMacMobility:1 EvpnRouterMac:00:00:78:01:00:00
      Remote VNI: 20000
      Rx SAFI: Unicast
  303 301
    10.255.0.1 from 10.0.2.1 (0.0.1.1), imported EVPN route, RD 10.255.0.1:20
      Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, received 00:14:43 ago,
      valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:64500:10020 Route-Target-AS:64500:20000
      TunnelEncap:tunnelTypeVxlan EvpnMacMobility:1 EvpnRouterMac:00:00:78:04:00:00
      EvpnNdFlags:pflag
      Remote VNI: 20000
      Rx SAFI: Unicast
```

The second route has **EvpnNdFlags:pflag** to indicate that this is a proxy MAC-IP route.

This command shows information about the SBD instance that is created when **evpn multicast** is configured under an IP VRF:

```
switch# show bgp evpn instance sbd red
EVPN instance: SBD red
  Route distinguisher: 100:1
  Service interface: VLAN-based
  Local IP address: 10.0.0.20
  Encapsulation type: VXLAN
vtep2#show bgp evpn instance sbd blue
EVPN instance: SBD red
  Route distinguisher: 200:1
  Service interface: VLAN-based
  Local IP address: 10.0.0.20
  Encapsulation type: VXLAN
```

The OISM supported capability is set in IMET routes for the SBD when **evpn multicast** is configured for a VRF. The IGMP proxy flag is not set in IMET routes for VLANs in the VRF unless**redistribute igmp** is configured in a VLAN:

```
switch# show bgp evpn route-type imet next-hop 10.0.0.10 detail
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
BGP routing table entry for imet 10.0.0.10, Route Distinguisher: 10:1
 Paths: 1 available
  Local
    10.0.0.10 from 10.0.0.1 (0.0.1.1)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:10:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10
      PMSI Tunnel: PIM-SSM Tree, MPLS Label: 10, Leaf Information Required: false,
      Tunnel ID: 10.0.0.10, 225.1.1.2
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
```

```
BGP routing table entry for imet 10.0.0.10, Route Distinguisher: 100:1
 Paths: 1 available
  Local
    10.0.0.10 from 10.0.0.1 (0.0.1.1)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:100:1 TunnelEncap:tunnelTypeVxlan Multicast
      Flags: IGMP proxy, OISM-supported, SBD
      VNI: 100
      PMSI Tunnel: Ingress Replication, MPLS Label: 100, Leaf Information Required: false,
      Tunnel ID: 10.0.0.10
```

This command shows information about the single SMET route for the group with the RD of ***VRF red***:

```
switch# show bgp evpn route-type smet multicast 228.1.1.1
BGP routing table information for VRF default
Router identifier 0.0.1.1, local AS number 300
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop
          Network                Next Hop              Metric  LocPref Weight  Path
 * >     RD: 100:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.10
                                 10.0.0.10             -       100     0       i
 * >     RD: 100:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD: 100:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD: 200:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD: 200:1 smet (S, G): (*, 228.1.1.1) originating IP: 10.0.0.40
                                 10.0.0.40             -       100     0       i
```

This command shows information about the SPMSI route for the group:

```
switch# show bgp evpn route-type spmsi
BGP routing table information for VRF defaultRouter identifier 0.0.0.1, local AS number 300
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop
          Network                Next Hop              Metric  LocPref Weight  Path
 * >     RD:  10:1 spmsi (S, G): (*, *) originating IP: 10.0.0.10
                                 10.0.0.10             -       100     0       i
 * >     RD:  10:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD:  20:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD:  30:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD:  40:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD:  10:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD:  20:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD:  30:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD:  40:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD:  30:1 spmsi (S, G): (*, *) originating IP: 10.0.0.40
                                 10.0.0.40             -       100     0       i
 * >     RD: 100:1 spmsi (S, G): (*, *) originating IP: 10.0.0.10
                                 10.0.0.10             -       100     0       i
 * >     RD: 100:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD: 200:1 spmsi (S, G): (*, *) originating IP: 10.0.0.20
                                 -                     -       -       0       i
 * >     RD: 100:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD: 200:1 spmsi (S, G): (*, *) originating IP: 10.0.0.30
                                 10.0.0.30             -       100     0       i
 * >     RD: 200:1 spmsi (S, G): (*, *) originating IP: 10.0.0.40
                                 10.0.0.40             -       100     0       i
```

18.12.3 Limitations

The following limitations are included in the EVPN VXLAN all-active multi-homing integrated routing and bridging feature:

- L2 loop-free protocols, such as Spanning Tree Protocol (STP) are not supported between PE-CE with EVPN VXLAN All-Active Multi-homing. Users must ensure their topology is loop-free. STP must be disabled for the Multihomed VLANs on PEs and CEs.
- A limit of two multi-homing destinations are installed at one time for any particular MAC address. Any other valid destinations are ignored in the context of switching unicast traffic until one of the active destinations is removed. The multi-homing PEs themselves operate normally regardless of the number of PEs on the ethernet segment; this limitation only affects the selection of a destination for unicast traffic.
- VXLAN GPE is not currently supported, so packets cannot be flagged as having been broadcast. As a result, unicast packets that are known at the sender and unknown at the receiver are dropped if the receiving PE is not a designated forwarder. Similarly, unicast packets that are unknown at the sender and known at the receiver are duplicated at the receiving CE. This state automatically resolves itself as the BGP network distributes the appropriate type 1 auto-discovery and type 2 MAC/IP advertisement EVPN routes.
- Fast mass withdrawal is not supported, so there may be a delay if an interface harboring a large number of MAC addresses goes down.

18.13 Configuring EVPN

18.13.1 Configuring BGP-EVPN and VCS on CVX

18.13.1.1 Configuring BGP-EVPN

Configuring VNI Bundle

A VNI-aware-bundle represents a MAC-VRF that contains Layer 2 route entries from all VXLAN Network Identifiers (VNI) available across multiple DCs. Use the **vni-aware-bundle** command available on CVX to create a MAC-VRF.

Note: This command is not available on switches.

Example

```
cvx(config)# router bgp 100
cvx(config-router-bgp)# vni-aware-bundle bundle1
cvx(config-macvrf-bundle1)#
```

Configuring RD and RT in VNI Bundle

Use the **rd** (Router-BGP VRF and VNI Configuration Modes) command to add a Route Distinguisher (RD) for uniquely identifying Layer 2 routes for the VNI bundle. Use the **route-target** command to configure a well-known extended community that is attached to the routes exported by BGP-EVPN; and to import routes with the specified well-known extended community into the MAC-VRF that corresponds to the VNI bundle.

Example

```
cvx(config)# router bgp 100
cvx(config-router-bgp)# vni-aware-bundle bundle1
cvx(config-macvrf-bundle1)# rd 530:12
```

```
cvx(config-macvrf-bundle1)# route-target both 530:12
```

Enabling Redistribution of Bridging Information

After the VNI aware bundle is created, use the **redistribute service vxlan** command to redistribute the Layer 2 bridging information received from VCS.

Example

```
cvx(config)# router bgp 100
cvx(config-router-bgp)# vni-aware-bundle bundle1
cvx(config-macvrf-bundle1)# redistribute service vxlan
```

Disabling Next-Hop Resolution in BGP-EVPN

When BGP-EVPN module receives a route from its BGP peer, it generally tries to resolve the next-hop indicated in the route. However in the DCI topology, the routes coming from a CVX in another DC contains next-hops (VTEP addresses) that may not be reachable from the CVX receiving the route. Use the **next-hop resolution disabled** command to disable the next-hop resolution on routes received from BGP-EVPN peers.

Note: CVX is a part of the control plane and it is only connected to the VTEPs in its own DC. It does not have IP connectivity to the VTEPs in a different DC.

Example

```
cvx(config)# router bgp 100
cvx(config-router-bgp)# address-family evpn
cvx(config-router-bgp-af)# next-hop resolution disabled
```

18.13.1.2 Configuring VCS

Enabling Redistribution of BGP-EVPN Routes

Use the **redistribute bgp evpn vxlan** command to redistribute BGP-EVPN routes to VCS, which, in turn advertises them to all VTEPs within the DC.

Example

```
cvx(config)# cvx
cvx(config-cvx)# no shutdown
cvx(config-cvx)# service vxlan
cvx(config-cvx-vxlan)# no shutdown
cvx(config-cvx-vxlan)# redistribute bgp evpn vxlan
```

18.13.2 EVPN MPLS Virtual Private Wire Service (VPWS)

Traffic to / from a given Attachment Circuit (AC) without any MAC lookup / learning can be forwarded using EVPN MPLS VPWS, which uses BGP for signalling. Port based and VLAN based services are supported.

Configuring EVPN MPLS VPWS

Configure the patch panel to specify the connection of the ACs to the VPWS service instances, and then the VPWS service instance, which is part of BGP. Finally, configure the individual participating ACs.

Patch Panel Configuration

The following configures the local AC as ***Ethernet2*** interface and the remote VPWS service instance as evi-1 and pseudowire pw1.

```
patch panel
            patch port
                connector 1 interface Ethernet2
                connector 2 pseudowire bgp vpws evi-1 pseudowire pw1
```

The following configures the local AC as ***Ethernet3.1*** subinterface and the remote VPWS service instance as ***evi-1*** and pseudowire ***pw2***.

```
patch panel
             patch subintf
                 connector 1 interface Ethernet3.1
                 connector 2 pseudowire bgp vpws evi-1 pseudowire pw2
```

Note: Connector ID is optional.

VPWS Service Instance Configuration

The following configures the VPWS service instance with the BGP ***vpws*** sub mode. This defines an EVPN instance under which any number of VPWS service instances can be configured. The BGP configuration itself can also define multiple EVPN instances under multiple **vpws** blocks, each with a unique name and Route-Distinguisher (RD) value. Only the **mpls control-word** and **mtu *value*** configuration items are optional; the rest are required for proper operation.

```
router bgp 1
   neighbor 10.0.0.1 remote-as 1
   neighbor 10.0.0.1 send-community extended
   neighbor 10.0.0.1 maximum-routes 12000
   !
   vpws evi-1
      rd 10.2.2.2:2
      route-target import export evpn 0.0.0.0:1
      mpls control-word
      !
      pseudowire pw1
         evpn vpws id local 2001 remote 1001
      !
      pseudowire pw2
         evpn vpws id local 2002 remote 1002
   !
   address-family evpn
      neighbor default encapsulation mpls next-hop-self source-interface
 Loopback0
      neighbor 10.0.0.1 activate
```

Note: It is strongly recommended that 'mpls control-word' is always enabled, when possible, to avoid any potential mis-forwarding where the PWE frames may be incorrectly interpreted as having an IP, as opposed to Ethernet, payload.

Attachment Circuit Configuration (double-tagged L3 subinterfaces)

The following configures the AC in Port mode.

```
interface Ethernet2
   no switchport
```

Note: Use Ethernet or Port-channel interface for Port mode.

The following configures the AC in VLAN mode.

```
interface Ethernet3
   no switchport
interface Ethernet3.1
   encapsulation dot1q vlan 1
```

Note: Use subinterfaces for VLAN mode.

The following configures the AC in Flexible Encapsulation mode. The **client** after 'network' preserves the corresponding client encapsulation specification.

```
interface Ethernet3
   no switchport
interface Ethernet3.1
   encapsulation vlan
   client dot1q 11 network client
```

Flexible Encapsulation EVPN MPLS VPWS

Flexible encapsulation enables the following actions for tags.

- Remove incoming encapsulation tag(s) and forward
- Preserve incoming encapsulation tag(s) and forward
- Replace one or two tags when forwarding in encapsulation and decapsulation directions

The table below explains the encapsulation and decapsulation behaviors for the various FlexEncap options. Applying a Flexible Encapsulation with a ***network*** specification to a subinterface creates a bidirectional mapping table that is applied to the sub-interface. The mapping embodied in this table is applied from ***client*** to ***network*** in the encap direction, and ***network*** to ***client*** in the decap direction.

Example	Behavior
client dot1q 10	**From Client:** match VLAN ID 10, consume and forward **To Client:** add VLAN ID 10 before transmit
client dot1q 10 inner 20	**From Client:** match VLAN IDs 10, 20 consume and forward **To Client:** add VLAN ID 10, 20 before transmit
client dot1q 10 network client	**From Client:** match VLAN ID 10 and retain it. **From Network:** match vlan=10, retain.
client dot1q outer 10 inner 20 network client	**From Client:** match VLAN IDs 10, 20 and retain both. **From Network:** match vlan=10,20, retain both.
client dot1q 10 network dot1q 100	
client dot1q 10 network dot1q 100	**From Client:** match VLAN ID 10, consume. Before forwarding, write vlan=100. **From Network:** match vlan=100, consume. Before transmit, write vlan=10.
client dot1q outer 10 inner 20 network dot1q outer 100 inner 200	**From Client:** match VLAN IDs 10, 20, and consume them. Before forwarding, write vlan=100,200. **From Network:** match vlan=100, 200, consume. Before transmit, write vlan=10, 20.

The following configures FlexEncap on a subinterface as a local connector and LDP pseudowire as remote connector.

- Packets received on ***Ethernet3/1*** with outermost 802.1q VLAN tag of ***1000*** get mapped to sub-interface ***Ethernet3/1.1000***.
- The tag of ***1000*** is preserved and forwarded to pseudowire ***PW1***.
- Packets terminating on ***PW1*** get forwarded to ***Et3/1.1000*** and get transmitted out with VLAN tag of ***1000***.

```
interface Ethernet3/1.1000
   encapsulation vlan
      client dot1q 1000 network client
patch panel
   patch patch-1
      connector 1 interface Ethernet3/1.1000
      connector 2 pseudowire ldp PW1
```

Displaying EVPN MPLS VPWS Configuration

This command shows both the client encapsulation and network encapsulation configured on sub-interfaces.

```
switch(config-if-Et3/1.1003)# show interfaces encapsulation vlan
Interface                 Status       Client Encapsulation      Network
 Encapsulation
```

```
------------------------ ------------ ------------------------------------
------------
Ethernet3/1.1000       active       dot1q outer 1000
Ethernet3/1.1001       active       dot1q outer 1001             client
Ethernet3/1.1002       active       dot1q outer 1002 inner 102
Ethernet3/1.1003       active       dot1q outer 1003 inner 103   client
Ethernet3/1.1004       active       dot1q outer 1004             dot1q 2004
Ethernet3/1.1005       active       dot1q outer 1005 inner 104   dot1q outer
 2005 inner 204
```

This command shows output of a patch with sub-interface as the local connector and VPWS as the remote connector.

```
switch(config-if-Et3/1.1003)# show patch panel PP_1000

Patch   Connector                              Status
------- ------------------------------------ ------
PP_1000 1: Ethernet3/1.1000                    Up
        2: BGP VPWS VPWS_1 Pseudowire PW_1000

tg481.12:19:52(s2)(config-if-Et3/1.1003)#show patch panel PP_1000 detail
PW Fault Legend:
   ET-IN - Ethernet receive fault
   ET-OUT - Ethernet transmit fault
   TUN-IN - Tunnel receive fault
   TUN-OUT - Tunnel transmit fault
   NF - Pseudowire not forwarding (other reason)

Patch: PP_1000, Status: Up
   Connector 1: Ethernet3/1.1000
      Status: Up
   Connector 2: BGP VPWS VPWS_1 Pseudowire PW_1000
      Status: Up
      Local MPLS label: 135363
         MTU: 1600, Control word: Y
      Neighbor 103.37.123.72, MPLS label: 136350
         Tunnel type: SR-TE Policy, Tunnel index: 132
         MTU: 1600, Control word: Y
      EVPN VPWS type: VLAN-based
```

Tag Matching Semantics

The matching rules are applied on a 'longest matching tag sequence' basis when rules are configured for multiple subinterfaces of a parent port. Considering the following rules on the same parent, the receive (encap) and transmit (decap) rule application is shown in the following tables.

Rule 1:

```
interface Ethernet 10.1
  encapsulation vlan
    client dot1q 11 network client
```

Rule 2:

```
interface Ethernet 10.2
  encapsulation vlan
    client dot1q 11 inner 20 network client
```

The receive (encap) matching behavior is as follows.

Received Packet	Matching Rule
outer=11, inner=20	Rule #2
single tag with 11	Rule #1
double tag with 11, not 20	Rule #1

The transmit (decap) matching behavior is as follows.

Forwarded Packet	Matching Rule
outer=11, inner=20	Rule #2
single tag with 11	Rule #1
double tag with 11, not 20	Rule #1

18.14 Sharing Equivalence Class entry across multiple VRF

This enables the sharing of the same Forwarding Equivalence Class (FEC) entry across multiple VRF's and achieves higher scale in VPN deployments by making optimal use of hardware resources. The VPN path routes are imported to multiple VRF for MPLS VPN address family in BGP, individual FEC entry in the hardware table is used for each VRF.

The following configuration in router BGP mode at BGP instance level allows sharing FEC, when the VPN path is imported to multiple VRFs.

bgp fec skip in-place update

The following configures disable FEC and restores default configuration.

[no| default] bgp fec skip in-place update

Note: Changing the configuration in a production system may increase FEC usage.

The following example displays BGP VPN-IPv4 routes for prefix ***20.0.1.0/24***, which is imported to four VRFs.

```
switch# show bgp vpn-ipv4 20.0.1.0/24 detail
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
BGP routing table entry for IPv4 prefix 20.0.1.0/24, Route Distinguisher:
 11.0.1.1:0
 Paths: 1 available
  301
    11.0.1.1 from 10.0.0.2 (0.0.1.1)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, best
      Extended Community: Route-Target-AS:300:0 Route-Target-AS:301:0
      MPLS label: 116507
```

This route was using 4 FEC entries for each VRF before the change.

```
switch# show platform jericho ip route 20.0.1.0/24
Tunnel Type: M(mpls), G(gre), MoG(mpls-over-gre),
             vxlan-o(vxlan outer-rewrite info), vxlan-i(vxlan inner-rewrite info)
* - Routes in LEM
D - ECMP is divergent across switching chips
 ------------------------------------------------------------------------------------------------------------
|                                  Routing Table                                                 |            |
|------------------------------------------------------------------------------------------------------------
|VRF|   Destination    |     |                  |     |       |                  | ECMP|  FEC | Tunnel
| ID|      Subnet      | Cmd |     Destination    | VID |Outlif |   MAC / CPU Code  |Index| Index|T Value
 ------------------------------------------------------------------------------------------------------------
|1  |20.0.1.0/24       |ROUTE| FEC 32776          |0    |  -    |                  |  -  |49152 |M 116507
```

```
|2  |20.0.1.0/24       |ROUTE| FEC 32776          |0    |  -    |                    |  -  |49154 |M 116507
|3  |20.0.1.0/24       |ROUTE| FEC 32776          |0    |  -    |                    |  -  |49155 |M 116507
|4  |20.0.1.0/24       |ROUTE| FEC 32776          |0    |  -    |                    |  -  |49156 |M 116507
```

The following uses single FEC entry For VPN routes imported to all VRFs after configuring **`bgp fec skip in-place update event all`** command.

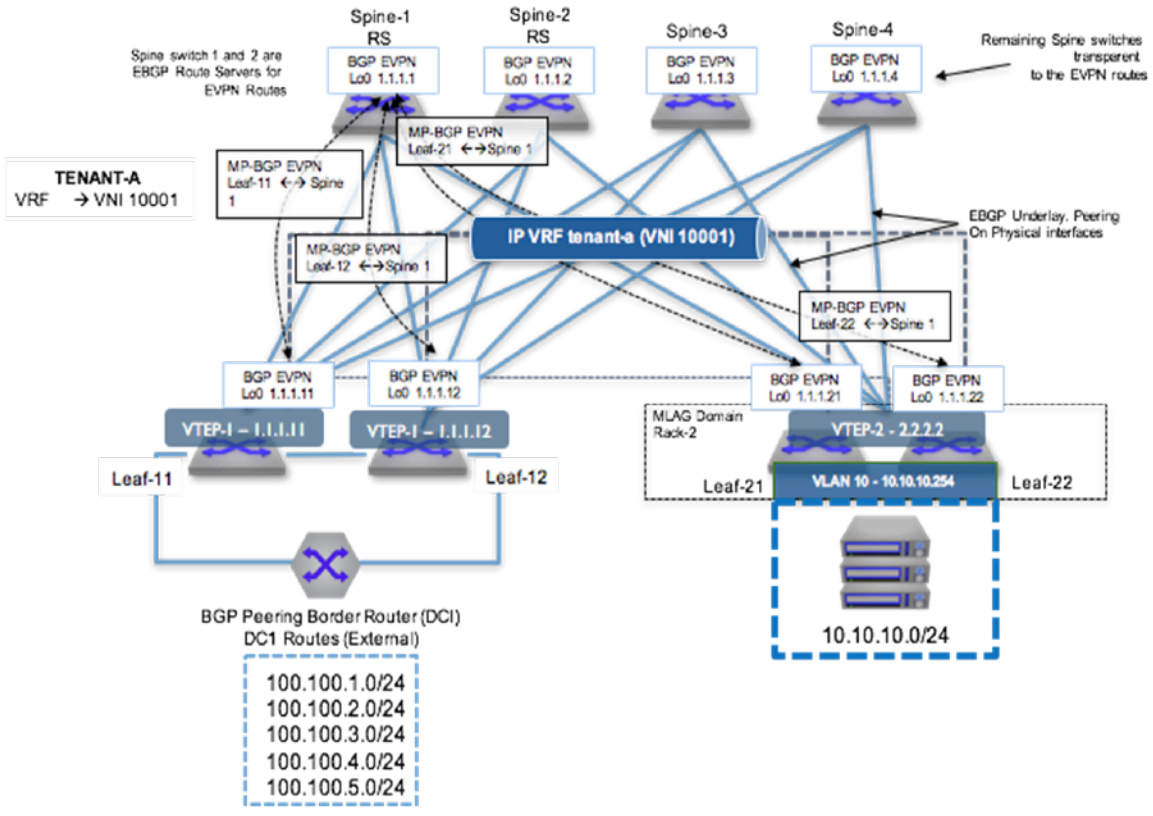
```
switch# show platform jericho ip route 20.0.1.0/24
Tunnel Type: M(mpls), G(gre), MoG(mpls-over-gre),
             vxlan-o(vxlan outer-rewrite info), vxlan-i(vxlan inner-rewrite info)
* - Routes in LEM
D - ECMP is divergent across switching chips
 ------------------------------------------------------------------------------------------------------------
|                                  Routing Table                                                 |           |
|-----------------------------------------------------------------------------------------------------------|
|VRF|   Destination    |      |                  |     |       |                  | ECMP|  FEC | Tunnel
| ID|      Subnet      | Cmd  |     Destination  | VID |Outlif |   MAC / CPU Code |Index| Index|T Value
 ------------------------------------------------------------------------------------------------------------
|1  |20.0.1.0/24       |ROUTE| FEC 32780          |0    |  -    |                    |  -  |49152 |M 116507
|2  |20.0.1.0/24       |ROUTE| FEC 32780          |0    |  -    |                    |  -  |49152 |M 116507
|3  |20.0.1.0/24       |ROUTE| FEC 32780          |0    |  -    |                    |  -  |49152 |M 116507
|4  |20.0.1.0/24       |ROUTE| FEC 32780          |0    |  -    |                    |  -  |49152 |M 116507
```

18.15 Sample Configurations

18.15.1 EVPN VXLAN IRB Sample Configuration

In the topology below, we are connecting a Layer 2 site with a Layer 3 site using Layer 3 EVPN (type-5 route). Right side leaves are MLAG leaves and have SVI 10 in VRF-Blue. A number of directly connected hosts are simulated behind the right side leaf. The left side leaves are individual leaves that connect with a remote switch in **`vrf`** VRF-Blue to learn Layer 3 routes using BGP. The left side leaves are configured as two independent Layer 3 only VTEPs.

Figure 18-48: Layer 3 EVPN Configuration



To provide VXLAN routing and bridging between the two MLAG domains, each leaf switch is EVPN peering with the four spine switches via a loopback interface.

eBGP Underlay Configuration: Leaf-11

Underlay configuration is straightforward and all neighbors are eBGP. Since all leaves share the same AS number, the **`allowas-in`** command was added in the leaf.

```
interface Ethernet1
   description Spine-1-et1/1
   mtu 9214
   no switchport
   ip address 172.168.1.1/31

interface Ethernet8/1
   description ck428-et8/1
   speed forced 40gfull
   no switchport
   ip address 172.168.1.10/31

interface Loopback0
   ip address 1.1.1.11/32

ip prefix-list loopback
   seq 10 permit 1.1.1.0/24 ge 24
!
route-map loopback permit 10
   match ip address prefix-list loopback

router bgp 65004
neighbor SPINE peer-group
   neighbor SPINE remote-as 65001
   neighbor SPINE allowas-in 1
   neighbor SPINE soft-reconfiguration inbound all
   neighbor SPINE send-community
   neighbor 172.168.1.0 peer-group SPINE
   neighbor 172.168.1.11 remote-as 65003
   redistribute connected route-map loopback
```

eBGP Underlay Configuration: Spine-1

```
interface Ethernet1/1
   description Leaf-11-et1
   mtu 9214
   no switchport
   ip address 172.168.1.0/31

interface Loopback0
   ip address 1.1.1.1/32
!
ip prefix-list loopback
   seq 10 permit 1.1.1.0/24 ge 24
!
route-map loopback permit 10
   match ip address prefix-list loopback
!
router bgp 65001
   neighbor 172.168.1.1 remote-as 65004
   redistribute connected route-map loopback
```

VRF Configuration: Leaf-11

VRF-Blue is configured on all the left leaves. The left leaves have pure Layer 3 interfaces and the right side has ***SVI 10***.

```
vrf instance VRF-Blue

ip routing vrf VRF-Blue

interface Ethernet36
   no switchport
   vrf VRF-Blue
   ip address 172.168.1.9/31

router bgp 65004
     vrf VRF-Blue
           neighbor 172.168.1.8 remote-as 65005
```

VRF Configuration: Leaf-21

```
vlan 10

vrf instance VRF-Blue

ip routing vrf VRF-Blue

interface Vlan10
   vrf VRF-Blue
   ip address virtual 10.10.10.1/24

ip virtual-router mac-address 00:aa:aa:aa:aa:aa

interface Port-Channel3
   switchport mode trunk
   mlag 3
```

VXLAN Configuration: Leaf-11

Make sure all VTEPs have unique loopback0 addresses to represent unique VTEP identifiers. For every VNI that EVPN receives, a dynamic VLAN is allocated, so it is a good practice to keep the same VNI.

```
interface Vxlan1
   vxlan source-interface Loopback0
   vxlan udp-port 4789
   vxlan vrf VRF-Blue vni 10001
```

VXLAN Configuration: Leaf-21

```
interface Vxlan1
   vxlan source-interface Loopback0
   vxlan udp-port 4789
   vxlan vrf VRF-Blue vni 10001
```

EVPN Configuration: Leaf-11

Leaf establishes the EVPN neighborship with all four spines for redundancy. EVPN neighborship is on the loopback address and the **multihop** keyword is used. Make sure to disable the IPv4 address family for EVPN neighbors.

Since the spine is acting like a route-reflector for EVPN routes, make sure to configure the next-hop-unchanged.

```
router bgp 65004
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN ebgp-multihop 3
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   !
   address-family evpn
      neighbor SPINE_EVPN activate
   !
   address-family ipv4
      no neighbor SPINE_EVPN activate
```

EVPN Configuration: Leaf-21

```
router bgp 65002
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN allowas-in 1
   neighbor SPINE_EVPN ebgp-multihop 3
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   !
   address-family evpn
      neighbor SPINE_EVPN activate
   !
   address-family ipv4
      no neighbor SPINE_EVPN activate
```

EVPN Configuration: Spine-1

```
router bgp 65004
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN ebgp-multihop 3
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   !
   address-family evpn
      neighbor SPINE_EVPN activate
   !
   address-family ipv4
      no neighbor SPINE_EVPN activate
```

Advertise VRF Routes in EVPN: Leaf-11

By configuring VRF under **`router-bgp`**, you are advertising routes from that VRF into EVPN using the RD/RT. The remote end can install the route by importing the RT.

Leaf-11 has routes in ***VRF-Blue*** learned through eBGP with the neighbor down south. Since the routes are already in BGP VRF table, we do not want to configure the **`redistribute`** command.

```
router bgp 65004
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN ebgp-multihop 3
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   !
   address-family evpn
      neighbor SPINE_EVPN activate
   !
   address-family ipv4
      no neighbor SPINE_EVPN activate
```

Advertise VRF Routes in EVPN: Leaf-21

On the other hand ***Leaf-21*** wants to export the connected SVI into EVPN and therefore require **`redistribute connected`** command.

```
router bgp 65002
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN allowas-in 1
   neighbor SPINE_EVPN ebgp-multihop 3
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   !
   address-family evpn
      neighbor SPINE_EVPN activate
   !
   address-family ipv4
      no neighbor SPINE_EVPN activate
```

18.15.2 Multi-Tenant EVPN VXLAN IRB Sample Configuration

The following configuration example shows a deployment using both symmetric and asymmetric IRB with VLAN-based and VLAN-aware bundle services; and eBGP overlay and underlay.

Figure 18-49: Tenant-A: Symmetric IRB

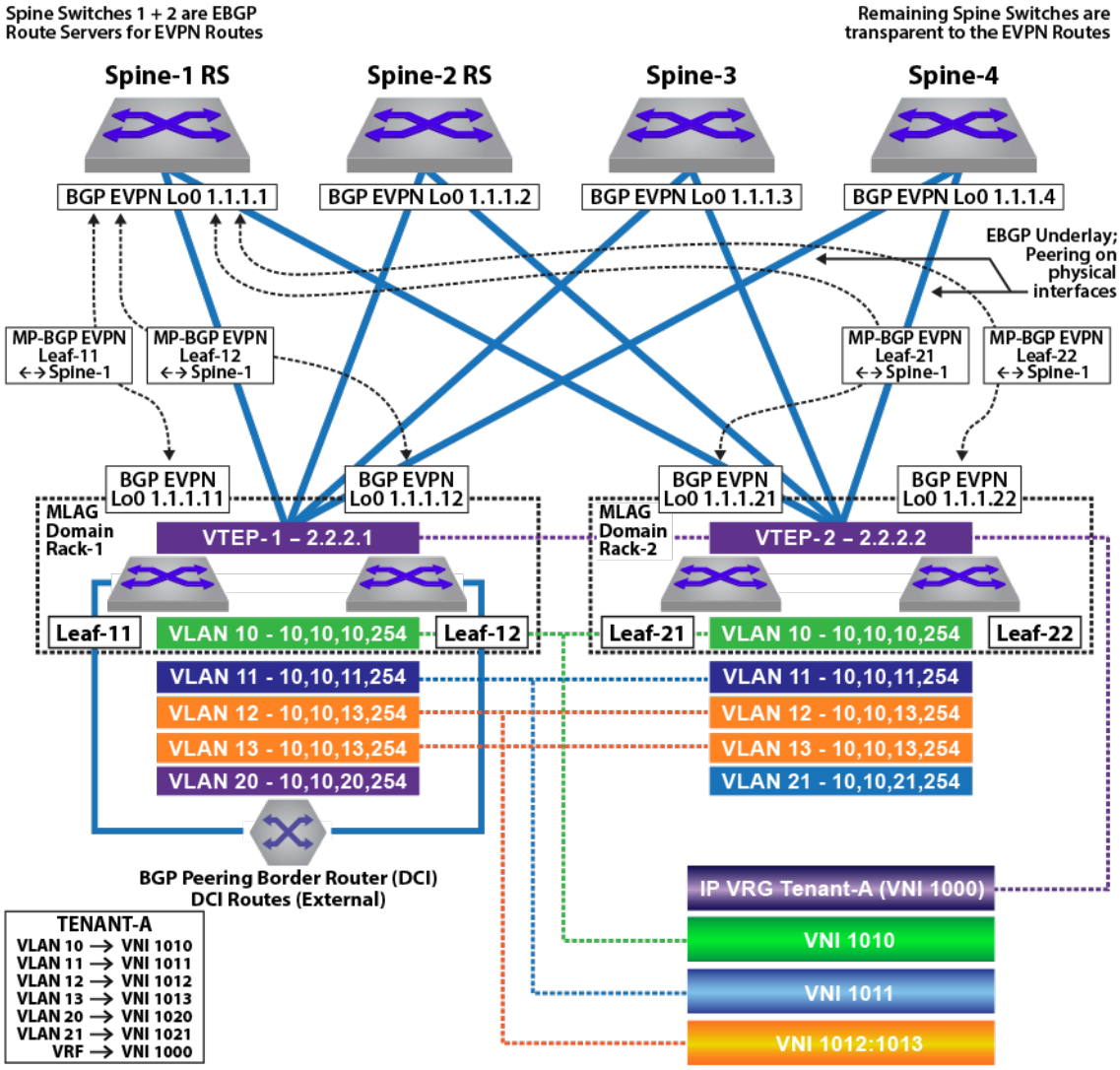
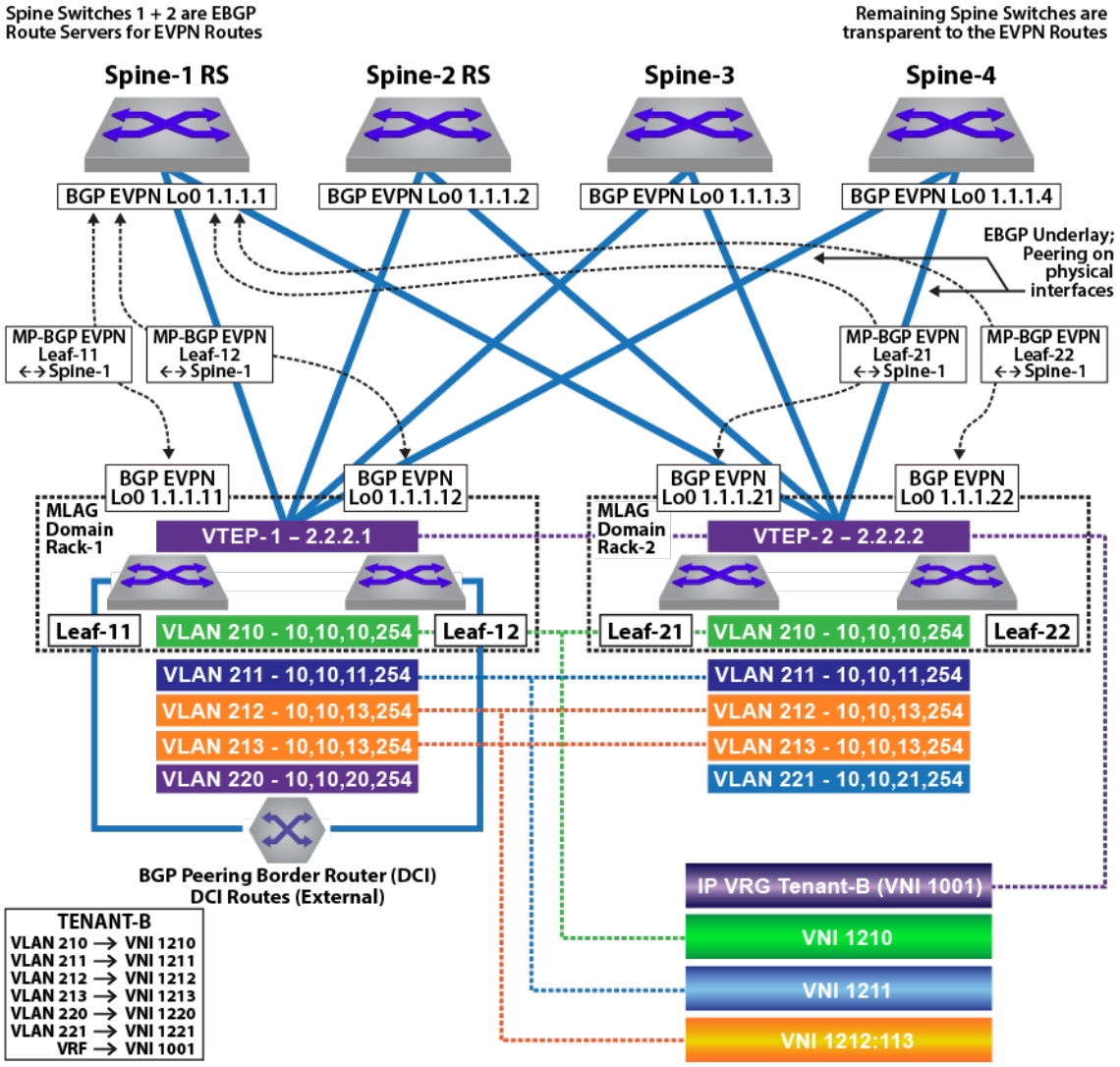


Figure 18-50: Tenant-B: Asymmetric IRB



In the symmetric and asymmetric IRB configurations illustrated in the figures above, for ***Tenant-A***, four subnets are stretched across the two MLAG domains with two subnets (***VLAN 10***, ***10.10.10.0/24*** and ***VLAN 11***, ***10.10.11.0/24***) configured as a VLAN-based service, and two other subnets (***VLAN 12***,***10.10.12.0/24*** and ***VLAN 13***, ***10.10.13.0/24***) as a VLAN-aware bundle service.

For ***Tenant-B***, four subnets are stretched across the two MLAG domains with two subnets (***VLAN 210***, ***10.10.10.0/24*** and ***VLAN 211***,***10.10.11.0/24***) configured as a VLAN-based service, and two other subnets (***VLAN 212***,***10.10.12.0/24*** and ***VLAN 213***,***10.10.13.0/24***) as a VLAN-aware bundle service.

In addition each MLAG domain has a single local subnet (***Rack-1*** subnet ***10.10.20.0/24*** and ***Rack-2*** subnet ***10.10.21.0/24***) for the tenant. To provide direct distributed routing, each leaf switch is configured with the same virtual IP address for the four stretched subnets. For the local-only subnets, the virtual IP address is configured in both physical leaf switches of the relevant MLAG domain.

For each MLAG domain, a logical VTEP is created with the same shared loopback address. For ***Rack-1***, the logical VTEP IP is ***2.2.2.1*** and for the ***Rack-2***, the logical VTEP IP is ***2.2.2.2***. Directly connected to each leaf switch is a host, which is a member of one of the two IP subnets. To provide Layer 2 connectivity across the racks, VXLAN bridging is enabled by mapping VLAN to VNIs as detailed in the diagram.

To provide IP connectivity across all subnets both stretched and directly connected, an IP-VRF is shared between the two MLAG domains for the tenant. This is used as a transit network for announcing and

forwarding the locally attached subnets. Each leaf switch is EVPN peering with the four spine switches via a loopback interface on the leaf and again on the spine switches. To provide external connectivity, ***Leaf-11*** and ***Leaf-12*** are eBGP peering via the tenants' VRFs with the border routers. Both core routers are advertising external prefixes for Internet and any remote site connectivity (default route and IP prefixes from the other DC for the tenant). To provide connectivity within the EVPN domain, the leaf switches (***Leaf-21*** and ***Leaf-22***) re-advertise the prefixes into the tenant's VRF via a type-5 route advertisement, with a next-hop equal to the advertising VTEP.

18.15.2.1 MLAG Configuration: Leaf-11 and Leaf-12

Leaf-11 MLAG Configuration

```
spanning-tree mode mstp
no spanning-tree vlan-id 4093-4094
!
ip virtual-router mac-address mlag-peer
!
vlan 4094
   name MLAG_PEER
   trunk group MLAG
!
vlan 4093
   name LEAF_PEER_L3
   trunk group LEAF_PEER_L3
!
interface Vlan4094
   ip address 172.168.10.1/30
!
interface Port-Channel100
   description port-channel to access switch
    switchport trunk allowed vlan 10-13,20,210-213,220
   switchport mode trunk
   mlag 1
!
interface Port-Channel1000
   switchport mode trunk
   switchport trunk group LEAF_PEER_L3
   switchport trunk group MLAG
!
mlag configuration
   domain-id Rack-1
   local-interface Vlan4094
   peer-address 172.168.10.2
   peer-link Port-Channel1000
```

Leaf-12 MLAG Configuration

```
spanning-tree mode mstp
no spanning-tree vlan-id 4093-4094
!
ip virtual-router mac-address mlag-peer
!
vlan 4094
   name MLAG_PEER
   trunk group MLAG
!
vlan 4093
   name LEAF_PEER_L3
   trunk group LEAF_PEER_L3
!
```

```
interface Vlan4094
   ip address 172.168.10.2/30
!
interface Port-Channel100
   description port-channel to access switch
    switchport trunk allowed vlan 10-13,20,210-213,220
   switchport mode trunk
   mlag 1
!
interface Port-Channel1000
   switchport mode trunk
   switchport trunk group LEAF_PEER_L3
   switchport trunk group MLAG
!
mlag configuration
   domain-id Rack-1
   local-interface Vlan4094
   peer-address 172.168.10.1
   peer-link Port-Channel1000
```

18.15.2.2 MLAG Configuration: Leaf-21 and Leaf-22

Leaf-21 MLAG Configuration

```
spanning-tree mode mstp
no spanning-tree vlan-id 4093-4094
!
ip virtual-router mac-address mlag-peer
!
vlan 4094
   name MLAG_PEER
   trunk group MLAG
!
vlan 4093
   name LEAF_PEER_L3
   trunk group LEAF_PEER_L3
!
interface Vlan4094
   ip address 172.168.10.1/30
!
interface Port-Channel100
   description port-channel to access switch
  switchport trunk allowed vlan 10-13,21,210-213,220-221
   switchport mode trunk
   mlag 1
!
interface Port-Channel1000
   switchport mode trunk
   switchport trunk group LEAF_PEER_L3
   switchport trunk group MLAG
!
mlag configuration
   domain-id Rack-1
   local-interface Vlan4094
   peer-address 172.168.10.2
   peer-link Port-Channel1000
```

Leaf-22 MLAG Configuration

```
spanning-tree mode mstp
```

```
no spanning-tree vlan-id 4093-4094
!
ip virtual-router mac-address mlag-peer
!
vlan 4094
   name MLAG_PEER
   trunk group MLAG
!
vlan 4093
   name LEAF_PEER_L3
   trunk group LEAF_PEER_L3
!
interface Vlan4094
   ip address 172.168.10.2/30
!
interface Port-Channel100
   description port-channel to access switch
  switchport trunk allowed vlan 10-13,21,210-213,220-221
   switchport mode trunk
   mlag 1
!
interface Port-Channel1000
   switchport mode trunk
   switchport trunk group LEAF_PEER_L3
   switchport trunk group MLAG
!
mlag configuration
   domain-id Rack-1
   local-interface Vlan4094
   peer-address 172.168.10.1
   peer-link Port-Channel1000hannel1000
```

18.15.2.3 VLAN and Distributed IP Address Configuration: Leaf-11 and Leaf-21

VLAN and interface configuration for ***VLAN 10*** (virtual IP address ***10.10.10.254***) and ***VLAN 11*** (virtual IP address ***10.10.11.254***), along with SVIs ***12***, ***13***, and ***20***, are similarly configured. To provide multi-tenancy, the two tenant VLANs are placed in a dedicated VRF, named ***Tenant-A***. A further five tenant VLANs are configured and assigned to VRF ***Tenant-B***.

The other VLANs are for peering, MLAG, and a unique VLAN SVI. These VLANs do not use virtual IP addresses.

The tenants' stretched subnets (***Tenant-A***: VLANs ***10***,***11***,***12***, and ***13***; ***Tenant-B***: VLANs ***210***, ***211***, ***211***, ***212***, and ***213***) are mapped to unique overlay VXLAN VNIs. The tenants' IP-VRF (***Tenant-A*** and ***Tenant-B***) is associated with a VNI using the **`vxlan vrf`** command under the VXLAN interface. In the forwarding model for symmetric IRB, this VNI will be used as the transit VNI for routing to subnets which are not locally configured on the VTEP.

As a standard MLAG configuration, both leaf switches in each MLAG domain share the same logical VTEP IP address. Thus MLAG domain, ***Rack-1*** (***Leaf-11*** + ***Leaf-12***) has a shared logical VTEP IP of ***2.2.2.1*** and ***Rack-2*** (***Leaf-21*** + ***Leaf-22***) has a shared logical VTEP IP of ***2.2.2.2***.

Leaf-11 VLAN and Distributed IP Address Configuration

```
!
ip virtual-router mac-address 00:aa:aa:aa:aa:aa
!
vlan 10-11,20,210-211,220,111,2111
!
vlan 12-13
   name VLAN-AWARE-BUNDLE-TENANT-A
```

```
!
vlan 212-213
   name VLAN-AWARE-BUNDLE-TENANT-B
!
vrf instance tenant-a
!
vrf instance tenant-b
!
interface lan10
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.10.254/24
!
interface Vlan11
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.11.254/24
!
interface Vlan12
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.12.254/24
!
interface Vlan13
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.13.254/24
!
interface Vlan20
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.20.254/24
!
interface Vlan210
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.10.254/24
!
interface Vlan211
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.11.254/24
!
interface Vlan212
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.12.254/24
!
interface Vlan213
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.13.254/24
!
interface Vlan220
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.20.254/24
!
interface Vlan1111
   description Unique-highest-IP-in-each-IP-Vrf
   mtu 9164
   vrf tenant-a
   ip address 223.255.255.249/30
!
```

```
interface Vlan2111
   description Unique-highest-IP-in-each-IP-Vrf
   mtu 9164
   vrf tenant-b
   ip address 223.255.255.249/30
!
interface Vlan4093
   ip address 172.168.11.1/30
```

Leaf-21 VLAN and Distributed IP Address Configuration

```
!
ip virtual-router mac-address 00:aa:aa:aa:aa:aa
!
vlan 10-11,20,210-211,220,111,2111
!
vlan 12-13
   name VLAN-AWARE-BUNDLE-TENANT-A
!
vlan 212-213
   name VLAN-AWARE-BUNDLE-TENANT-B
!
vrf instance tenant-a
!
vrf instance tenant-b
!
interface Vlan10
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.10.254/24
!
interface Vlan11
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.11.254/24
!
interface Vlan12
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.12.254/24
!
interface Vlan13
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.13.254/24
!
interface Vlan21
   mtu 9164
   vrf tenant-a
   ip address virtual 10.10.21.254/24
!
interface Vlan210
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.10.254/24
!
interface Vlan211
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.11.254/24
!
interface Vlan212
```

```
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.12.254/24
!
interface Vlan213
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.13.254/24
!
interface Vlan221
   mtu 9164
   vrf tenant-b
   ip address virtual 10.10.21.254/24
!
interface Vlan1111
   description Unique-highest-IP-in-each-IP-Vrf
   mtu 9164
   vrf tenant-a
   ip address 223.255.255.253/30
!
interface Vlan2111
   description Unique-highest-IP-in-each-IP-Vrf
   mtu 9164
   vrf tenant-b
   ip address 223.255.255.253/30
!
interface Vlan4093
   ip address 172.168.11.1/30
!
```

18.15.2.4 VXLAN Interface Configuration: Leaf-11 and Leaf-21

The tenants' VLANs are mapped to unique overlay VXLAN VNIs. ***VLAN 10*** is mapped to ***VNI 1010*** on both MLAG domains, and ***VLAN 11*** is mapped to ***VNI 1011***. As standard MLAG configuration, both leaf switches in each MLAG domain share the same logical VTEP IP address. Thus MLAG domain ***Rack-1*** (***Leaf-11*** + ***Leaf-12***) has a shared logical VTEP IP of ***2.2.2.1*** and ***Rack-2*** (***Leaf-21*** + ***Leaf-22***) has a shared logical VTEP IP of ***2.2.2.2***. Also configured is the VRF-to-VXLAN mapping for ***Tenant-A***.

Leaf-11 VXLAN Interface Configuration

```
!
interface Loopback1
   ip address 2.2.2.1/32
!
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 10 vni 1010
   vxlan vlan 11 vni 1011
   vxlan vlan 12 vni 1012
   vxlan vlan 13 vni 1013
   vxlan vlan 20 vni 1020
   vxlan vlan 210 vni 1210
   vxlan vlan 211 vni 1211
   vxlan vlan 212 vni 1212
   vxlan vlan 213 vni 1213
   vxlan vlan 220 vni 1220
   vxlan vrf tenant-a vni 1000
   vxlan vrf tenant-b vni 1001
```

Leaf-21 VXLAN Interface Configuration

```
!
interface Loopback1
   ip address 2.2.2.2/32
!
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 10 vni 1010
   vxlan vlan 11 vni 1011
   vxlan vlan 12 vni 1012
   vxlan vlan 13 vni 1013
   vxlan vlan 21 vni 1021
   vxlan vlan 210 vni 1210
   vxlan vlan 211 vni 1211
   vxlan vlan 212 vni 1212
   vxlan vlan 213 vni 1213
   vxlan vlan 221 vni 1221
   vxlan vrf tenant-a vni 1000
   vxlan vrf tenant-b vni 1001
```



Note: This configuration uses VXLAN routing. For single-chip T2 and TH platforms, recirculation must be enabled. For R-Series platforms, the following configuration commands must be added:

`hardware tcam`

`system profile vxlan-routing`

Refer to diagrams for VLAN and SVI assignment to tenant; ***Leaf-11*** also has peering out to the border router in addition to the connected SVIs.

18.15.2.5 eBGP Underlay Configuration on the Leaf Switches

The leaf switches for the underlay network peer with each spine on the physical interface. For EVPN route advertisement, the BGP EVPN session is between loopback addresses.

In this case, the underlay is all eBGP, and peering is on the physical interfaces. The MLAG leaves also peer with each other in the underlay to retain BGP EVPN connectivity (loopback reachability) in the very unlikely case that all spine links are down. This is a failover configuration that can be implemented if there is ever the chance a leaf could be “core isolated.” The configuration can be viewed on each leaf using the command **`show running-configuration section bgp`**.

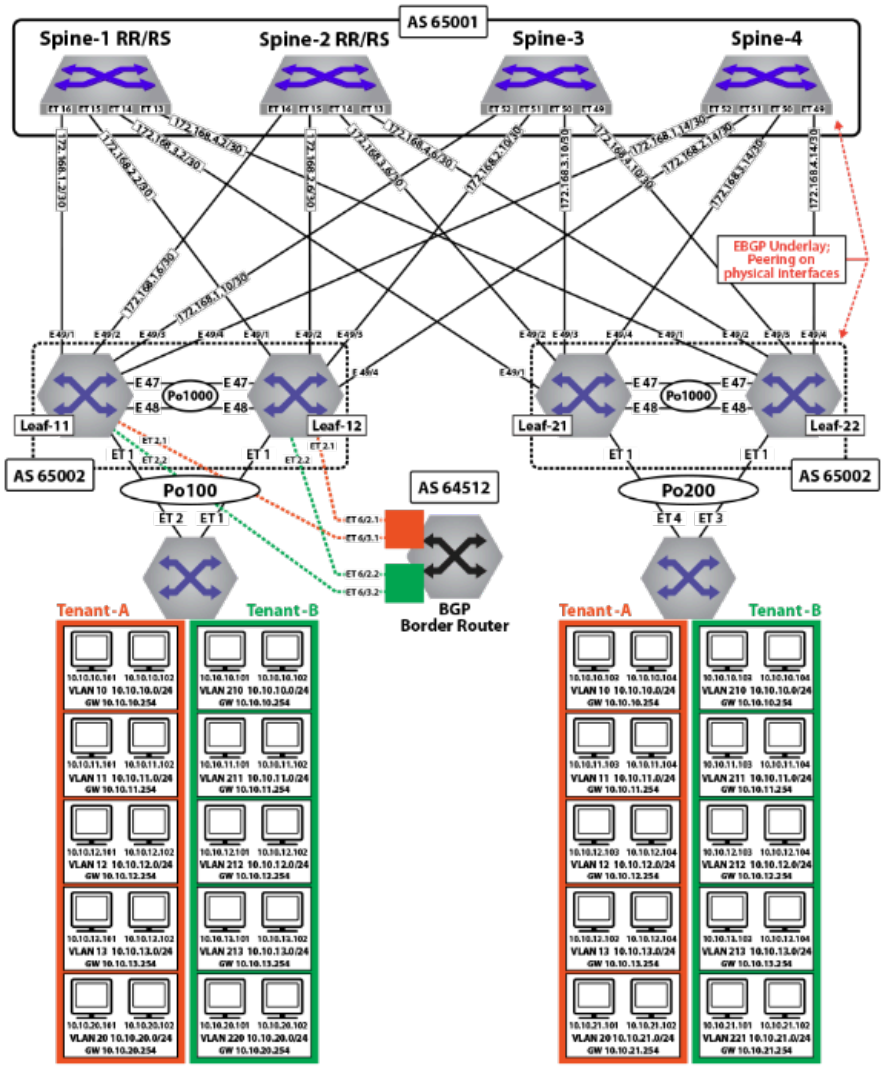
The examples below show the underlay configuration on all four leaf switches, and also on two of the spine switches as an example of the underlay configuration on the spine.

The configuration uses the following peer groups:

SPINE configuration inherited for underlay (eBGP) peering to the spines

SPINE_EVPN overlay eBGP peering between spine and leaf, using loopbacks

Figure 18-51: Physical Underlay Topology



eBGP Underlay Configuration: Leaf-11

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks deny 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks permit 20
!
ip prefix-list loopback
    seq 10 permit 1.1.1.11/32
    seq 20 permit 1.1.1.12/32
    seq 30 permit 1.1.1.22/32
    seq 40 permit 1.1.1.21/32
    seq 50 permit 2.2.2.1/32
    seq 60 permit 2.2.2.2/32
```

```
!
router bgp 65002
   router-id 1.1.1.11
   maximum-paths 8 ecmp 16
   neighbor SPINE peer-group
   neighbor SPINE remote-as 65001
   neighbor SPINE allowas-in 1
   neighbor SPINE soft-reconfiguration inbound all
   neighbor SPINE route-map loopback out
   neighbor SPINE send-community
   neighbor 172.168.1.1 peer-group SPINE
   neighbor 172.168.1.5 peer-group SPINE
   neighbor 172.168.1.9 peer-group SPINE
   neighbor 172.168.1.13 peer-group SPINE
   neighbor 172.168.11.2 remote-as 65004
   neighbor 172.168.11.2 local-as 65002 no-prepend replace-as
   neighbor 172.168.11.2 allowas-in 1
   neighbor 172.168.11.2 maximum-routes 12000
   redistribute connected route-map loopback
```

eBGP Underlay Configuration: Leaf-12

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks deny 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks permit 20
!
ip prefix-list loopback
    seq 10 permit 1.1.1.11/32
    seq 20 permit 1.1.1.12/32
    seq 30 permit 1.1.1.22/32
    seq 40 permit 1.1.1.21/32
    seq 50 permit 2.2.2.1/32
    seq 60 permit 2.2.2.2/32
!
router bgp 65002
   router-id 1.1.1.12
   maximum-paths 8 ecmp 16
   neighbor SPINE peer-group
   neighbor SPINE remote-as 65001
   neighbor SPINE allowas-in 1
   neighbor SPINE soft-reconfiguration inbound all
   neighbor SPINE route-map loopback out
   neighbor SPINE send-community
   neighbor 172.168.2.1 peer-group SPINE
   neighbor 172.168.2.5 peer-group SPINE
   neighbor 172.168.2.9 peer-group SPINE
   neighbor 172.168.2.13 peer-group SPINE
   neighbor 172.168.11.1 remote-as 65002
   neighbor 172.168.11.1 local-as 65004 no-prepend replace-as
   neighbor 172.168.11.1 allowas-in 1
   neighbor 172.168.11.1 maximum-routes 12000
   redistribute connected route-map loopback
```

eBGP Underlay Configuration: Leaf-21

```
route-map loopback permit 10
   match ip address prefix-list loopback
```

```
!
ip prefix-list loopback
    seq 10 permit 1.1.1.11/32
    seq 20 permit 1.1.1.12/32
    seq 30 permit 1.1.1.22/32
    seq 40 permit 1.1.1.21/32
    seq 50 permit 2.2.2.1/32
    seq 60 permit 2.2.2.2/32
!
router bgp 65002
   router-id 1.1.1.21
   maximum-paths 8 ecmp 16
   neighbor SPINE peer-group
   neighbor SPINE remote-as 65001
   neighbor SPINE allowas-in 1
   neighbor SPINE soft-reconfiguration inbound all
   neighbor SPINE route-map loopback out
   neighbor SPINE send-community
   neighbor SPINE maximum-routes 20000
   neighbor 172.168.3.1 peer-group SPINE
   neighbor 172.168.3.5 peer-group SPINE
   neighbor 172.168.3.9 peer-group SPINE
   neighbor 172.168.3.13 peer-group SPINE
   neighbor 172.168.11.2 remote-as 65004
   neighbor 172.168.11.2 local-as 65002 no-prepend replace-as
   neighbor 172.168.11.2 allowas-in 1
   neighbor 172.168.11.2 maximum-routes 12000
   redistribute connected route-map loopback
```

eBGP Underlay Configuration: Leaf-22

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
ip prefix-list loopback
    seq 10 permit 1.1.1.11/32
    seq 20 permit 1.1.1.12/32
    seq 30 permit 1.1.1.22/32
    seq 40 permit 1.1.1.21/32
    seq 50 permit 2.2.2.1/32
    seq 60 permit 2.2.2.2/32
!
router bgp 65002
   router-id 1.1.1.22
   maximum-paths 8 ecmp 16
   neighbor SPINE peer-group
   neighbor SPINE remote-as 65001
   neighbor SPINE allowas-in 1
   neighbor SPINE soft-reconfiguration inbound all
   neighbor SPINE route-map loopback out
   neighbor SPINE send-community
   neighbor SPINE maximum-routes 20000
   neighbor 172.168.4.1 peer-group SPINE
   neighbor 172.168.4.5 peer-group SPINE
   neighbor 172.168.4.9 peer-group SPINE
   neighbor 172.168.4.13 peer-group SPINE
   neighbor 172.168.11.1 remote-as 65002
   neighbor 172.168.11.1 local-as 65004 no-prepend replace-as
   neighbor 172.168.11.2 allowas-in 1
   neighbor 172.168.11.1 maximum-routes 12000
   redistribute connected route-map loopback
```

18.15.2.6 EVPN BGP Configuration on the Spine Switches

The EVPN BGP configuration on two of the spine switches is summarized below. Note that only the EVPN BGP sessions are listed for the two spine switches: the BGP underlay configuration is not included.

EVPN BGP Configuration: Spine-1

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
ip prefix-list loopback
    seq 10 permit 1.1.1.11/32
    seq 20 permit 1.1.1.12/32
    seq 30 permit 1.1.1.22/32
    seq 40 permit 1.1.1.21/32
    seq 50 permit 2.2.2.1/32
    seq 60 permit 2.2.2.2/32
!
router bgp 65001
   router-id 1.1.1.1
   distance bgp 20 200 200
   maximum-paths 8 ecmp 16
   neighbor LEAF peer-group
   neighbor LEAF remote-as 65002
   neighbor LEAF maximum-routes 20000
   neighbor 172.168.1.2 peer-group LEAF
   neighbor 172.168.2.2 peer-group LEAF
   neighbor 172.168.3.2 peer-group LEAF
   neighbor 172.168.4.2 peer-group LEAF
   redistribute connected route-map loopback
```

EVPN BGP Configuration: Spine-2

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
ip prefix-list loopback
    seq 10 permit 1.1.1.11/32
    seq 20 permit 1.1.1.12/32
    seq 30 permit 1.1.1.22/32
    seq 40 permit 1.1.1.21/32
    seq 50 permit 2.2.2.1/32
    seq 60 permit 2.2.2.2/32
!
router bgp 65001
   router-id 1.1.1.2
   distance bgp 20 200 200
   maximum-paths 8 ecmp 16
   neighbor LEAF peer-group
   neighbor LEAF remote-as 65002
   neighbor LEAF maximum-routes 20000
   neighbor 172.168.1.6 peer-group LEAF
   neighbor 172.168.2.6 peer-group LEAF
   neighbor 172.168.3.6 peer-group LEAF
   neighbor 172.168.4.6 peer-group LEAF
   redistribute connected route-map loopback
```

18.15.2.7 eBGP Overlay on Leaf Switches

The MAC VRFs and IP VRF for the tenants' subnets are created in the BGP router context with unique Route-Distinguishers (RD) and Route-Targets (RT) attached to each MAC-VRF and IP-VRF. The RDs provide support for overlapping MAC and IP addresses across tenants, while the RTs allow control of the routes imported and exported between MAC VRFs.

To ensure all routes are correctly imported between VTEPs sharing the same Layer-2 domain, the import and export RTs are equal across the two MLAG domains. The **`redistribute learned`** statement under each MAC VRF ensures any locally learned MACs in the VLAN are automatically announced as type-2 routes.

The IP VRF (***Tenant-A***) is created on all leaf switches which have subnets attached to the tenant's VRF with the same route target ensuring that routes are correctly imported and exported between VTEPs in the VRF. On ***Leaf-21*** and ***Leaf-22***, to import the external routes an eBGP session with the BGP peering router is created under the IP VRF (***Tenant-A***) context, and a peering from each to the other is created on the overlay.



Note: All MAC VRFs are unique, and each has its own RT, matched by the other leaves in the DC. The "tenants" as such are defined at layer 3 by assigning SVIs to the appropriate VRF. To view this assignment, use the **`show ip route vrf <tenant> connected`** command. Note below that VLANs ***12***-***13*** and ***212***-***213*** (shown in bold) are configured as a bundle-aware EVPN service. Also note the peering from ***Leaf-11*** to the BGP border router in each tenant VRF.

EVPN BGP Overlay Configuration for the Tenants' MAC VRFs and IP VRF: Leaf-11

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks deny 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks permit 20
!
ip prefix-list loopback
    seq 10 permit 1.1.1.11/32
    seq 20 permit 1.1.1.12/32
    seq 30 permit 1.1.1.22/32
    seq 40 permit 1.1.1.21/32
    seq 50 permit 2.2.2.1/32
    seq 60 permit 2.2.2.2/32
!
router bgp 65002
   router-id 1.1.1.11
   maximum-paths 4
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN allowas-in 2
   neighbor SPINE_EVPN ebgp-multihop 5
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   neighbor 1.1.1.2 peer-group SPINE_EVPN
   redistribute connected route-map loopback
   !
  vlan 10
      rd 1.1.1.11:1010
      route-target both 1010:1010
      redistribute learned
   !
   vlan 11
      rd 1.1.1.11:1011
      route-target both 1011:1011
```

```
      redistribute learned
   !
   vlan 20
      rd 1.1.1.11:1020
      route-target both 1020:1020
      redistribute learned
   !
   vlan 210
      rd 1.1.1.11:1210
      route-target both 1210:1210
      redistribute learned
      no redistribute host-route
   !
   vlan 211
      rd 1.1.1.11:1211
      route-target both 1211:1211
      redistribute learned
      no redistribute host-route
   !
   vlan 220
      rd 1.1.1.11:1220
      route-target both 1220:1220
      redistribute learned
      no redistribute host-route
   !
  vlan-aware-bundle Tenant-A-VLAN-12-13
      rd 1.1.1.11:1213
      route-target both 12:13
      redistribute learned
      vlan 12-13
   !
   vlan-aware-bundle Tenant-B-VLAN-212-213
       rd 1.1.1.11:21213
       route-target both 212:213
       redistribute learned
       no redistribute host-route
       vlan 212-213
   !
   address-family evpn
      neighbor SPINE_EVPN activate
   !
   address-family ipv4
      no neighbor SPINE_EVPN activate
   !
   vrf tenant-a
      rd 1.1.1.11:1000
      route-target import 1000:1000
      route-target export 1000:1000
      neighbor 192.168.168.9 remote-as 64512
      neighbor 192.168.168.9 local-as 65002 no-prepend replace-as
      neighbor 192.168.168.9 maximum-routes 12000
      neighbor 223.255.255.250 peer-group LEAF_PEER_OVERLAY
      neighbor 223.255.255.250 remote-as 65004
      neighbor 223.255.255.250 local-as 65002 no-prepend replace-as
      redistribute connected route-map dont_advertise_loopbacks
   !
   vrf tenant-b
      rd 1.1.1.11:1001
      route-target import 1001:1001
      route-target export 1001:1001
      neighbor 192.168.168.21 remote-as 64513
      neighbor 192.168.168.21 local-as 65002 no-prepend replace-as
      neighbor 192.168.168.21 maximum-routes 12000
```

```
      neighbor 223.255.255.249 peer-group LEAF_PEER_OVERLAY
      neighbor 223.255.255.249 remote-as 65004
      neighbor 223.255.255.249 local-as 65002 no-prepend replace-as
      redistribute connected route-map dont_advertise_loopbacks
```

EVPN BGP Overlay Configuration for the Tenants' MAC VRFs and IP VRF: Leaf-12

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks deny 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks permit 20
!
ip prefix-list loopback
    seq 10 permit 1.1.1.11/32
    seq 20 permit 1.1.1.12/32
    seq 30 permit 1.1.1.22/32
    seq 40 permit 1.1.1.21/32
    seq 50 permit 2.2.2.1/32
    seq 60 permit 2.2.2.2/32
!
router bgp 65002
   router-id 1.1.1.12
   maximum-paths 4
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN allowas-in 2
   neighbor SPINE_EVPN ebgp-multihop 5
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   neighbor 1.1.1.2 peer-group SPINE_EVPN
   redistribute connected route-map loopback
!
  vlan 10
      rd 1.1.1.12:1010
      route-target both 1010:1010
      redistribute learned
   !
   vlan 11
      rd 1.1.1.12:1011
      route-target both 1011:1011
      redistribute learned
   !
   vlan 20
      rd 1.1.1.12:1020
      route-target both 1020:1020
      redistribute learned
   !
   vlan 210
      rd 1.1.1.12:1210
      route-target both 1210:1210
      redistribute learned
      no redistribute host-route
   !
   vlan 211
      rd 1.1.1.12:1211
      route-target both 1211:1211
      redistribute learned
```

```
      no redistribute host-route
   !
   vlan 220
      rd 1.1.1.12:1220
      route-target both 1220:1220
      redistribute learned
      no redistribute host-route
   !
   vlan-aware-bundle Tenant-A-VLAN-12-13
      rd 1.1.1.12:1213
      route-target both 12:13
      redistribute learned
      vlan 12-13
   !
   vlan-aware-bundle Tenant-B-VLAN-212-213
      rd 1.1.1.12:21213
      route-target both 212:213
      redistribute learned
      no redistribute host-route
      vlan 212-213
   !
   address-family evpn
      neighbor SPINE_EVPN activate
   !
   address-family ipv4
      no neighbor SPINE_EVPN activate
   !
   vrf tenant-a
      rd 1.1.1.12:1000
      route-target import 1000:1000
      route-target export 1000:1000
      neighbor 192.168.168.13 remote-as 64512
      neighbor 192.168.168.13 local-as 65002 no-prepend replace-as
      neighbor 192.168.168.13 maximum-routes 12000
      neighbor 223.255.255.249 peer-group LEAF_PEER_OVERLAY
      neighbor 223.255.255.249 remote-as 65002
      neighbor 223.255.255.249 local-as 65004 no-prepend replace-as
      redistribute connected route-map dont_advertise_loopbacks
   !
   vrf tenant-b
      rd 1.1.1.12:1001
      route-target import 1001:1001
      route-target export 1001:1001
      neighbor 192.168.168.23 remote-as 64513
      neighbor 192.168.168.23 local-as 65002 no-prepend replace-as
      neighbor 192.168.168.23 maximum-routes 12000
      neighbor 223.255.255.249 peer-group LEAF_PEER_OVERLAY
      neighbor 223.255.255.249 remote-as 65002
      neighbor 223.255.255.249 local-as 65004 no-prepend replace-as
      redistribute connected route-map dont_advertise_loopbacks
```

EVPN BGP Overlay Configuration for the Tenants' MAC VRFs and IP VRF: Leaf-21

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks deny 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks permit 20
!
router bgp 65002
```

```
   router-id 1.1.1.21
   maximum-paths 4
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN allowas-in 2
   neighbor SPINE_EVPN ebgp-multihop 5
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   neighbor 1.1.1.2 peer-group SPINE_EVPN
   redistribute connected route-map loopback
   !
   vlan 10
      rd 1.1.1.21:1010
      route-target both 1010:1010
      redistribute learned
   !
   vlan 11
      rd 1.1.1.21:1011
      route-target both 1011:1011
      redistribute learned
   !
   vlan 21
      rd 1.1.1.21:1021
      route-target both 1021:1021
      redistribute learned
   !
   vlan 210
      rd 1.1.1.21:1210
      route-target both 1210:1210
      redistribute learned
      no redistribute host-route
   !
   vlan 211
      rd 1.1.1.21:1211
      route-target both 1211:1211
      redistribute learned
      no redistribute host-route
   !
   vlan 221
      rd 1.1.1.21:1221
      route-target both 1221:1221
      redistribute learned
      no redistribute host-route
   !
   vlan-aware-bundle Tenant-A-VLAN-12-13
      rd 1.1.1.21:1213
      route-target both 12:13
      redistribute learned
      vlan 12-13
   !
   vlan-aware-bundle Tenant-B-VLAN-212-213
      rd 1.1.1.21:21213
      route-target both 212:213
      redistribute learned
      redistribute host-route
      vlan 212-213
   !
   address-family evpn
      neighbor SPINE_EVPN activate
   !
   address-family ipv4
```

```
      no neighbor SPINE_EVPN activate
   !
   vrf tenant-a
      rd 1.1.1.21:1000
      route-target import 1000:1000
      route-target export 1000:1000
      neighbor 223.255.255.254 remote-as 65002
      neighbor 223.255.255.254 next-hop-self
      neighbor 223.255.255.254 update-source Vlan1111
      neighbor 223.255.255.254 allowas-in 1
      neighbor 223.255.255.254 maximum-routes 12000
      redistribute connected route-map dont_advertise_loopbacks
   !
   vrf tenant-b
      rd 1.1.1.21:1001
      route-target import 1001:1001
      route-target export 1001:1001
      neighbor 223.255.255.254 remote-as 65002
      neighbor 223.255.255.254 next-hop-self
      neighbor 223.255.255.254 update-source Vlan2111
      neighbor 223.255.255.254 allowas-in 1
      neighbor 223.255.255.254 maximum-routes 12000
      redistribute connected route-map dont_advertise_loopbacks
```

EVPN BGP Overlay Configuration for the Tenants' MAC VRFs and IP VRF: Leaf-22

```
route-map loopback permit 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks deny 10
   match ip address prefix-list loopback
!
route-map dont_advertise_loopbacks permit 20
!
router bgp 65002
   router-id 1.1.1.22
   maximum-paths 4
   neighbor SPINE_EVPN peer-group
   neighbor SPINE_EVPN remote-as 65001
   neighbor SPINE_EVPN update-source Loopback0
   neighbor SPINE_EVPN allowas-in 2
   neighbor SPINE_EVPN ebgp-multihop 5
   neighbor SPINE_EVPN send-community extended
   neighbor SPINE_EVPN maximum-routes 12000
   neighbor 1.1.1.1 peer-group SPINE_EVPN
   neighbor 1.1.1.2 peer-group SPINE_EVPN
   redistribute connected route-map loopback
   !
   vlan 10
      rd 1.1.1.22:1010
      route-target both 1010:1010
      redistribute learned
   !
   vlan 11
      rd 1.1.1.22:1011
      route-target both 1011:1011
      redistribute learned
   !
   vlan 21
      rd 1.1.1.22:1021
      route-target both 1021:1021
      redistribute learned
```

```
!
vlan 210
   rd 1.1.1.22:1210
   route-target both 1210:1210
   redistribute learned
   no redistribute host-route
!
vlan 211
   rd 1.1.1.22:1211
   route-target both 1211:1211
   redistribute learned
   no redistribute host-route
!
vlan 221
   rd 1.1.1.22:1221
   route-target both 1221:1221
   redistribute learned
   no redistribute host-route
!
vlan-aware-bundle Tenant-A-VLAN-12-13
   rd 1.1.1.22:1213
   route-target both 12:13
   redistribute learned
   vlan 12-13
!
vlan-aware-bundle Tenant-B-VLAN-212-213
   rd 1.1.1.22:21213
   route-target both 212:213
   redistribute learned
   no redistribute host-route
   vlan 212-213
!
address-family evpn
   neighbor SPINE_EVPN activate
!
address-family ipv4
   no neighbor SPINE_EVPN activate
!
vrf tenant-a
   rd 1.1.1.22:1000
   route-target import 1000:1000
   route-target export 1000:1000
   neighbor 223.255.255.253 remote-as 65002
   neighbor 223.255.255.253 next-hop-self
   neighbor 223.255.255.253 update-source Vlan1111
   neighbor 223.255.255.253 allowas-in 1
   neighbor 223.255.255.253 maximum-routes 12000
   redistribute connected route-map dont_advertise_loopbacks
!
vrf tenant-b
   rd 1.1.1.22:1001
   route-target import 1001:1001
   route-target export 1001:1001
   neighbor 223.255.255.253 remote-as 65002
   neighbor 223.255.255.253 next-hop-self
   neighbor 223.255.255.253 update-source Vlan2111
   neighbor 223.255.255.253 allowas-in 1
   neighbor 223.255.255.253 maximum-routes 12000
   redistribute connected route-map dont_advertise_loopbacks
```

18.15.2.8 eBGP Overlay on Spine Switches

The EVPN BGP configuration on the spine switches is summarised in the examples below. Note that only the EVPN BGP sessions are listed for two spine switches; the BGP underlay configuration is not included.

EVPN BGP Overlay Configuration: Spine-1

```
!
router bgp 65001
   router-id 1.1.1.1
   distance bgp 20 200 200
   maximum-paths 8 ecmp 16
   neighbor LEAF_EVPN peer-group
   neighbor LEAF_EVPN remote-as 65002
   neighbor LEAF_EVPN update-source Loopback0
   neighbor LEAF_EVPN ebgp-multihop 5
   neighbor LEAF_EVPN send-community extended
   neighbor LEAF_EVPN next-hop-unchanged
   neighbor LEAF_EVPN maximum-routes 12000
   neighbor 1.1.1.11 peer-group LEAF_EVPN
   neighbor 1.1.1.12 peer-group LEAF_EVPN
   neighbor 1.1.1.21 peer-group LEAF_EVPN
   neighbor 1.1.1.22 peer-group LEAF_EVPN
   !
   address-family evpn
      neighbor LEAF_EVPN activate
   !
   address-family ipv4
      no neighbor LEAF_EVPN activate
!
   address-family ipv6
      no neighbor LEAF_EVPN activate
!
```

EVPN BGP Overlay Configuration: Spine-2

```
!
router bgp 65001
   router-id 1.1.1.2
   distance bgp 20 200 200
   maximum-paths 8 ecmp 16
   neighbor LEAF_EVPN peer-group
   neighbor LEAF_EVPN remote-as 65002
   neighbor LEAF_EVPN update-source Loopback0
   neighbor LEAF_EVPN ebgp-multihop 5
   neighbor LEAF_EVPN send-community extended
   neighbor LEAF_EVPN next-hop-unchanged
   neighbor LEAF_EVPN maximum-routes 12000
   neighbor 1.1.1.11 peer-group LEAF_EVPN
   neighbor 1.1.1.12 peer-group LEAF_EVPN
   neighbor 1.1.1.21 peer-group LEAF_EVPN
   neighbor 1.1.1.21 peer-group LEAF_EVPN
   !
   address-family evpn
      neighbor LEAF_EVPN activate
   !
   address-family ipv4
      no neighbor LEAF_EVPN activate
!
   address-family ipv6
      no neighbor LEAF_EVPN activate
```

```
!
```

18.15.2.9 Symmetric IRB Configuration (Tenant-A)

In symmetric IRB, the host routes are generated by advertising type-2 routes with both the MAC VRF VNI and the routing (or VRF) VNI. On ***Leaf-11***, the MAC VRFs for ***Tenant-A*** are left in their default configuration (i.e., redistributing host routes). The example below shows the configuration for the MAC VRF.

MAC VRF Configuration for Tenant-A: Leaf-11

The **`redistribute learned`** commands below cause type-2 routes to be advertised with two labels: in ***VLAN 10***, ***1010*** and ***1000***; in ***VLAN 11***, 1011 and ***1000***; in ***VLAN 21***, ***1021*** and ***1000***.

```
vlan 10
      rd 1.1.1.11:1010
      route-target both 1010:1010
      redistribute learned
!
   vlan 11
      rd 1.1.1.11:1011
      route-target both 1011:1011
      redistribute learned
!
   vlan 21
      rd 1.1.1.11:1021
      route-target both 1021:1021
      redistribute learned
   !
```

With this configuration, any locally learned MAC-IP binding on a leaf switch will be advertised as a type-2 route with two labels. For example, on switches ***Leaf-21*** and ***Leaf-22***, any MAC-IP binding locally learned on subnets ***10.10.10.0/24***, ***10.10.11.0/24***, or ***10.10.21.0/24*** will be advertised as type-2 routes with two labels (the MAC VRF of ***1010***, ***1011***, or ***1021*** and the IP VRF of ***1000***) and two route targets equal to the relevant MAC VRF for the host and IP VRF for the tenant (***1000:1000***). The remote leaf switches (***Leaf-11*** and ***Leaf-12***), will now learn the host route in the IP VRF.

In addition to advertising the type-2 routes with dual labels, the switch will still advertise type-5 routes. This ensures connectivity to the remote subnet even when no host on the subnet has been learned. With both a layer-2 route and layer-3 host route for Server-3 learned on the MAC VRF(***1010***) and the IP VRF (***1000***) on ***Leaf-11***, traffic ingressing on ***Leaf-11*** from the local subnet ***10.10.10.103*** (i.e., ***VLAN 10***) will be VXLAN bridged based on the MAC VRF entry. Traffic ingressing from outside the subnet (i.e., ***VLAN 11***, ***12***, ***13***, or ***20***) will be routed to the host via the IP VRF host route.

The VLAN-aware bundle VLAN type-2 routes are advertised with the VNI ID within the update.

The type-5 routes are advertised with the IP VRF Route Distinguisher and the VNI label, signifying that the forwarding path for the prefix would be the IP VRF. The imported routes from the eBGP peering with the BGP border router in ***Leaf-11*** and ***Leaf-12*** are imported by both switches respectively and redistributed via type-5 advertisements to ***Leaf-21*** and ***Leaf-22***.

18.15.2.10 Asymmetric IRB Configuration (Tenant-B)

In asymmetric IRB, the host routes are generated by advertising type-2 routes with just the MAC VRF VNI. On leaf 11, the MAC VRFs for ***Tenant-B*** are configured with no redistribute host route within the MAC VRF configuration. The example below shows the configuration for the MAC VRF.

MAC VRF Configuration for Tenant-B: Leaf-11

The **`no redistribute host-route`** commands below cause type-2 routes to be advertised with a single label: in ***VLAN 210***, ***1110***; in ***VLAN 211***, ***1211***; in ***VLAN 220***, ***1220***; and in the VLAN-aware bundle (***Tenant-B-VLAN-212-213***), ***1212*** and ***1213***.

```
vlan 210
      rd 1.1.1.11:1210
      route-target both 1210:1210
      redistribute learned
      no redistribute host-route
   !
   vlan 211
      rd 1.1.1.11:1211
      route-target both 1211:1211
      redistribute learned
      no redistribute host-route
   !
   vlan 220
      rd 1.1.1.11:1220
      route-target both 1220:1220
      redistribute learned
      no redistribute host-route
!
     vlan-aware-bundle Tenant-B-VLAN-212-213
      rd 1.1.1.11:21213
      route-target both 212:213
      redistribute learned
      no redistribute host-route
      vlan 212-213
   !
```

With this configuration, any locally learned MAC-IP binding on a leaf switch will be advertised as a type-2 route with a single label. For example, on ***Leaf-11*** and ***Leaf-12***, any MAC-IP binding locally learned on subnets ***10.10.10.0/24***, ***10.10.11.0/24***, or ***10.10.21.0/24*** will be advertised as type-2 routes with a single label, the MAC VRF (***1210***, ***1211***, ***1220***, ***1212***, ***1213***, or ***21111***). The IP VRF (***1001***) still advertises the type-5 prefix routes. This ensures connectivity to the remote subnet even when no host on the subnet has been learned.

The VLAN-aware bundle VLAN type-2 routes are advertised with the VNI ID within the update.

18.15.3 EVPN MPLS Sample Configuration

This section describes configuring and verifying BGP VPN which has steps similar to the EVPN VXLAN demonstration. Here, we examine BGP EVPN layer 3 VPN over LDP, Segment Routing (ISIS-SR), and BGP-SR transport LSPs. This highlights the difference between the transport and the VPN overlay service.

18.15.3.1 Layer 3 VPN Over ISIS-SR

The following figures illustrate the overview of combined control and data planes.

Figure 18-52: Control Plane Tenant-A Over ISIS-SR

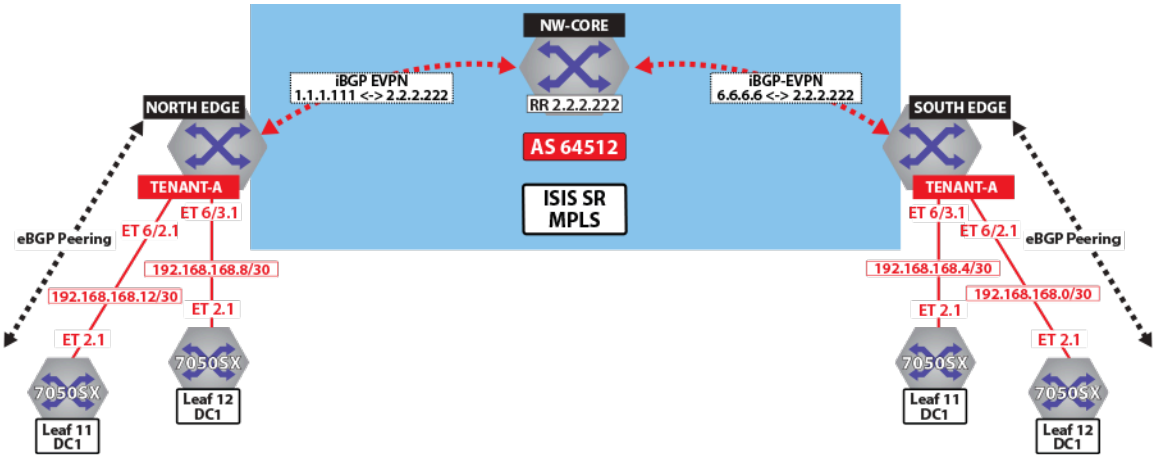


Figure 18-53: Control Plane Tenant-B over ISIS-SR

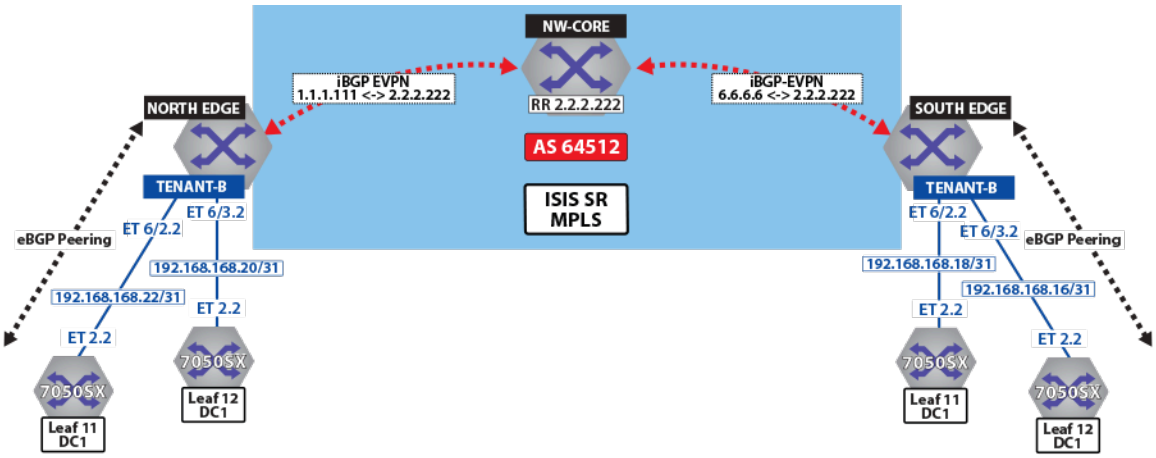
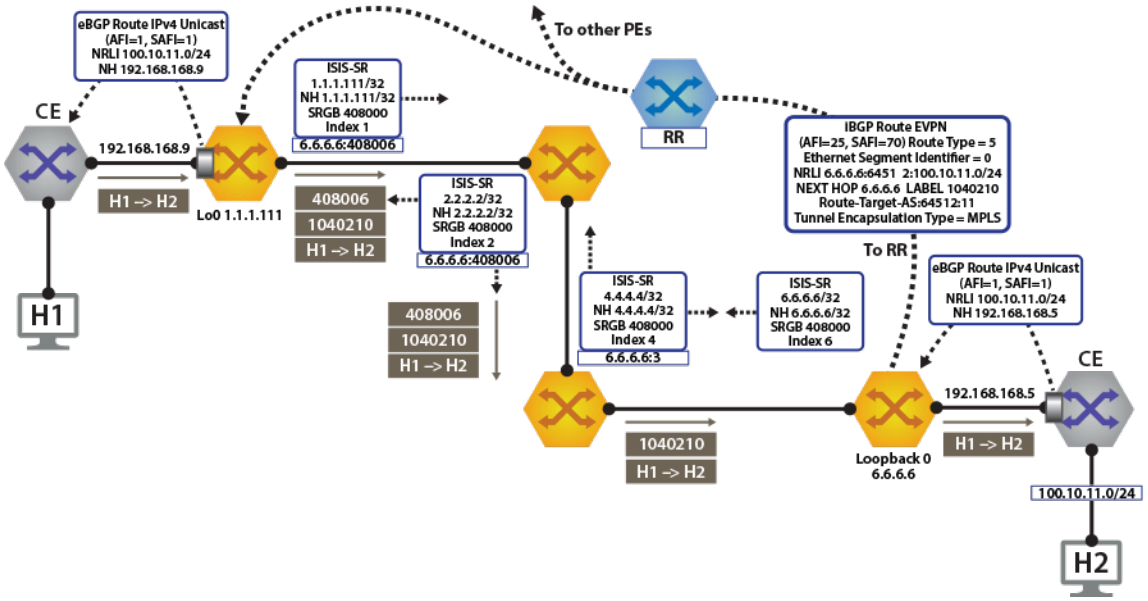


Figure 18-54: Control Plane & Forwarding Tenant-A Over ISIS-SR



The North Edge router has an eBGP peering session out to ***Leaf-11*** and ***Leaf-12*** in ***DC1***, while the South Edge router has peerings to ***Leaf-11*** and***Leaf-12*** in ***DC2***. Tenant-a has few additional local interfaces used for testing.

Example

The **`show ip route vrf tenant-a connected`** command displays the interfaces assigned to the tenant-a of North Edge router.

```
north-edge# show ip route vrf tenant-a connected

VRF: tenant-a
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

 C      192.168.168.8/30 is directly connected, Ethernet6/3.1
 C      192.168.168.12/30 is directly connected, Ethernet6/2.1
```

Activating EVPN

In all scenarios, the EVPN must be activated under BGP and neighbors configured to exchange Layer 2 VPN/EVPN NLRI.The tenant's VRF (tenant-a and tenant-b) is associated with a dynamically assigned label by BGP.

An activated EVPN provides the following functionalities:

- Enables the multi-agent routing protocol model, which is required for EVPN support.
- Sets the local autonomous system number to 64512 and configures IBGP neighbors that are activated for the Layer 2 VPN/EVPN address family.
- Sets the EVPN encapsulation type to MPLS.

- Specifies that Loopback0 will be used as the next-hop for all advertised EVPN routes. The underlay configuration must provide MPLS LSPs from remote PEs to this loopback interface address.

Example

The **`service routing protocols model multi-agent`** command activates EVPN on the north edge router.

```
service routing protocols model multi-agent

router bgp 64512
   router-id 1.1.1.111
   maximum-paths 128 ecmp 128
   neighbor 2.2.2.222 remote-as 64512
   neighbor 2.2.2.222 update-source Loopback0
   neighbor 2.2.2.222 bfd
   neighbor 2.2.2.222 send-community extended
   !
   address-family evpn
     neighbor default encapsulation mpls next-hop-self source-interface
 Loopback0
       neighbor default graceful-restart
       neighbor 2.2.2.222 activate
   !
```

Layer 3 Overlay Configuration

Distribution of layer 3 routes over BGP is enabled by configuring one or more IP VRFs under the router bgp configuration mode. Additionally, IP routing must be enabled in the VRF.

The VRF is assigned a unique Route-Distinguisher (RD). The RD allows the PE to advertise EVPN routes for the same IP prefix that have been exported by different VRFs. The NLRI RouteKey of a route exported from the VRF's IPv4 table into EVPN consists of both the RD and the original IP prefix.

The Route-Target (RT) extended communities for the VRF. The RTs are associated with all routes exported from the VRF. Received EVPN type-5 routes carrying at least one RT matching the VRFs configuration are imported into the VRF. The route target directives are configured under the IPv4 or IPv6 address- family.

Example

The **`vrf tenant-a`** and **`vrf tenant-a`** commands define overlay VRFs (***tenant-a*** and ***tenant-b***) on the VTEP of North Edge router and enables IPv4 routing within them.

```
vrf tenant-a
      rd 1.1.1.1:64512
      route-target import evpn 64512:11
      route-target export evpn 64512:11
      router-id 1.1.1.111
      neighbor 192.168.168.10 remote-as 65002
      neighbor 192.168.168.10 local-as 64512 no-prepend replace-as
      neighbor 192.168.168.10 default-originate
      neighbor 192.168.168.10 maximum-routes 12000
      neighbor 192.168.168.14 remote-as 65002
      neighbor 192.168.168.14 local-as 64512 no-prepend replace-as
      neighbor 192.168.168.14 default-originate
      neighbor 192.168.168.14 maximum-routes 12000
      redistribute connected
      redistribute static
   !
   vrf tenant-b
      rd 1.1.1.1:64513
      route-target import evpn 64513:11
      route-target export evpn 64513:11
```

```
    router-id 1.1.1.111
    neighbor 192.168.168.20 remote-as 65002
    neighbor 192.168.168.20 local-as 64513 no-prepend replace-as
    neighbor 192.168.168.20 maximum-routes 12000
    neighbor 192.168.168.22 remote-as 65002
    neighbor 192.168.168.22 local-as 64513 no-prepend replace-as
    neighbor 192.168.168.22 maximum-routes 12000
    redistribute connected
    redistribute static
!
```

Verifying BGP EVPN Layer 3 VPN

Show commands are executed in the North Edge router to view routes to the South Edge router. Execute the same commands in the South Edge router to view vice-versa routes.

Examples

- The **`show bgp evpn summary`** command displays the status of EVPN peers in North Edge router.

```
north-edge# show bgp evpn summary
BGP summary information for VRF default
Router identifier 1.1.1.111, local AS number 64512
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State
PfxRcd PfxAcc
  2.2.2.222        4  64512            195       127    0    0 01:13:31 Estab  78     78
```

- The **`show bgp evpn route-type ip-prefix ipv4 next-hop 6.6.6.6`** command displays all BGP EVPN ip prefix routes received from the South Edge router (***6.6.6.6***). Not all are advertised via the ***RR 2.2.2.222***.



Note: Each entry in the table represents a BGP path. The path specific information includes Route-Distinguisher and IP prefix. Paths are either received from EVPN peers or exported from local VRFs.

```
north-edge# show bgp evpn route-type ip-prefix ipv4 next-hop 6.6.6.6
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP
 head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local
 Nexthop

         Network             Next Hop         Metric  LocPref Weight Path
 * >     RD: 6.6.6.6:64512 ip-prefix 0.0.0.0/0
                             6.6.6.6          0       100     0       ? Or-ID: 6.6.6.6 C-
LST: 2.2.2.222
 * >     RD: 6.6.6.6:64513 ip-prefix 0.0.0.0/0
                             6.6.6.6          0       100     0       ? Or-ID: 6.6.6.6 C-
LST: 2.2.2.222
 * >     RD: 6.6.6.6:64514 ip-prefix 10.255.255.0/30
                             6.6.6.6          -       100     0      65010 i Or-ID:
 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 6.6.6.6:64512 ip-prefix 100.10.10.0/24
                             6.6.6.6          -       100     0      65006 i Or-ID:
 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 6.6.6.6:64513 ip-prefix 100.10.10.0/24
                             6.6.6.6          -       100     0      65006 i Or-ID:
 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 6.6.6.6:64512 ip-prefix 100.10.10.103/32
                             6.6.6.6          -       100     0      65006 65005 65006 i
 Or-ID: 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 6.6.6.6:64512 ip-prefix 100.10.10.104/32
                             6.6.6.6          -       100     0      65006 65005 65006 i
 Or-ID: 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 6.6.6.6:64512 ip-prefix 100.10.11.0/24
                             6.6.6.6          -       100     0      65006 i Or-ID:
 6.6.6.6 C-LST: 2.2.2.222
```

```
 * >     RD: 6.6.6.6:64513 ip-prefix 100.10.11.0/24
                            6.6.6.6          -       100     0       65006 i Or-ID:
 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 6.6.6.6:64512 ip-prefix 100.10.11.103/32
                            6.6.6.6          -       100     0       65006 65005 65006 i
 Or-ID: 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 6.6.6.6:64512 ip-prefix 100.10.11.104/32
                            6.6.6.6          -       100     0       65006 65005 65006 i
 Or-ID: 6.6.6.6 C-LST: 2.2.2.222
```

- The **`show bgp evpn route-type ip-prefix 100.10.11.0/24 detail`** command displays a detailed view of the IP prefix route for ***100.10.11.0/24***. The output again includes the RD and IP prefix identifying the route. As seen above the route is received from the route reflector, and the VPN label for ***tenant-a*** is ***958810***.

```
north-edge# show bgp evpn route-type ip-prefix 100.10.11.0/24 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for ip-prefix 100.10.11.0/24, Route Distinguisher:
 6.6.6.6:64512
 Paths: 1 available
  65006
    6.6.6.6 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:11 TunnelEncap:t
unnelTypeMpls
      MPLS label: 958810
BGP routing table entry for ip-prefix 100.10.11.0/24, Route Distinguisher:
 6.6.6.6:64513
 Paths: 1 available
  65006
    6.6.6.6 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64513:11 TunnelEncap:t
unnelTypeMpls
      MPLS label: 953372
```



Note: ***Tenant-a*** and ***tenant-b*** share the same route. Therefore, both route with ***RD 6.6.6.6:64513*** and ***RT 64513:11***.

- The **`show ip bgp vrf tenant-a`** command displays the BGP table for VRF in tenant-a containing imported EVPN routes. Each entry in the table represent a BGP path that is either locally redistributed / received into the VRF or imported from the EVPN table.

```
north-edge# show ip bgp vrf tenant-a
BGP routing table information for VRF tenant-a
Router identifier 1.1.1.111, local AS number 64512
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e
 - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local
 Nexthop

         Network             Next Hop         Metric  LocPref Weight Path
 * >     0.0.0.0/0           6.6.6.6          0       100     0      ? Or-ID: 6.6.6.6 C-LST:
 2.2.2.222
 * >Ec   10.10.10.0/24       192.168.168.14   -       100     0      65002 i
 *  ec   10.10.10.0/24       192.168.168.10   -       100     0      65002 i
 * >Ec   10.10.10.103/32     192.168.168.14   -       100     0      65002 i
 *  ec   10.10.10.103/32     192.168.168.10   -       100     0      65002 i
 * >Ec   10.10.10.104/32     192.168.168.14   -       100     0      65002 i

 * >Ec   10.10.44.1/32       192.168.168.14   -       100     0      65002 i
 *  ec   10.10.44.1/32       192.168.168.10   -       100     0      65002 i
 * >     100.10.10.0/24      6.6.6.6          -       100     0      65006 i Or-ID: 6.6.6.6 C-
LST: 2.2.2.222
 * >     100.10.10.103/32    6.6.6.6          -       100     0      65006 65005 65006 i Or-ID:
 6.6.6.6
```

```
C-LST: 2.2.2.222
 * >     100.10.10.104/32    6.6.6.6          -       100     0      65006 65005 65006 i Or-ID:
 6.6.6.6
C-LST: 2.2.2.222
C-LST: 2.2.2.222
 * >     100.10.21.102/32    6.6.6.6          -       100     0      65006 65005 65006 i Or-ID:
 6.6.6.6
C-LST: 2.2.2.222
 * >     100.10.30.0/24      6.6.6.6          -       100     0      65006 i Or-ID: 6.6.6.6 C-
LST: 2.2.2.222
 * >     100.10.32.0/24      6.6.6.6          -       100     0      65006 i Or-ID: 6.6.6.6 C-
LST: 2.2.2.222
 * >     192.168.168.0/30    6.6.6.6          -       100     0      i Or-ID: 6.6.6.6 C-LST:
 2.2.2.222
 * >     192.168.168.4/30    6.6.6.6          -       100     0      i Or-ID: 6.6.6.6 C-LST:
 2.2.2.222
 * >     192.168.168.8/30    -                -       -       0      i
 *  Ec   192.168.168.8/30    192.168.168.14   -       100     0      65002 i
 *  ec   192.168.168.8/30    192.168.168.10   -       100     0      65002 i
 * >     192.168.168.12/30   -                -       -       0      i
 *  Ec   192.168.168.12/30   192.168.168.14   -       100     0      65002 i
 *  ec   192.168.168.12/30   192.168.168.10   -       100     0      65002 i
 * >     223.255.254.248/30  6.6.6.6          -       100     0      65006 i Or-ID: 6.6.6.6 C-
LST: 2.2.2.222
 * >     223.255.254.252/30  6.6.6.6          -       100     0      65006 65005 65006 i Or-ID:
 6.6.6.6
C-LST: 2.2.2.222
 * >Ec   223.255.255.248/30  192.168.168.14   -       100     0      65002 i
 *  ec   223.255.255.248/30  192.168.168.10   -       100     0      65002 i
 * >Ec   223.255.255.252/30  192.168.168.14   -       100     0      65002 i
 *  ec   223.255.255.252/30  192.168.168.10   -       100     0      65002 i
```

Note: EVPN routes are received from router ***2.2.2.222*** C-List (cluster list - basically identifying this route as from a route-reflector) with originating router being ***6.6.6.6***.

- The **`show ip route vrf tenant-b`** command displays the BGP table for VRF in ***tenant-b*** containing imported EVPN routes.

```
north-edge# show ip route vrf tenant-b

VRF: tenant-b
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort:
 B I    0.0.0.0/0 [200/0] via 6.6.6.6/32, IS-IS SR tunnel index 6, label 953372
                          via 192.168.58.12, Ethernet1/1, label 408006
                          via 192.168.59.12, Ethernet2/1, label 408006

 B E    10.10.10.0/24 [200/0] via 192.168.168.22, Ethernet6/2.2
                              via 192.168.168.20, Ethernet6/3.2

 B E    10.10.21.0/24 [200/0] via 192.168.168.22, Ethernet6/2.2
                              via 192.168.168.20, Ethernet6/3.2
 B I    100.10.10.0/24 [200/0] via 6.6.6.6/32, IS-IS SR tunnel index 6, label 953372
                                  via 192.168.58.12, Ethernet1/1, label 408006
                                  via 192.168.59.12, Ethernet2/1, label 408006

 C      192.168.168.20/31 is directly connected, Ethernet6/3.2
 C      192.168.168.22/31 is directly connected, Ethernet6/2.2
 B I    223.255.254.248/30 [200/0] via 6.6.6.6/32, IS-IS SR tunnel index 6, label 953372
                                     via 192.168.58.12, Ethernet1/1, label 408006
                                     via 192.168.59.12, Ethernet2/1, label 408006
 B I    223.255.254.252/30 [200/0] via 6.6.6.6/32, IS-IS SR tunnel index 6, label 953372
                                     via 192.168.58.12, Ethernet1/1, label 408006
                                     via 192.168.59.12, Ethernet2/1, label 408006
 B E    223.255.255.248/30 [200/0] via 192.168.168.22, Ethernet6/2.2
                                    via 192.168.168.20, Ethernet6/3.2
 B E    223.255.255.252/30 [200/0] via 192.168.168.22, Ethernet6/2.2
                                    via 192.168.168.20, Ethernet6/3.2
```

Note: If we look at the routes in the VRF for tenant-b, we see that the VPN label has now changed, whilst the transport label for ***NH 6.6.6.6*** is the same. The only difference seen in ***tenant-b***, aside from the different VPN label, is that there are no host-routes in ***tenant-b*** because within each DC ***tenant-b*** is running in asymmetric mode, therefore no host routes are generated/installed in the IP VRF.

18.15.3.2 Layer 3 EVPN Over LDP

The following figures illustrate an overview of the combines control and data planes.

Figure 18-55: Control Plane Tenant-A Over LDP

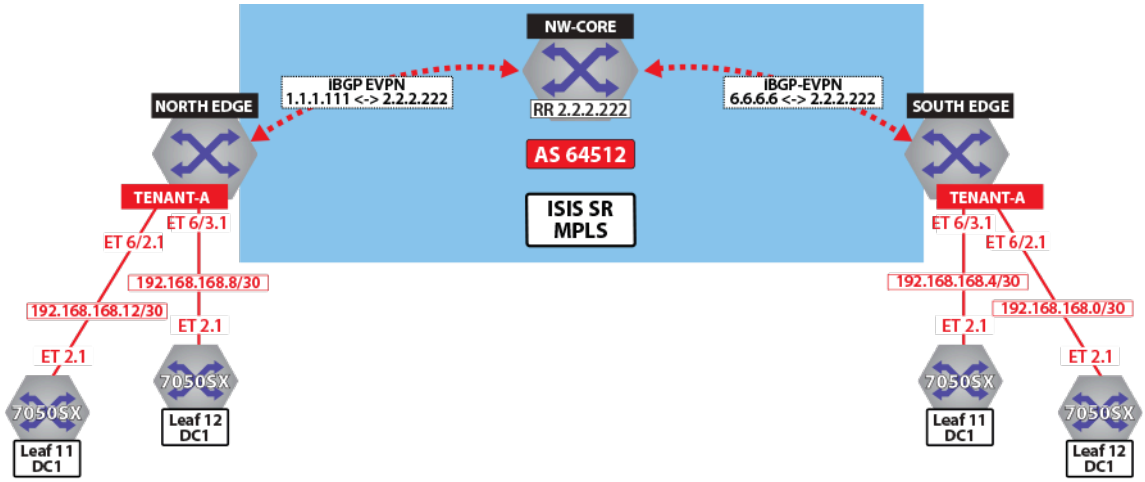


Figure 18-56: Control Plane Tenant-B over LDP

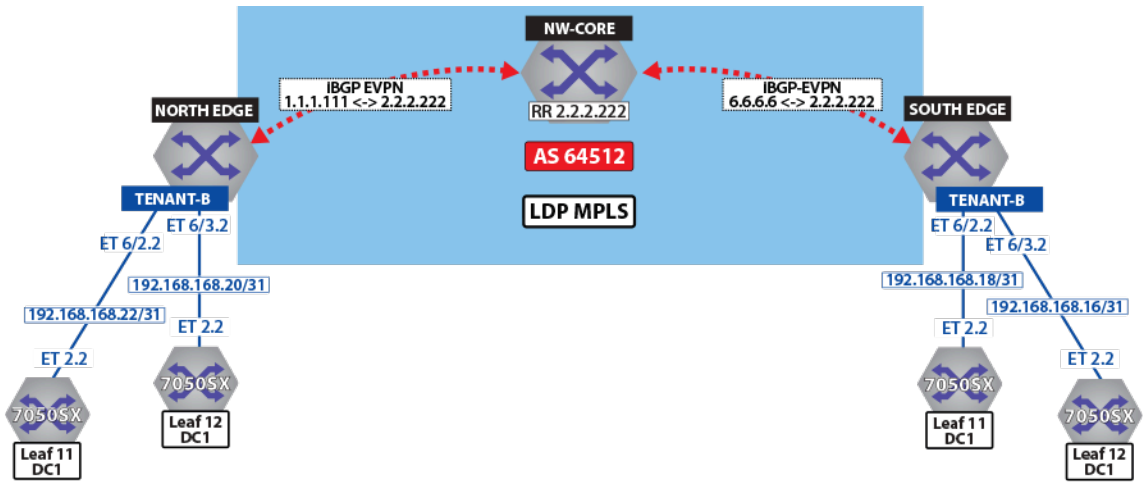
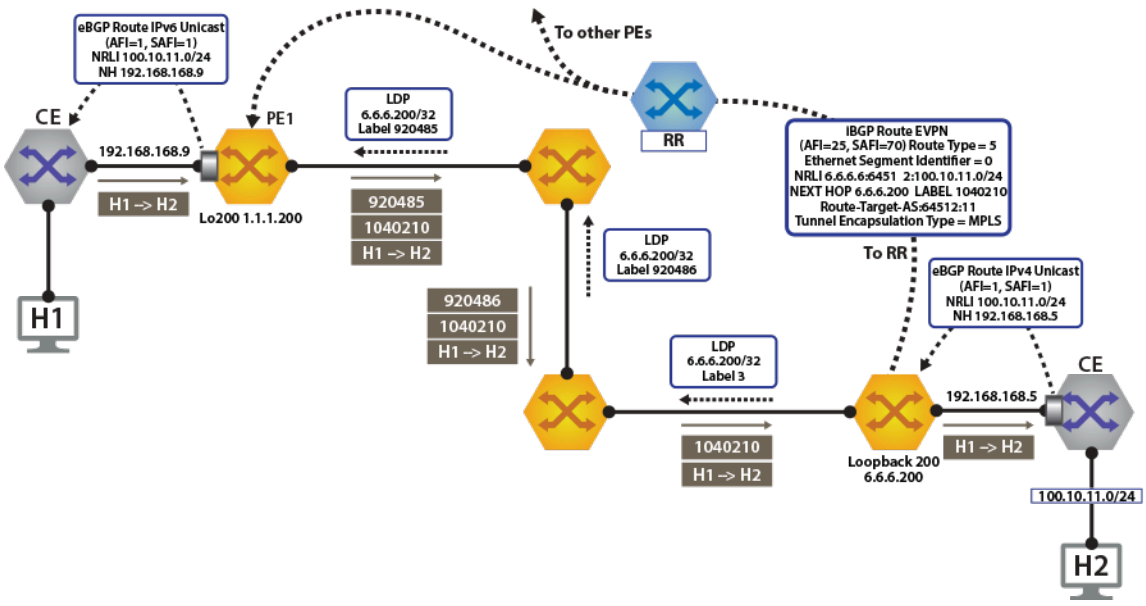


Figure 18-57: Control Plane & Forwarding Tenant-A Over LDP



To switch to using the MPLS LDP transport, we simply need to change the next-hop advertised for EVPN routes. As illustrated above, the next hop needs to be set to loopback ***200*** to use the LDP LSP.

This is simply achieved by configuring the next-hop for EVPN routes on both North Edge and South Edge routes. The output again includes the RD and IP prefix identifying the route. As seen in the output, we now have the NH set to ***6.6.6.200*** for ***tenant-a*** and ***tenant-b***.

```
router bgp 64512
   !
   address-family evpn
     neighbor default encapsulation mpls next-hop-self source-interface
 Loopback200
```

Once this is configured, we can check the BGP updates and the routes in the VRF.

```
north-edge# show bgp evpn route-type ip-prefix 100.10.11.0/24 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for ip-prefix 100.10.11.0/24, Route Distinguisher: 6.6.6.6:64512
 Paths: 1 available
  65006
    6.6.6.200 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:11 TunnelEncap:tunnelTypeMpls
      MPLS label: 958810
BGP routing table entry for ip-prefix 100.10.11.0/24, Route Distinguisher: 6.6.6.6:64513
 Paths: 1 available
  65006
    6.6.6.200 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64513:11 TunnelEncap:tunnelTypeMpls
      MPLS label: 953372
```



Note: Again, we have the same route in tenant-a and tenant-b in DC2. Therefore, the two other routes with ***RD 6.6.6.6:64513*** and ***RT 64513:11***. The VPN label has not changed, reinforcing the fact that the BGP VPN label is orthogonal to the transport label.

Finally, let us look at the routes in the VRF ***tenant-a***.

```
north-edge# show ip route vrf tenant-a

VRF: tenant-a
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS ----level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort:
 B I    0.0.0.0/0 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 958810
                          via 192.168.58.12, Ethernet1/1, label 904097
                          via 192.168.59.12, Ethernet2/1, label 904098

 B E    10.10.10.103/32 [200/0] via 192.168.168.14, Ethernet6/2.1
                                  via 192.168.168.10, Ethernet6/3.1
 B E    10.10.10.104/32 [200/0] via 192.168.168.14, Ethernet6/2.1

                                 via 192.168.168.10, Ethernet6/3.1
 B I    100.10.10.103/32 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 958810
                                    via 192.168.58.12, Ethernet1/1, label 904097
                                    via 192.168.59.12, Ethernet2/1, label 904098

 B I    192.168.168.4/30 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 958810
                                    via 192.168.58.12, Ethernet1/1, label 904097
                                    via 192.168.59.12, Ethernet2/1, label 904098
 C      192.168.168.8/30 is directly connected, Ethernet6/3.1
 C      192.168.168.12/30 is directly connected, Ethernet6/2.1
 B I    223.255.254.248/30 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 958810
```

```
                                       via 192.168.58.12, Ethernet1/1, label 904097
                                       via 192.168.59.12, Ethernet2/1, label 904098
 B I    223.255.254.252/30 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 958810
                                       via 192.168.58.12, Ethernet1/1, label 904097
                                       via 192.168.59.12, Ethernet2/1, label 904098
 B E    223.255.255.248/30 [200/0] via 192.168.168.14, Ethernet6/2.1
                                   via 192.168.168.10, Ethernet6/3.1
 B E    223.255.255.252/30 [200/0] via 192.168.168.14, Ethernet6/2.1
                                   via 192.168.168.10, Ethernet6/3.1
```

Note: As can be seen from the highlighted route above the label stack, the route has the same VPN route ***958810***, but the transport labels are now ***904097*** and ***904098*** on top (this is the ECMP label path to reach ***NH 6.6.6.200***).

As a comparison, let us look at the routes for ***tenant-b***.

```
north-edge# show ip route vrf tenant-b

VRF: tenant-b
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort:
 B I    0.0.0.0/0 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 953372
                          via 192.168.58.12, Ethernet1/1, label 904097
                          via 192.168.59.12, Ethernet2/1, label 904098

 B E    10.10.10.0/24 [200/0] via 192.168.168.22, Ethernet6/2.2
                              via 192.168.168.20, Ethernet6/3.2

                              via 192.168.168.20, Ethernet6/3.2
 B I    100.10.10.0/24 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 953372
                                  via 192.168.58.12, Ethernet1/1, label 904097
                                  via 192.168.59.12, Ethernet2/1, label 904098

                                     via 192.168.59.12, Ethernet2/1, label 904098
 B I    192.168.168.18/31 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 953372
                                     via 192.168.58.12, Ethernet1/1, label 904097
                                     via 192.168.59.12, Ethernet2/1, label 904098
 C      192.168.168.20/31 is directly connected, Ethernet6/3.2
 C      192.168.168.22/31 is directly connected, Ethernet6/2.2
 B I    223.255.254.248/30 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 953372
                                       via 192.168.58.12, Ethernet1/1, label 904097
                                       via 192.168.59.12, Ethernet2/1, label 904098
 B I    223.255.254.252/30 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 953372
                                       via 192.168.58.12, Ethernet1/1, label 904097
                                       via 192.168.59.12, Ethernet2/1, label 904098
 B E    223.255.255.248/30 [200/0] via 192.168.168.22, Ethernet6/2.2
                                   via 192.168.168.20, Ethernet6/3.2
 B E    223.255.255.252/30 [200/0] via 192.168.168.22, Ethernet6/2.2
```

Note: The only difference apart from the missing host routes (no host-route inject for this tenant), is the VPN label.

18.15.3.3 Layer 3 EVPN Over BGP-SR

The following figures illustrate an overview of the combined control and data planes.

Figure 18-58: Control Plane Tenant-A Over BGP-SR

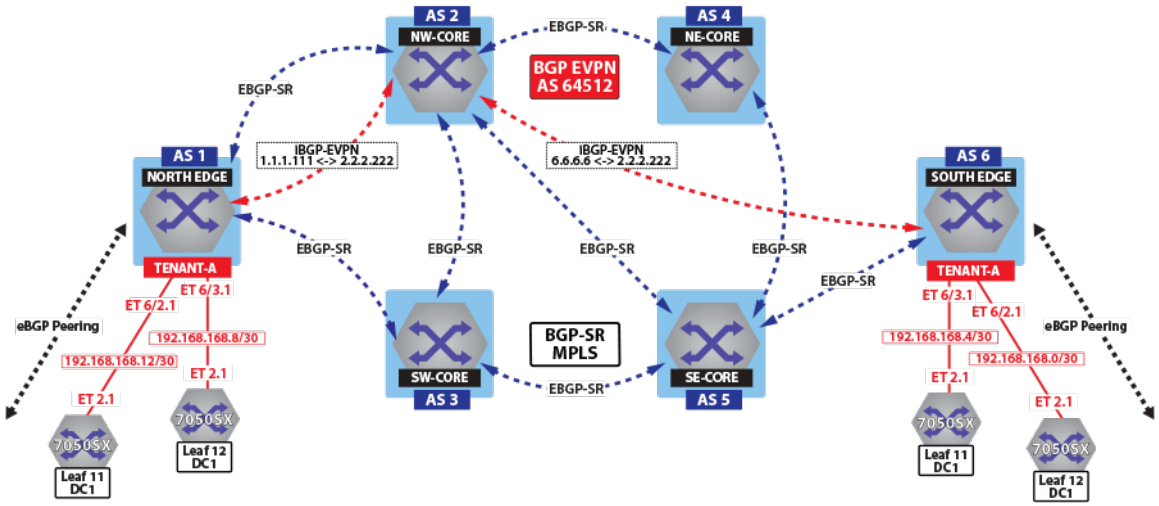


Figure 18-59: Control Plane Tenant-B Over BGP-SR

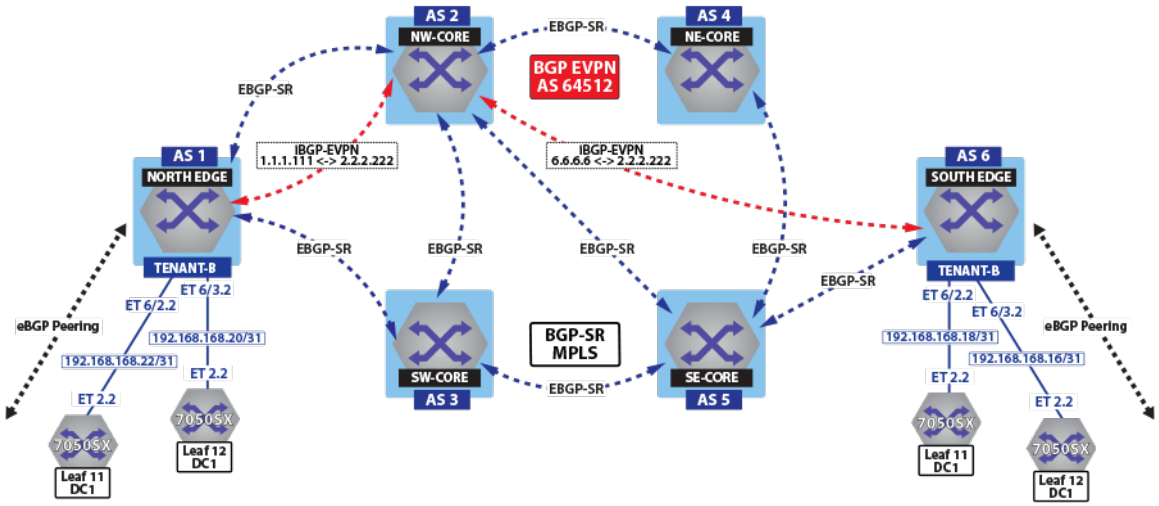
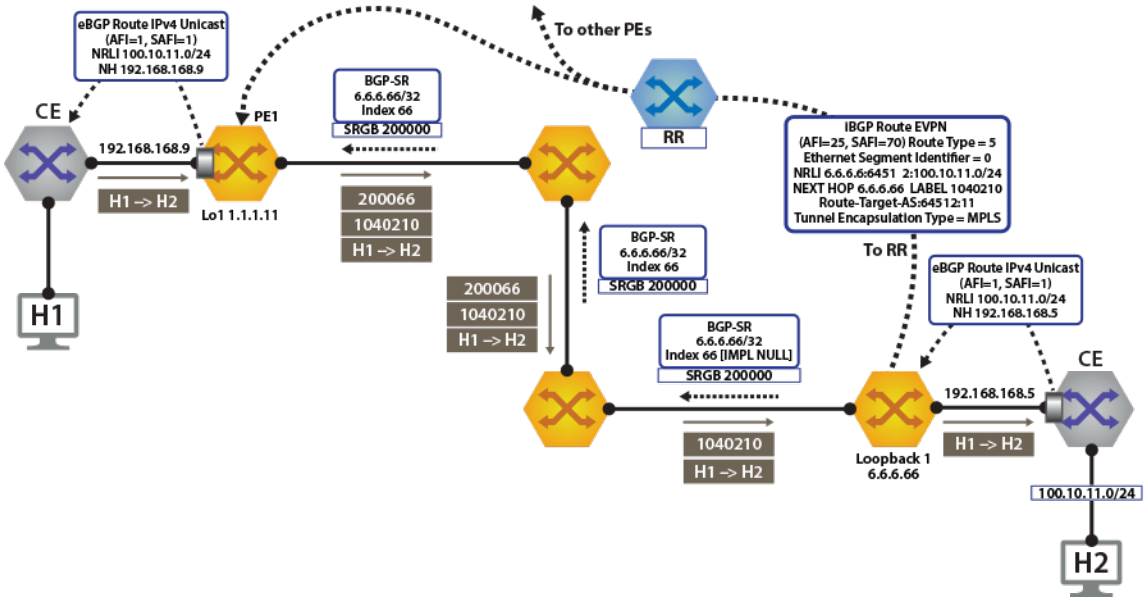


Figure 18-60: Control Plane & Forwarding Tenant-A Over BGP-SR



To switch to using the MPLS BGP-SR transport, we simply need to change the next-hop advertised for the EVPN routes. As shown in Control Plane ***tenant-b*** Over BGP-SR, the next hop needs to be set to ***loopback 1*** for using the BGP-SR LSP. This is achieved by configuring the next-hop for the EVPN routes.

```
router bgp 64512
   !
   address-family evpn
     neighbor default encapsulation mpls next-hop-self source-interface
 Loopback1
```

Once the next-hop for the EVPN routes are configured, we can check the BGP updates and the routes in the VRF. The output again includes the RD and IP prefix identifying the route. As seen in the output, we now have the NH set to ***6.6.6.66*** for ***tenant-a*** and ***tenant-b***.

```
North Edge.17:52:30# show bgp evpn route-type ip-prefix 100.10.11.0/24 detail

north-edge(config-if-Et2/1)#show bgp evpn route-type ip-prefix 100.10.11.0/24
 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for ip-prefix 100.10.11.0/24, Route Distinguisher:
 6.6.6.6:64512
 Paths: 1 available
  65006
    6.6.6.66 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:11 TunnelEncap:tunnelTypeMpls
      MPLS label: 958810
BGP routing table entry for ip-prefix 100.10.11.0/24, Route Distinguisher:
 6.6.6.6:64513
 Paths: 1 available
  65006
    6.6.6.66 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64513:11 TunnelEncap:tunnelTypeMpls
      MPLS label: 953372
```

Note: Again, we have the same route in tenant-a and tenant-b in DC2. Therefore, the two other routes with ***RD 6.6.6.6:64513*** and ***RT 64513:11***. The VPN label has not changed, reinforcing the fact that the BGP VPN label is orthogonal to the transport label.

Finally, let us look at the routes in the VRF ***tenant-a***.

```
North Edge.17:55:01# show ip route vrf tenant-a

VRF: tenant-a
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort:
 B I    0.0.0.0/0 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8, label 958810
                            via 192.168.58.12, Ethernet1/1, label 200066
                            via 192.168.59.12, Ethernet2/1, label 200066

 B E    10.10.10.103/32 [200/0] via 192.168.168.14, Ethernet6/2.1
                                via 192.168.168.10, Ethernet6/3.1
 B E    10.10.10.104/32 [200/0] via 192.168.168.14, Ethernet6/2.1
                                via 192.168.168.10, Ethernet6/3.1

                               via 192.168.168.10, Ethernet6/3.1
 B I    100.10.10.103/32 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8, label
 958810
                                    via 192.168.58.12, Ethernet1/1, label
 200066
                                    via 192.168.59.12, Ethernet2/1, label
 200066

 B I    192.168.168.4/30 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8, label
 958810
                                    via 192.168.58.12, Ethernet1/1, label
 200066
                                    via 192.168.59.12, Ethernet2/1, label
 200066
 C      192.168.168.8/30 is directly connected, Ethernet6/3.1
 C      192.168.168.12/30 is directly connected, Ethernet6/2.1
 B I    223.255.254.248/30 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8,
 label 958810
                                      via 192.168.58.12, Ethernet1/1, label
 200066
                                      via 192.168.59.12, Ethernet2/1, label
 200066
 B I    223.255.254.252/30 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8,
 label 958810
                                      via 192.168.58.12, Ethernet1/1, label
 200066
                                      via 192.168.59.12, Ethernet2/1, label
 200066
 B E    223.255.255.248/30 [200/0] via 192.168.168.14, Ethernet6/2.1
                                   via 192.168.168.10, Ethernet6/3.1
 B E    223.255.255.252/30 [200/0] via 192.168.168.14, Ethernet6/2.1
                                   via 192.168.168.10, Ethernet6/3.1
```

As can be seen from the highlighted route above the label stack, the route are the transport labels ***958810*** and ***200066*** on top (this is the ECMP label path to reach ***NH 6.6.6.66***), with the ***tenant-a*** VPN label ***958810*** next in the stack, identifying the route as belonging to ***tenant-a***.

As a comparison, let us look at the routes for ***tenant-b***. As seen in the output, the VPN label assigned to ***tenant-b*** is ***953372***.

```
north-edge# show bgp evpn route-type ip-prefix 100.10.11.0/24 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for ip-prefix 100.10.11.0/24, Route Distinguisher:
 6.6.6.6:64512
 Paths: 1 available
  65006
    6.6.6.66 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:11 TunnelEncap:tunnelTypeMpls
      MPLS label: 958810
BGP routing table entry for ip-prefix 100.10.11.0/24, Route Distinguisher:
 6.6.6.6:64513
 Paths: 1 available
  65006
    6.6.6.66 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64513:11 TunnelEncap:tunnelTypeMpls
      MPLS label: 953372
north-edge#
```

If we now look at the routes in the VRF for ***tenant-b***, we see that the VPN label has now changed, while the transport label (for ***NH 6.6.6.66*** is the same). The only difference seen in ***tenant-b***, aside from the different VPN label, is that there are no host-routes in ***tenant-b*** because within each DC ***tenant-b*** is running in asymmetric mode; therefore, no host routes are generated/installed in the IP VRF.

```
north-edge# show ip route vrf tenant-b

VRF: tenant-b
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort:
 B I    0.0.0.0/0 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8, label 953372
                            via 192.168.58.12, Ethernet1/1, label 200066
                            via 192.168.59.12, Ethernet2/1, label 200066

 B E    10.10.10.0/24 [200/0] via 192.168.168.22, Ethernet6/2.2
                               via 192.168.168.20, Ethernet6/3.2

 B E    10.10.21.0/24 [200/0] via 192.168.168.22, Ethernet6/2.2
                               via 192.168.168.20, Ethernet6/3.2
 B I    100.10.10.0/24 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8, label
 953372
                                   via 192.168.58.12, Ethernet1/1, label 200066
                                   via 192.168.59.12, Ethernet2/1, label 200066
```

```
B I    192.168.168.18/31 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8, label
953372
                                    via 192.168.58.12, Ethernet1/1, label
200066
                                    via 192.168.59.12, Ethernet2/1, label
200066
C      192.168.168.20/31 is directly connected, Ethernet6/3.2
C      192.168.168.22/31 is directly connected, Ethernet6/2.2
B I    223.255.254.248/30 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8,
label 953372
                                     via 192.168.58.12, Ethernet1/1, label
200066
                                     via 192.168.59.12, Ethernet2/1, label
200066
B I    223.255.254.252/30 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8,
label 953372
                                     via 192.168.58.12, Ethernet1/1, label
200066
                                     via 192.168.59.12, Ethernet2/1, label
200066
B E    223.255.255.248/30 [200/0] via 192.168.168.22, Ethernet6/2.2
                                   via 192.168.168.20, Ethernet6/3.2
B E    223.255.255.252/30 [200/0] via 192.168.168.22, Ethernet6/2.2
                                   via 192.168.168.20, Ethernet6/3.2
```

18.15.4 EVPN VxLAN IPv6 Overlay

The EVPN VXLAN L3 Gateway using EVPN IRB supports routing traffic from one IPv6 host to another IPv6 host on a stretched VXLAN VLAN on platforms that support ND Proxy and ND suppression. The **`ipv6 address virtual`** command enables the use of one MAC address for all SVI instead of one per SVI. Both EVPN IRB and VXLAN tunnel interface are required for the feature to work. The VXLAN must be configured with a VNI or the VRF for the VLAN must be configured with a VRF/VNI mapping.

Configuring for Overlay

The following configures the switches for global IPv6 unicast routing and IPv6 unicast routing for each VRF.

```
switch(config)# ipv6 unicast-routing
switch(config)# ipv6 unicast-routing vrf tenant-c
```

The following configures the switches with a virtual MAC address, which is used for mapping all virtual router IP addresses. For VARP configs, the address is receive-only; the switch never sends packets with this address as the source. For **`ip address virtual`**, the address is also used as the source for ARP packets.

```
Switch(config)# ipv6 virtual-router mac-address <mac>
```

The following shows the switch with IPv6 configured where one SVI uses one physical IP address.

```
switch# show run int vlan 501
interface Vlan501
      vrf forwarding tenant-c
      ipv6 enable
ipv6 address 2004:220::1:2/112
ipv6 virtual-router address 2004:220::1:10
```

The following shows configuration for the switch such that all SVI use the virtual MAC address and only one physical IP address.

```
switch# show run int vlan 501
interface Vlan501
```

```
      vrf forwarding tenant-c
      ipv6 enable
ipv6 address virtual 2004:220::1:10/112
```

Limitations

Any topology that requires a VXLAN Virtual VTEP address configuration is not supported.

Example Configurations

VRF-TO-VNI MAP and VLAN-TO-VNI MAP

Under ***Vxlan1*** interface:

```
switch(config)#
interface vxlan1
   vxlan vrf tenant-c vni 4001
   vxlan vlan 501 vni 10501
```

MAC-VRF

Under BGP router configuration mode:

```
switch(config)#
Router bgp 65000
vlan 501
      rd 20.1.1.1:10501
      route-target both 1:10501
      redistribute learned
```

IPv6 VRF BGP

```
switch(config)#
router bgp 65000
vrf tenant-c
   rd 2.0.0.1:4001
   router-target import evpn 4001:4001
   router-target export evpn 4001:4001

! configure IPv4 router ID under the BGP VRF configuration
! for activating V6-only VRF
!
router-id 4.0.0.1
```

The selective installation configuration is the same for ARP and IPv6 ND.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)#arp ?
 proxy              Proxy ARP
 selective-install  Install ARP entries for remote hosts on demand
switch(config-rtr-l2-vpn)#arp selective-install
```

The following disables the ND proxy reply to an NS for the specified target IPv6 address(es).

```
switch(config)#
ipv6 prefix-list list-test
seq 10 deny 2000:0:0:69::19/64
! do not perform ND proxy on 2000:0:0:69::19/64

switch(config)# router l2-vpnswitch(config-rtr-l2-vpn)#nd proxy prefix-list
 list-test
```

The following restores the proxy behavior.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# no nd proxy prefix-list list-test
```

The following disables router solicitation packets sent by a host from getting flooded to all VTEPs.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# nd rs flooding disabled
```

The following restores the default behavior.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# no nd rs flooding disabled
```

The following disables Duplicate-Address-Detection (DAD) multicast packets from getting flooded to all VTEPs when there is no matching IP to MAC binding found in EVPN published IP to MAC bindings. When there is a match found, a DAD frame is flooded to all VTEPs (instead of doing a proxy reply) to confirm that host liveliness.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# nd dad flooding disabled
```

The following restores the default behavior.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# no nd dad flooding disabled
```

The following disables Neighbor Advertisement (NA) multicast packets from the SVI configured as a virtual router from getting flooded to all VTEPs.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# virtual-router neighbor advertisement flooding
 disabled
```

The following restores the default behavior.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# no virtual-router neighbor advertisement flooding
 disabled
```

The following disables Gratuitous ARP multicast packets from the SVI configured as a virtual router from getting flooded to all VTEPs.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# virtual-router arp advertisement flooding disabled
```

The following restores the default behavior.

```
switch(config)# router l2-vpn
switch(config-rtr-l2-vpn)# no virtual-router arp advertisement flooding
 disabled
```

Checking the Status of the Switches

IPv6 Local Host

The following displays the ND bindings for a given VRF. The output shows that the local host ***002c.0100.0001*** has an IPv6 link local address ***fe80::22c:1ff:fe00:1*** and a global IPv6 address ***2004:220::1:50***. The host is connected to the MLAG port-channel ***20***.

```
switch# show ipv6 neighbors vrf tenant-c vlan 501 | i 002c.0100.0001
2004:220::1:50          N/A 002c.0100.0001   REACH Vl501, Port-Channel20
fe80::22c:1ff:fe00:1    N/A 002c.0100.0001   REACH Vl501, Port-Channel20
```

EVPN IRB redistributes all the local hosts in ***VLAN 501***. The MAC address of the host is advertised as EVPN Type 2 MAC-only route advertisement. The global IPv6 to MAC binding is advertised using MAC-IP route.

Note: By default, the IPv6 link local binding is not advertised by EVPN.

The following displays the two MAC-only routes and two MAC-IP routes. In both cases, one route is locally originated and the second one advertised by the MLAG peer with the same VTEP IP *10.0.0.1*.

```
switch# show bgp evpn route-type mac-ip 002c.0100.0001
BGP routing table information for VRF default
Router identifier 1.0.1.1, local AS number 65000
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
 - ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
 Link Local Nexthop

         Network                Next Hop            Metric  LocPref Weight
 Path
 * >     RD: 20.1.1.1:10501 mac-ip 002c.0100.0001
                                -                     -       -       0
 i
         RD: 20.1.1.2:10501 mac-ip 002c.0100.0001
                                10.0.0.1               -       100     0
 65002 65003 i
 * >     RD: 20.1.1.1:10501 mac-ip 002c.0100.0001 2004:220::1:50
                                -                     -       -       0
 i
         RD: 20.1.1.2:10501 mac-ip 002c.0100.0001 2004:220::1:50
                                10.0.0.1               -       100     0
 65002 65003 i
```

IPv6 Link Local Redistribution

The following configures **`link-local redistribution`** command under BGP router MAC-VRF configuration mode to redistribute IPv6 link local binding.

```
vlan 501
rd 20.1.1.1:10501
route-target both 1:10501
redistribute learned
redistribute link-local ipv6
```

When this is configured, NS from a local host for a link local target will get proxy-replied by the ingress VTEP if the binding is published to EVPN by a remote VTEP. The NS in that case will not get replicated to other VTEPs.

IPv6 Remote Host

The following displays the MAC-only and MAC-IP routes for remote host ***002d.0100.0001***. These two routes originated from VTEP ***10.0.0.2***.

```
switch# show bgp evpn route-type mac-ip 002d.0100.0001 detail
BGP routing table information for VRF default
Router identifier 1.0.1.1, local AS number 65000

BGP routing table entry for mac-ip 002d.0100.0001, Route Distinguisher:
 20.1.1.3:10501
 Paths: 1 available
  65002 65004
    10.0.0.2 from 1.0.1.111 (1.0.1.111)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, best
      Extended Community: Route-Target-AS:1:10501 TunnelEncap:tunnelTypeVxlan
      VNI: 10501 ESI: 0000:0000:0000:0000:0000

BGP routing table entry for mac-ip 002d.0100.0001 2004:220::1:151, Route
 Distinguisher: 20.1.1.3:10501
 Paths: 1 available
  65002 65004
    10.0.0.2 from 1.0.1.111 (1.0.1.111)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, best
      Extended Community: Route-Target-AS:1:10501 Route-Target-AS:4001:4001
 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:28:99:3a:be:53:42
      VNI: 10501 L3 VNI: 4003 ESI: 0000:0000:0000:0000:0000
```

IPv6 Remote Binding for Asymmetric IRB

The following displays the local MAC-VRF ***vlan 501*** is configured to import RT two octets ASN ***RT 1:10501***. The MAC-IP route is imported into remote binding for ***vlan 501***.

```
switch# show ipv6 neighbors remote vlan 501
ARP remote bindings
VLAN IP Address      MAC Address
---- --------------- --------------
501  2004:220::1:151 002d.0100.0001
```

Without ARP Selective install, always install the remote IPv6 ND binding.

The following displays the ND bindings installed in the IPv6 cache. The interface for remote hosts is always ***vxlan1 501*** andage is displayed with a '-'.

```
switch# show ipv6 neighbors vrf tenant-c vlan 501 2004:220::1:151
IPv6 Address          Age Hardware Addr    State Interface
2004:220::1:151         - 002d.0100.0001   REACH Vl501, Vxlan1
```

IPv6 Remote Host for Symmetric IRB

The following displays the BGP information for a specific IPv6 prefix in a VRF.

```
switch# show ipv6 bgp  2004:220::1:151 vrf tenant-c
BGP routing table information for VRF tenant-c
Router identifier 100.52.7.254, local AS number 65000
BGP routing table entry for 2004:220::1:151/128
  Paths: 2 available
  65002 65004
  10.0.0.2 from 1.0.1.111 (1.0.1.111), imported EVPN route, RD 20.1.1.3:10501
  Origin IGP, metric -, localpref 100, weight 0, valid, external, best
Extended Community: Route-Target-AS:1:10501 Route-Target-AS:4001:4001
 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:28:99:3a:be:53:42
```

```
    Remote VNI: 4003
    65000 65002 65004
 2005:951:1:1::1:2 from 2005:951:1:1::1:2 (100.52.7.254)
 Origin IGP, metric -, localpref 100, weight 0, valid, external
 Not best: As path length
```

The following displays the route for a specific IPv6 prefix in a VRF.

```
switch# show ipv6 route vrf tenant-c 2004:220::1:151
VRF: tenant-c
Routing entry for 2004:220::1:151
Codes: C - connected, S - static, K - kernel, O3 - OSPFv3, B - BGP, R - RIP, A
 B - BGP Aggregate,
I L1 - IS-IS level 1, I L2 - IS-IS level 2, DH - DHCP, NG - Nexthop Group
 Static Route, M - Martian,
DP - Dynamic Policy Route, L - VRF Leaked

B      2004:220::1:151/128 [200/0]
         via VTEP 10.0.0.2 VNI 4003 router-mac 28:99:3a:be:53:42
```

The following displays the VXLAN SW counters for IPv6 Neighbor Discovery Packets.

```
switch# show vxlan counters software | egrep 'ND|neighbor'
ND NS pkts skipped HER as target Ip matched SVI IP   :  0
ND NS proxy errors during transmit                   :  0
ND NS proxy neighbor remote binding misses           :  0
ND NS proxy neighbor cache misses                    :  0
ND NS proxy denied due to ACL                        :  0
ND NS proxy not applied as neighbor entry is dynamic :  0
ND NS proxy not applied as target link is local      :  0
ND NS proxy not applied as target IP is local        :  0
ND NS proxy not applied as sender link not in fdb    :  0
ND NS proxy not applied as pkt is invalid            :  0
ND NS proxy DAD frames suppressed                    :  0
ND NS proxy neighbor advt sent                       :  0
ND NS pkts from unspecified source                   :  9
ND NS pkts total suppressed                          :  0
ND NS pkts total received                            :  9
ND NA pkts total suppressed                          :  0
ND NA pkts total received                            :  0
ND NA pkts invalid                                   :  0
ND NA pkts not suppressed as source is SVI           :  0
ND NA pkts suppressed as source is SVI               :  0
ND RS pkts total suppressed                          :  0
total dynamic neighbor cache entries added in error  :  0
```

The following displays the VXLAN VARP packets for IPv6 **ipv6 address virtual** configurations.

```
Switch# show vxlan counters varp | grep 'neighbor'
neighbor advertisements received                     :  0
neighbor advertisements received in error            :  0
neighbor advertisements not headend replicated       :  0
neighbor sync msgs sent to mlag-peer                 :  0
neighbor cache installed                             :  0
neighbor cache install err                           :  0
neighbor cache install conflicts                     :  0
neighbor sync msgs received from mlag-peer           :  0
neighbor cache synced install err                    :  0
neighbor cache synced install conflicts              :  0
```

18.15.5 IP VPNs Sample Configuration

Here, we examine BGP EVPN layer 3 VPN over an LDP, ISIS-SR, and BGP-SR transport LSPs. This highlights the separation between the transport and the VPN overlay service.

The following figures illustrate the sample VPN Physical Topology.

Figure 18-61: IPv4 VPN Physical Topology

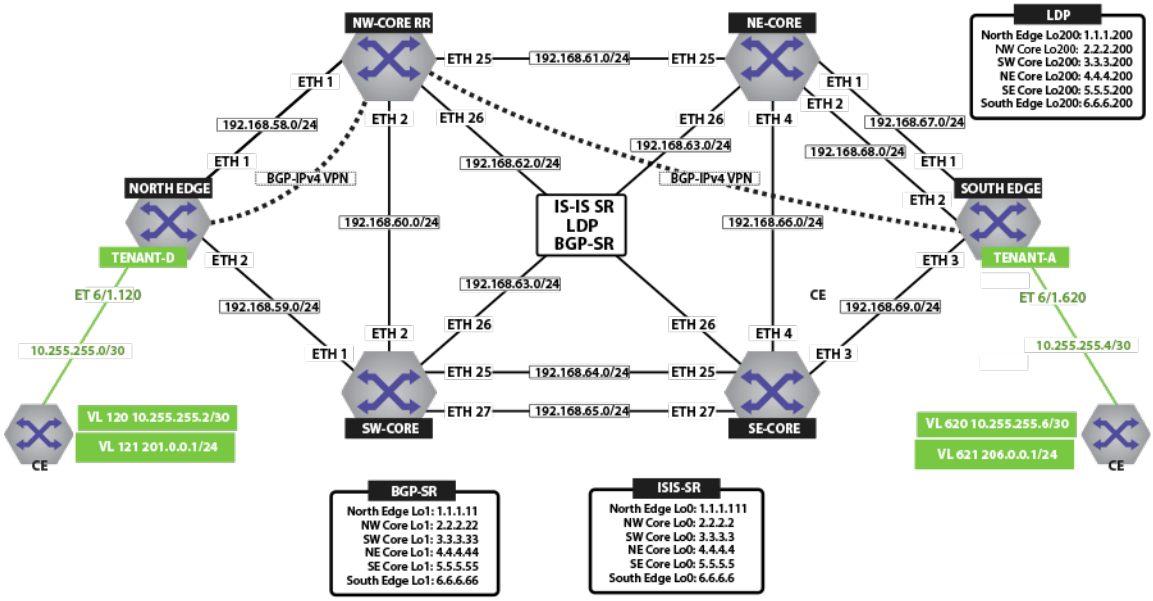
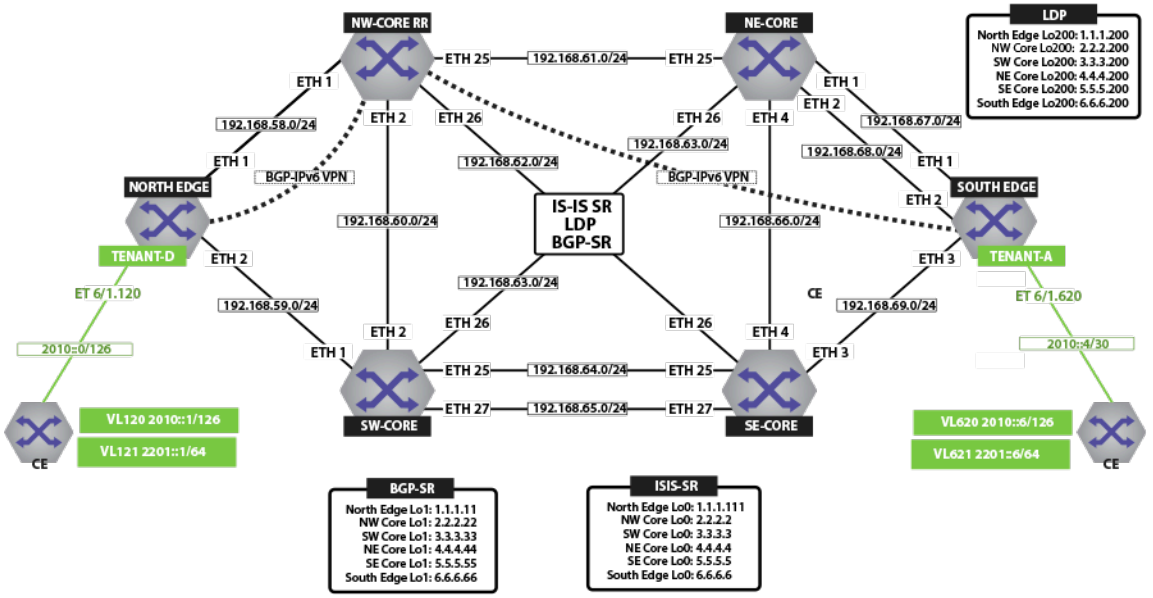


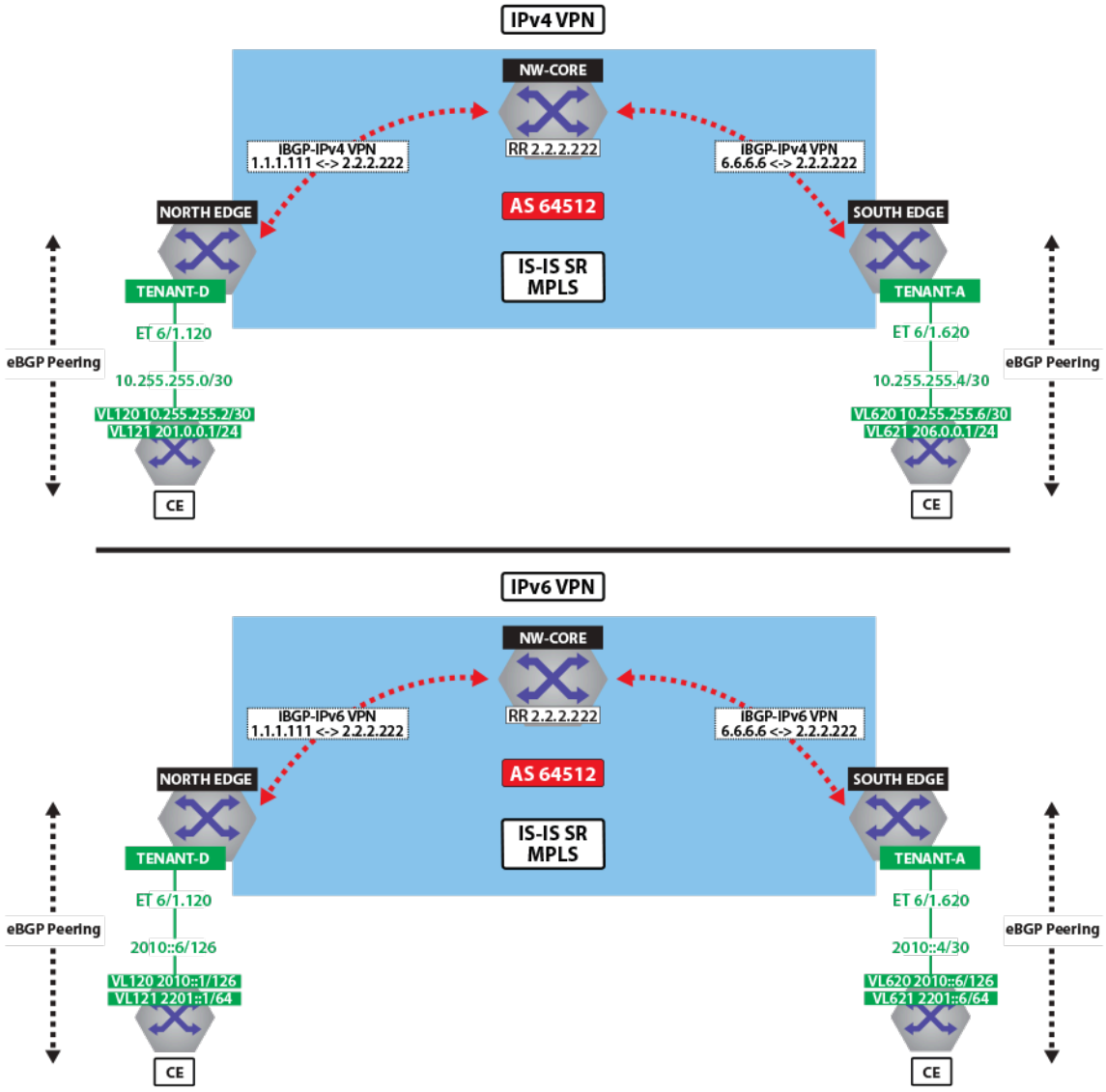
Figure 18-62: IPv6 VPN Physical Topology



18.15.5.1 IP VPN over ISIS-SR

The figure below illustrates an overview of the combined control and data planes.

Figure 18-63: IPv4 VPN and IPv6 VPN Over ISIS-SR MPLS



The next two figures illustrate the forwarding path and control plane for both IP traffic over ISIS MPLS segment routing.

Figure 18-64: IPv4 VPN Forwarding Over ISIS-SR MPLS

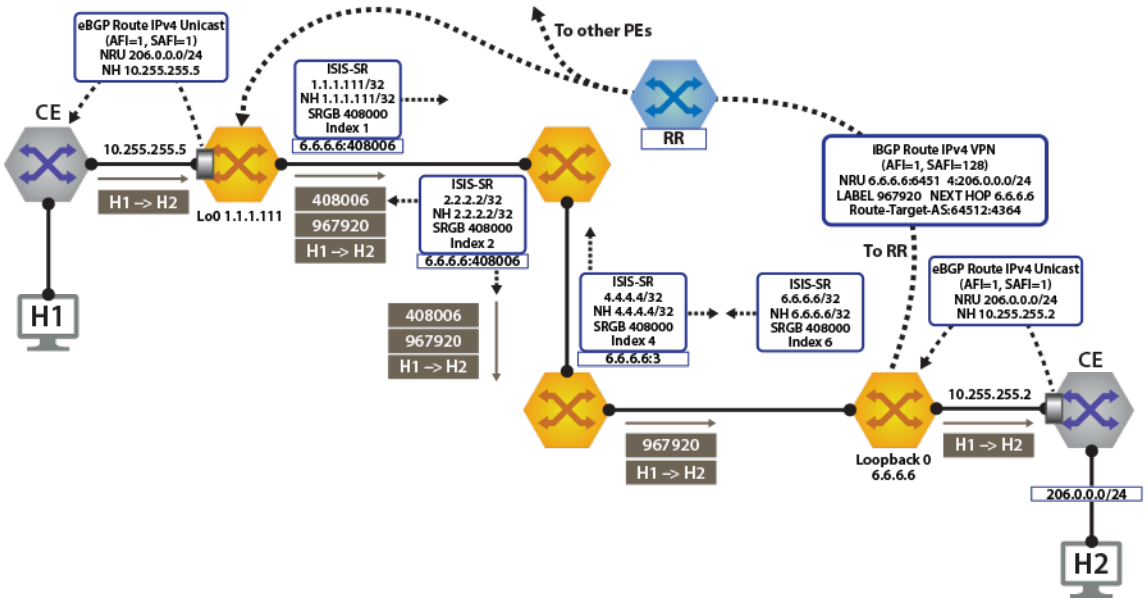
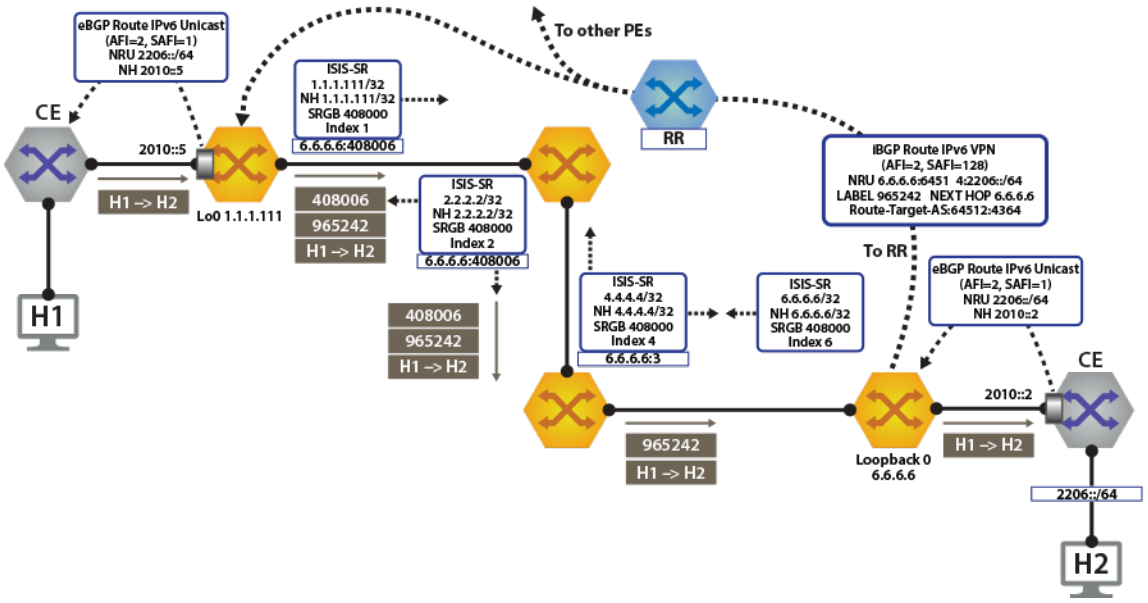


Figure 18-65: IPv6 VPN Forwarding Over ISIS-SR MPLS



View IPv4 and IPv6 Routes in the VRF

Both North Edge and South Edge routers have an eBGP peering session out to the CE; and learning routes from CE and remote PE.

- The **`show ip route vrf tenant-d`** command displays IPv4 Routes in the VRF of North Edge.

```
north-edge# show ip route vrf tenant-d

VRF: tenant-d
```

```
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort is not set

 B I    10.255.255.0/30 [200/0] via 6.6.6.6/32, IS-IS SR tunnel index 6,
 label 967920
                                    via 192.168.58.12, Ethernet1/1, label
 408006
 C      10.255.255.4/30 is directly connected, Ethernet6/1.120
 B E    201.0.0.0/24 [200/0] via 10.255.255.6, Ethernet6/1.120
 B I    206.0.0.0/24 [200/0] via 6.6.6.6/32, IS-IS SR tunnel index 6, label
 967920
                                  via 192.168.58.12, Ethernet1/1, label 408006
```

- The **`show ip route vrf tenant-d`** command displays IPv4 Routes in the VRF of South Edge.

```
south-edge# show ip route vrf tenant-d

VRF: tenant-d
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort is not set

 C      10.255.255.0/30 is directly connected, Ethernet6/1.620
 B I    10.255.255.4/30 [200/0] via 1.1.1.111/32, IS-IS SR tunnel index 5,
 label 951536
                                    via 192.168.68.11, Ethernet2/1, label
 408001
 B I    201.0.0.0/24 [200/0] via 1.1.1.111/32, IS-IS SR tunnel index 5,
 label 951536
                                  via 192.168.68.11, Ethernet2/1, label 408001
 B E    206.0.0.0/24 [200/0] via 10.255.255.2, Ethernet6/1.620
```

- The **`show ipv6 route vrf tenant-d`** command displays IPv6 Routes in the VRF of North Edge.

```
north-edge# show ipv6 route vrf tenant-d
VRF: tenant-d
Displaying 4 of 7 IPv6 routing table entries
Codes: C - connected, S - static, K - kernel, O3 - OSPFv3, B - BGP, R - RIP,
 A B - BGP Aggregate, I L1 -
IS-IS level 1, I L2 - IS-IS level 2, DH - DHCP, NG - Nexthop Group Static
 Route, M - Martian, DP - Dynamic
Policy Route

 B    2010::/126 [200/0]
       via 6.6.6.6/32, IS-IS SR tunnel index 6, label 965242
          via 192.168.58.12, Ethernet1/1, label 408006
```

```
 C    2010::4/126 [0/0]
       via Ethernet6/1.120, directly connected
 B    2201::/64 [200/0]
       via 2010::6, Ethernet6/1.120
 B    2206::/64 [200/0]
       via 6.6.6.6/32, IS-IS SR tunnel index 6, label 965242
          via 192.168.58.12, Ethernet1/1, label 408006
```

- The **show ipv6 route vrf tenant-d** command displays IPv6 Routes in the VRF of South Edge.

```
south-edge# show ipv6 route vrf tenant-d

VRF: tenant-d
Displaying 4 of 7 IPv6 routing table entries
Codes: C - connected, S - static, K - kernel, O3 - OSPFv3, B - BGP, R - RIP,
 A B - BGP Aggregate, I L1 -
IS-IS level 1, I L2 - IS-IS level 2, DH - DHCP, NG - Nexthop Group Static
 Route, M - Martian, DP - Dynamic
Policy Route

 C    2010::/126 [0/0]
       via Ethernet6/1.620, directly connected
 B    2010::4/126 [200/0]
       via 1.1.1.111/32, IS-IS SR tunnel index 5, label 948858
          via 192.168.68.11, Ethernet2/1, label 408001
 B    2201::/64 [200/0]
       via 1.1.1.111/32, IS-IS SR tunnel index 5, label 948858
          via 192.168.68.11, Ethernet2/1, label 408001
 B    2206::/64 [200/0]
       via 2010::2, Ethernet6/1.620
```

Activating IP VPN

In all scenarios, the IP VPN must be activated under BGP and neighbors configured to exchange the IP VPN NLRIs.The tenant's VRF (tenant-d) is associated with a dynamically assigned label by BGP.

North Edge

```
service routing protocols model multi-agent

router bgp 64512
   router-id 1.1.1.111
   maximum-paths 128 ecmp 128
   neighbor 2.2.2.222 remote-as 64512
   neighbor 2.2.2.222 update-source Loopback0
   neighbor 2.2.2.222 bfd
   neighbor 2.2.2.222 send-community extended
   neighbor 2.2.2.222 maximum-routes 12000
   !
   address-family vpn-ipv4
      neighbor 2.2.2.222 activate
      neighbor default encapsulation mpls next-hop-self source-interface
 Loopback0
   !
   address-family vpn-ipv6
      neighbor 2.2.2.222 activate
      neighbor default encapsulation mpls next-hop-self source-interface
 Loopback0
   !
```

South Edge

```
service routing protocols model multi-agent

router bgp 64512
   router-id 6.6.6.6
   maximum-paths 128 ecmp 128
   neighbor 2.2.2.222 remote-as 64512
   neighbor 2.2.2.222 update-source Loopback0
   neighbor 2.2.2.222 bfd
   neighbor 2.2.2.222 send-community extended
   neighbor 2.2.2.222 maximum-routes 12000
   !
   address-family vpn-ipv4
      neighbor 2.2.2.222 activate
      neighbor default encapsulation mpls next-hop-self source-interface
 Loopback0
   !
   address-family vpn-ipv6
      neighbor 2.2.2.222 activate
      neighbor default encapsulation mpls next-hop-self source-interface
 Loopback0
   !
```

The configuration above provides the following:

- It enables the multi-agent routing protocol model, which is required for BGP VPN support.
- It sets the local autonomous system number to ***64512*** and configured the route-reflector for both IPv4 VPN and IPv6 VPN capabilities.
- It sets the IP VPN encapsulation type to MPLS (default).
- It specifies that ***Loopback0*** will be used as the next-hop for all advertised VPN routes. The underlay configuration must provide MPLS LSPs from remote PEs to this loopback interface address.

Layer 3 Overlay Configuration

Distribution of Layer 3 routes over BGP is enabled by configuring one or more IP VRFs under the router bgp configuration mode. Additionally, either IPv4 or IPv6 routing must be enabled in the VRF.

- Configure IP VRF in the North Edge router.

```
vrf instance tenant-d
ip routing vrf tenant-d
ipv6 unicast-routing vrf tenant-d
!
router bgp 64512
    vrf tenant-d
      rd 1.1.1.1:64514
      route-target import vpn-ipv4 64512:4364
      route-target import vpn-ipv6 64512:4364
      route-target export vpn-ipv4 64512:4364
      route-target export vpn-ipv6 64512:4364
      neighbor 10.255.255.6 remote-as 65011
      neighbor 10.255.255.6 maximum-routes 12000
      neighbor 2010::6 remote-as 65011
      neighbor 2010::6 maximum-routes 12000
      !
      address-family ipv6
         neighbor 2010::6 activate
      redistribute connected
      !
```

- Configure IP VRF in the South Edge router.

```
vrf instance tenant-d
ip routing vrf tenant-d
ipv6 unicast-routing vrf tenant-d
!
router bgp 64512
   vrf tenant-d
      rd 6.6.6.6:64514
      route-target import vpn-ipv4 64512:4364
      route-target import vpn-ipv6 64512:4364
      route-target export vpn-ipv4 64512:4364
      route-target export vpn-ipv6 64512:4364
      neighbor 10.255.255.2 remote-as 65010
      neighbor 10.255.255.2 maximum-routes 12000
      neighbor 2010::2 remote-as 65010
      neighbor 2010::2 maximum-routes 12000
      !
      address-family ipv6
         neighbor 2010::2 activate
       redistribute connected
      !
```

These IP VRF configurations provide the following functionalities:

- It defines overlay VRFs (tenant-d) on the PE and enables IP unicast routing.
- The VRF is assigned a unique Route-Distinguisher (RD). The RD allows the PE to advertise VPN routes for the same IP prefix that have been exported by different VRFs. The NLRI RouteKey of a route exported from the VRFs IPv4 table into VPN consists of both the RD and the original IP prefix.
- The Route-Target (RT) extended communities for the VRF. The RTs are associated with all routes exported from the VRF. Received VPN routes carrying at least one RT matching the VRFs configuration are imported into the VRF.

Verifying IP VPNs over ISIS-SR

- The **`show bgp vpn-ipv4 summary`** command displays the status of the VPN IP peers in the North Edge router with the BGP VPN enabled.

```
north-edge# show bgp vpn-ipv4 summary
BGP summary information for VRF default
Router identifier 1.1.1.111, local AS number 64512
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down
 State
PfxRcd PfxAcc
  2.2.2.222        4  64512            172        45    0    0 00:17:16
 Estab  2      2
north-edge#  show bgp vpn-ipv6 summary
BGP summary information for VRF default
Router identifier 1.1.1.111, local AS number 64512
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down
 State
PfxRcd PfxAcc
  2.2.2.222        4  64512            172        45    0    0 00:17:20
 Estab  2      2
```

- The **`show bgp vpn-ipv4`** command displays routes sent and received through IP VPN.

```
north-edge# show bgp vpn-ipv4
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
```

```
Route status codes: s - suppressed, * - valid, > - active, # - not
 installed, E - ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop
 - Link Local Nexthop

         Network             Next Hop         Metric  LocPref Weight Path
 * >     RD: 6.6.6.6:64514 IPv4 prefix 10.255.255.0/30
                            6.6.6.6          -       100     0      65010 i
 Or-ID: 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 1.1.1.1:64514 IPv4 prefix 10.255.255.4/30
                            -                -       100     0      65011 i
 * >     RD: 1.1.1.1:64514 IPv4 prefix 201.0.0.0/24
                            -                -       100     0      65011 i
 * >     RD: 6.6.6.6:64514 IPv4 prefix 206.0.0.0/24
                            6.6.6.6          -       100     0      65010 i
 Or-ID: 6.6.6.6 C-LST: 2.2.2.222

north-edge# show bgp vpn-ipv6
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
Route status codes: s - suppressed, * - valid, > - active, # - not
 installed, E - ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop
 - Link Local Nexthop

         Network             Next Hop         Metric  LocPref Weight Path
 * >     RD: 6.6.6.6:64514 IPv6 prefix 2010::/126
                            6.6.6.6          -       100     0      65010 i
 Or-ID: 6.6.6.6 C-LST: 2.2.2.222
 * >     RD: 1.1.1.1:64514 IPv6 prefix 2010::4/126
                            -                -       100     0      65011 i
 * >     RD: 1.1.1.1:64514 IPv6 prefix 2201::/64
                            -                -       100     0      65011 i
 * >     RD: 6.6.6.6:64514 IPv6 prefix 2206::/64
                            6.6.6.6          -       100     0      65010 i
 Or-ID: 6.6.6.6 C-LST: 2.2.2.222
```

Note: Each entry in the table represents a BGP path. The path specific information includes the Route-Distinguisher and the IP prefix. Paths are either received from VPN peers or exported from local VRFs.

- The **show bgp vpn-ipv4 206.0.0.0/24 detail** and **show bgp vpn-ipv6 2206::/64 detail** commands display detailed view of the IP prefix route for ***206.0.0.0/24*** and ***2206.::/64*** of the North Edge router.

```
north-edge# show bgp vpn-ipv4 206.0.0.0/24 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for IPv4 prefix 206.0.0.0/24, Route Distinguisher:
 6.6.6.6:64514
 Paths: 1 available
  65010
    6.6.6.6 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:4364
      MPLS label: 967920
```

```
north-edge# show bgp vpn-ipv6 2206::/64 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for IPv6 prefix 2206::/64, Route Distinguisher:
 6.6.6.6:64514
 Paths: 1 available
  65010
    6.6.6.6 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:4364
      MPLS label: 965242
```

Note: The output includes the RD and IP prefix identifying the route. As seen in the output, the IPv4 VPN route is received from ***2.2.2.222*** because it is set-up to be a route-reflector, but the next hop is ***6.6.6.6***. Both are advertised with tenant VPN label ***967920*** and ***965242*** and an RT.

- The **show ip bgp vrf tenant-d** command displays the BGP table for the VRF containing the imported EVPN routes.

```
north-edge# show ip bgp vrf tenant-d
BGP routing table information for VRF tenant-d
Router identifier 1.1.1.1, local AS number 64512
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop
          Network            Next Hop         Metric  LocPref Weight Path
 * >Ec   10.255.255.0/30     6.6.6.6          -       100     0      65010 i Or-ID: 6.6.6.6 C-LST: 2.2.2.222
 *  ec   10.255.255.0/30     6.6.6.6          -       100     0      65010 i Or-ID: 6.6.6.6 C-LST: 2.2.2.222
 * >     10.255.255.4/30     10.255.255.6     -       100     0      65011 i
 * >     201.0.0.0/24        10.255.255.6     -       100     0      65011 i
 * >Ec   206.0.0.0/24        6.6.6.6          -       100     0      65010 i Or-ID: 6.6.6.6 C-LST: 2.2.2.222
 *  ec   206.0.0.0/24        6.6.6.6          -       100     0      65010 i Or-ID: 6.6.6.6 C-LST: 2.2.2.222
```

Note: Each entry in the table represent a BGP path that is either locally redistributed and received into the VRF or imported from the IPv4 VPN table. VPN routes are received from router *2.2.2.222* C-List (cluster list - basically identifying this route as from a route-reflector) with originating router being *6.6.6.6*.

Finally, let us look at the routes in the VRF tenant-d.

```
VRF: tenant-d
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort is not set

 B I    10.255.255.0/30 [200/0] via 6.6.6.6/32, IS-IS SR tunnel index 6, label
 967920
                                    via 192.168.58.12, Ethernet1/1, label 408006
 C      10.255.255.4/30 is directly connected, Ethernet6/1.120
 B E    201.0.0.0/24 [200/0] via 10.255.255.6, Ethernet6/1.120
 B I    206.0.0.0/24 [200/0] via 6.6.6.6/32, IS-IS SR tunnel index 6, label
 967920
                                  via 192.168.58.12, Ethernet1/1, label 408006
```

Note: As displayed in the highlighted route above the label stack, the route is the transport label 408006 on top (this is the label to reach ***NH 6.6.6.6***), with the ***tenant-a*** VPN label ***967920*** next in the stack, identifying the route as belonging to ***tenant-d***.

A check of the Tunnel FIB confirms that ***408006*** is the ISIS-SR LSP.

```
north-edge# show mpls tunnel fib
! 'show mpls tunnel fib' has been deprecated. Please use 'show tunnel fib [options]' moving forward.
   Tunnel Type         Index       Endpoint           Nexthop            Interface          Labels
 Forwarding
------------------- --------- ----------------- ------------------ ------------------ ----------------
   IS-IS SR IPv4       9           2.2.2.22/32        192.168.58.12       Ethernet1/1        [ 3 ]         None
   LDP                 4           2.2.2.200/32       192.168.58.12       Ethernet1/1        [ 3 ]         None
   IS-IS SR IPv4       2           2.2.2.222/32       192.168.58.12       Ethernet1/1        [ 3 ]         None
   IS-IS SR IPv4       4           3.3.3.3/32         192.168.58.12       Ethernet1/1        [ 408003 ]    None
   BGP LU              5           3.3.3.33/32        192.168.58.12       Ethernet1/1        [ 200033 ]    None
   LDP                 5           3.3.3.200/32       192.168.58.12       Ethernet1/1        [ 904099 ]    None
   IS-IS SR IPv4       8           4.4.4.4/32         192.168.58.12       Ethernet1/1        [ 408004 ]    None
   IS-IS SR IPv4       5           4.4.4.44/32        192.168.58.12       Ethernet1/1        [ 408044 ]    None
   LDP                 2           4.4.4.200/32       192.168.58.12       Ethernet1/1        [ 904098 ]    None
   IS-IS SR IPv4       3           5.5.5.5/32         192.168.58.12       Ethernet1/1        [ 408005 ]
 Primary
   BGP LU              7           5.5.5.55/32        192.168.58.12       Ethernet1/1        [ 200055 ]    None
   LDP                 3           5.5.5.200/32       192.168.58.12       Ethernet1/1        [ 904100 ]    None
   IS-IS SR IPv4       6           6.6.6.6/32         192.168.58.12       Ethernet1/1        [ 408006 ]
 Primary
   BGP LU              8           6.6.6.66/32        192.168.58.12       Ethernet1/1        [ 200066 ]    None
   LDP                 1           6.6.6.200/32       192.168.58.12       Ethernet1/1        [ 904097 ]    None
   IS-IS SR IPv4       1           23.1.1.11/32       192.168.1.154       Ethernet36/1       [ 3 ]
 Primary
   IS-IS SR IPv4       7           23.1.1.33/32       192.168.1.174       Ethernet23/1       [ 3 ]
 Primary
```

18.15.5.2 IP VPNs Over LDP

The following figures illustrate an overview of the combined control and data planes.

Figure 18-66: IPv4 VPN and IPv6 VPN Over LDP MPLS

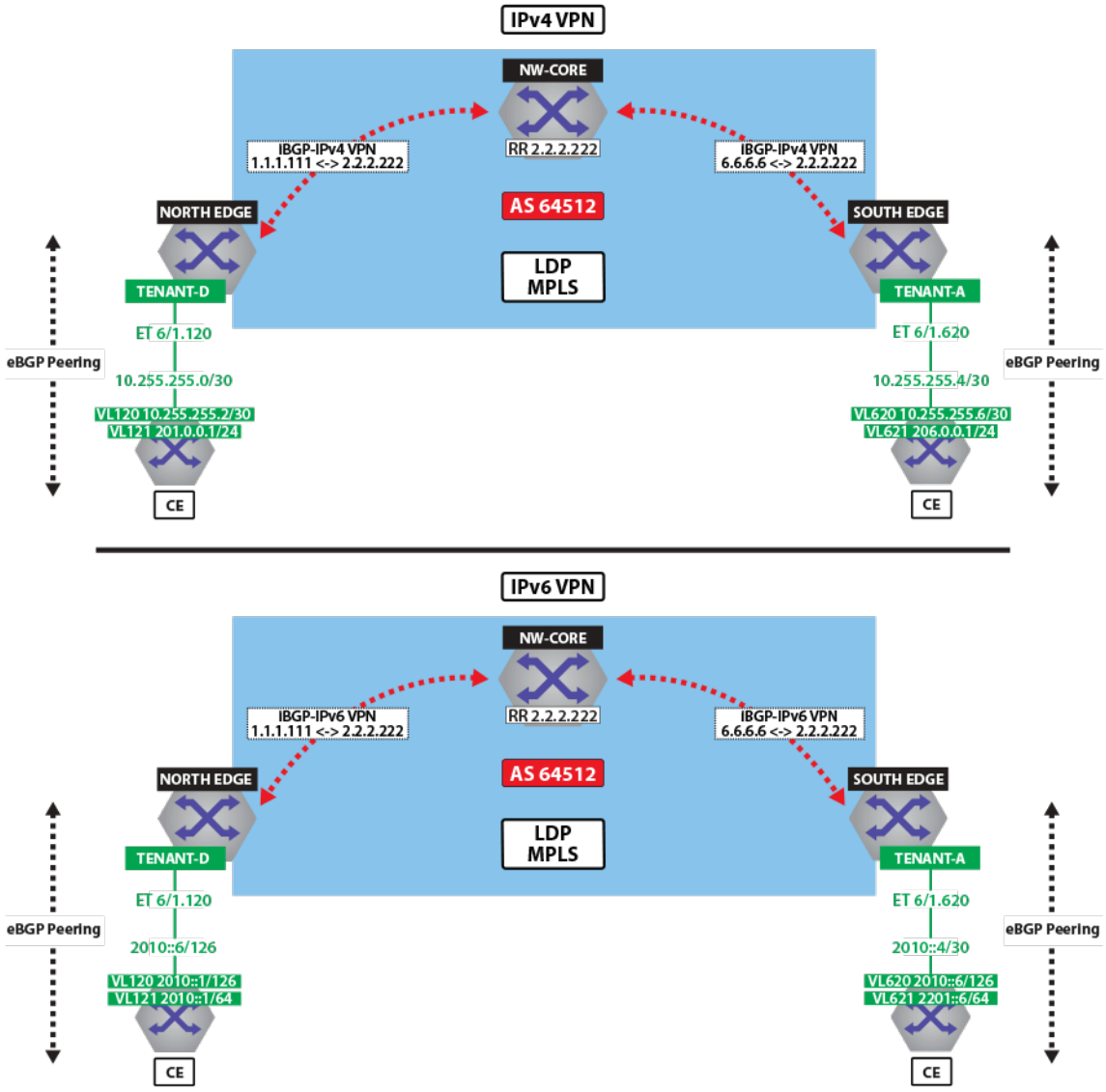


Figure 18-67: IPv4 VPN Forwarding Over LDP MPLS

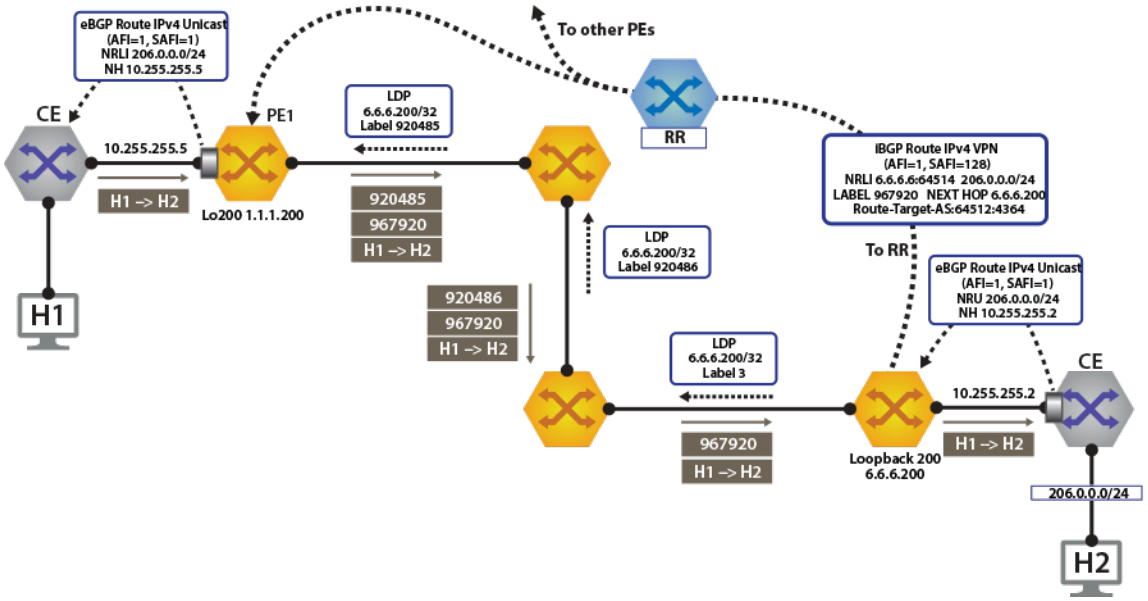
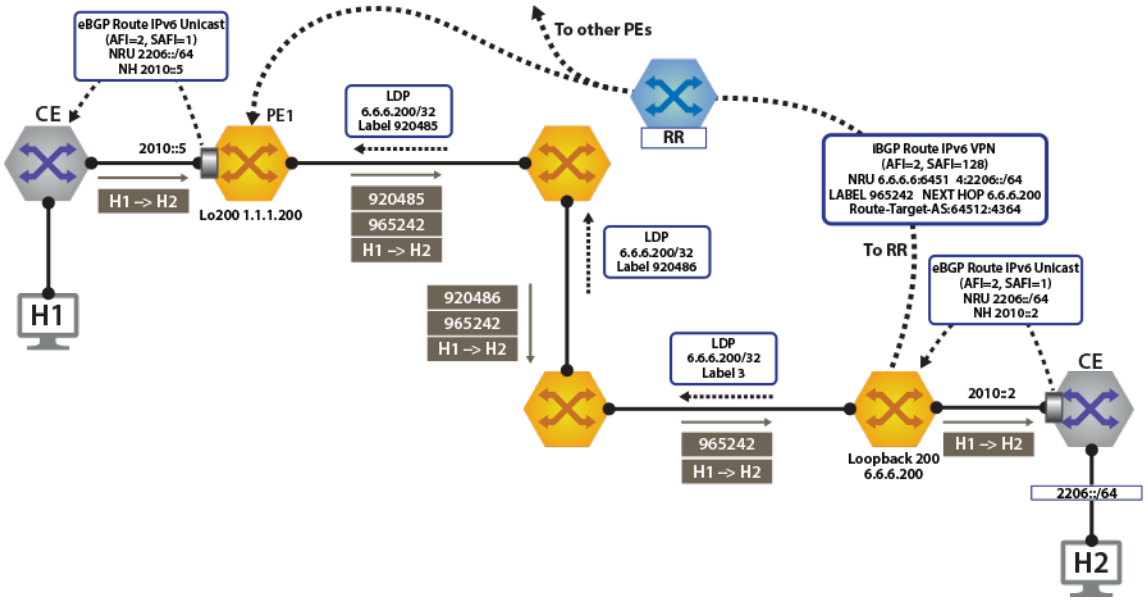


Figure 18-68: IPv6 VPN Forwarding Over LDP MPLS



To switch to using the MPLS LDP transport, we just need to change the next-hop we advertised for the VPN routes. As shown above, the next hop needs to be set to ***loopback 200*** for using the LDP LSP.

This is achieved by configuring the next-hop for the EVPN routes on both north and south edge routers.

```
router bgp 64512
   !
   address-family evpn
     neighbor default encapsulation mpls next-hop-self source-interface
 Loopback200
```

Once this is configured, we can check the BGP updates and the routes in the VRF. The output again includes the RD and IP prefix identifying the route. We now have the NH set to ***6.6.6.200*** for ***tenant-d***.

```
north-edge# show bgp vpn-ipv4 206.0.0.0/24 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for IPv4 prefix 206.0.0.0/24, Route Distinguisher:
 6.6.6.6:64514
 Paths: 1 available
  65010
    6.6.6.200 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:4364
      MPLS label: 967920
north-edge#

north-edge# show bgp vpn-ipv6 2206::/64 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for IPv6 prefix 2206::/64, Route Distinguisher:
 6.6.6.6:64514
 Paths: 1 available
  65010
    6.6.6.200 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:4364
      MPLS label: 965242
north-edge#
```

Note: The VPN label has not changed from the ISIS-SR case above (***967920*** and ***965242***), reinforcing the fact that the BGP VPN label is orthogonal to the transport label.

```
north-edge# show ip route vrf tenant-d

VRF: tenant-d
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort is not set

 B I    10.255.255.0/30 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label
 967920
                                  via 192.168.58.12, Ethernet1/1, label 904097
 C      10.255.255.4/30 is directly connected, Ethernet6/1.120
 B E    201.0.0.0/24 [200/0] via 10.255.255.6, Ethernet6/1.120
 B I    206.0.0.0/24 [200/0] via 6.6.6.200/32, LDP tunnel index 1, label 967920
                               via 192.168.58.12, Ethernet1/1, label 904097

north-edge(config-router-bgp)# show ipv6 route vrf tenant-d

VRF: tenant-d
Displaying 4 of 7 IPv6 routing table entries
Codes: C - connected, S - static, K - kernel, O3 - OSPFv3, B - BGP, R - RIP, A
 B - BGP Aggregate,
```

```
I L1 - IS-IS level 1, I L2 - IS-IS level 2, DH - DHCP, NG - Nexthop Group
 Static Route,
M - Martian, DP - Dynamic Policy Route

 B    2010::/126 [200/0]
       via 6.6.6.6/32, IS-IS SR tunnel index 6, label 965242
          via 192.168.58.12, Ethernet1/1, label 408006
 C    2010::4/126 [0/0]
       via Ethernet6/1.120, directly connected
 B    2201::/64 [200/0]
       via 2010::6, Ethernet6/1.120
 B    2206::/64 [200/0]
       via 6.6.6.6/32, IS-IS SR tunnel index 6, label 965242
          via 192.168.58.12, Ethernet1/1, label 408006
```



Note: As seen from the highlighted route above the label stack, the route are the transport label ***904097*** on top (this is the label path to reach ***NH 6.6.6.200***), with the ***tenant-d*** VPN label ***967920*** next in the stack, and identifying the route as belonging to ***tenant-a***.

A capture of the dataplane on North-Edge matching on the LDP transport label confirms the encapsulated traffic on the wire. ***904097:976920:[Source IP Address][Destination IP Address]***.

18.15.5.3 IP VPNs Over BGP-SR

The following figures illustrate an overview of the combined control and data planes.

Figure 18-69: IPv4 VPN and IPv6 VPN Over BGP-SR MPLS

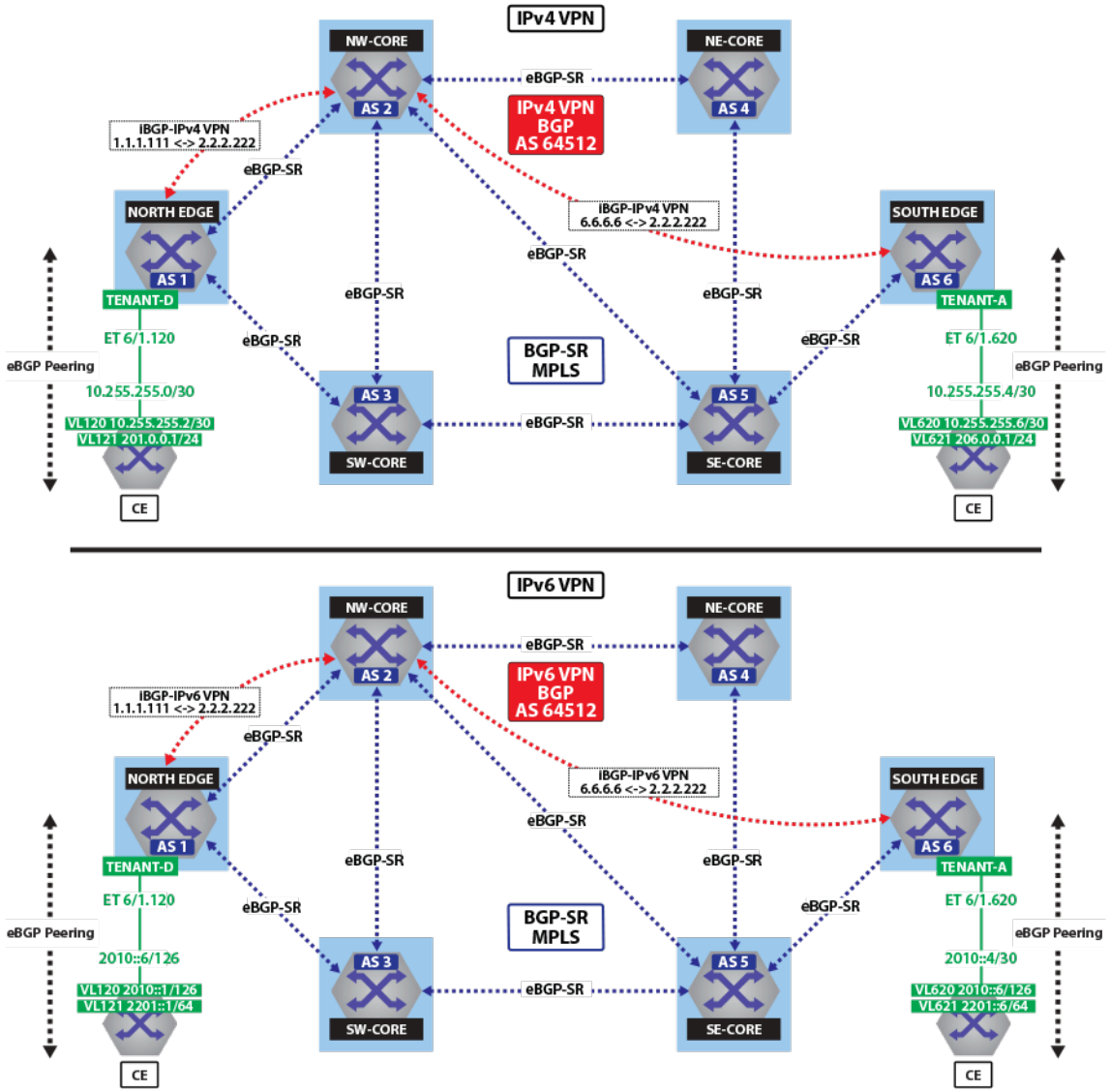


Figure 18-70: IPv4 VPN Forwarding Over BGP-SR MPLS

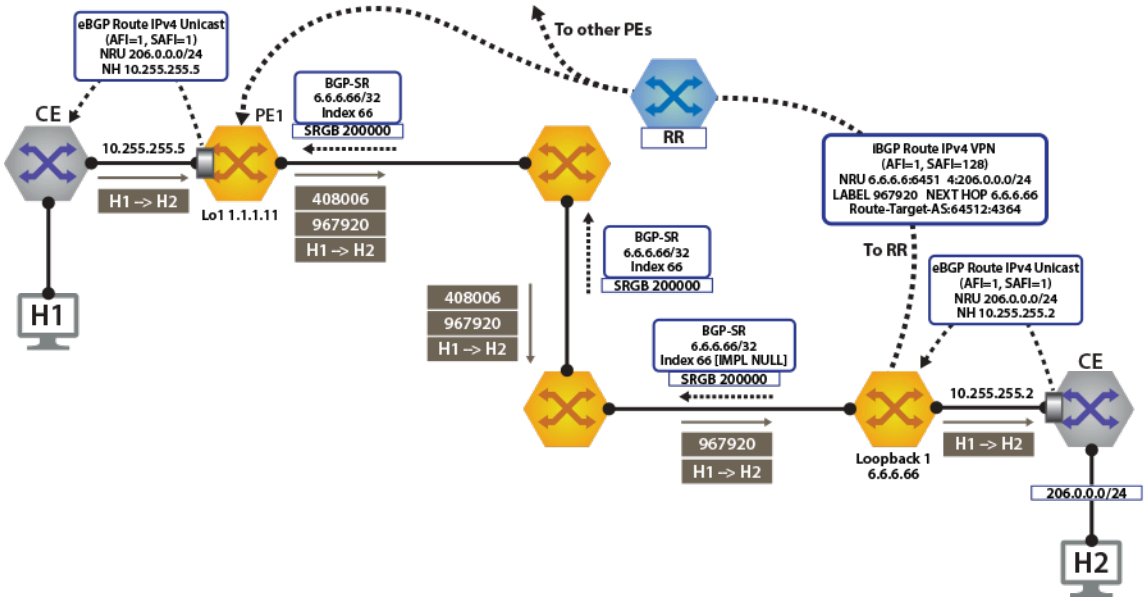
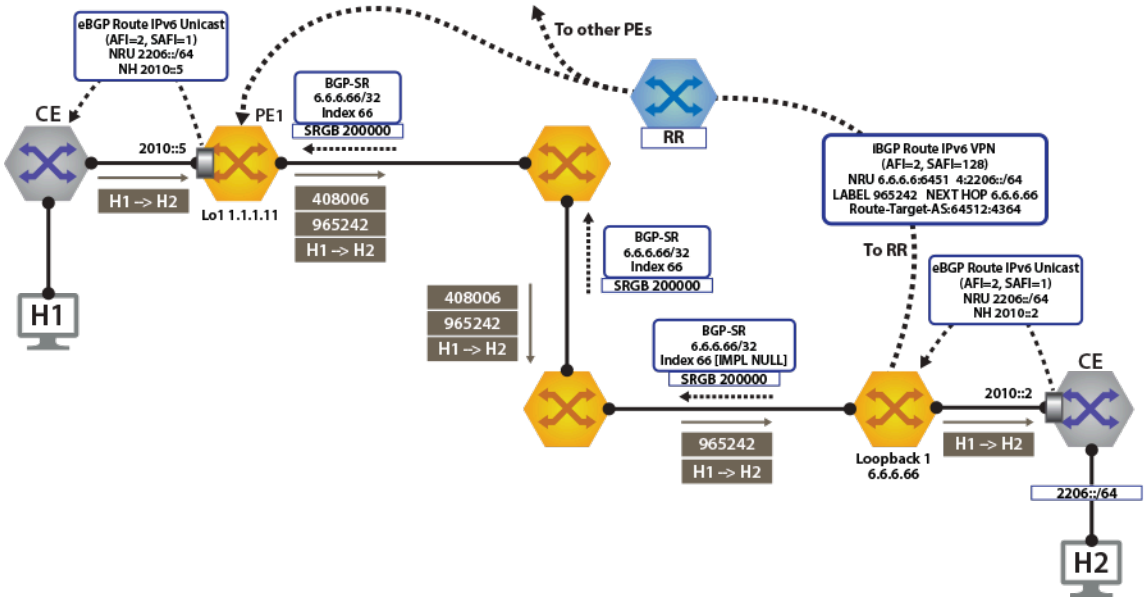


Figure 18-71: IPv6 VPN Forwarding Over BGP-SR MPLS



To switch to using the MPLS BGP-SR transport, we just need to change the next-hop we advertised for the VPN routes. As shown above, the next hop needs to be set to ***loopback 1*** for using the BGP-SR LSP.

This is achieved by configuring the next-hop for EVPN routes.

```
router bgp 64512
   !
   address-family evpn
     neighbor default encapsulation mpls next-hop-self source-interface
 Loopback1
```

Once this is configured, we can check the BGP updates and the routes in the VRF. The output again includes the RD and IP prefix identifying the route. As seen in the output, we now have the NH set to ***6.6.6.66*** for ***tenant-d***.

```
north-edge# show bgp vpn-ipv4 206.0.0.0/24 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for IPv4 prefix 206.0.0.0/24, Route Distinguisher: 6.6.6.6:64514
 Paths: 1 available
  65010
    6.6.6.66 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:4364
      MPLS label: 967920
north-edge#
north-edge#show bgp vpn-ipv6 2206::/64 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for IPv6 prefix 2206::/64, Route Distinguisher: 6.6.6.6:64514
 Paths: 1 available
  65010
    6.6.6.66 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:4364
      MPLS label: 965242
north-edge#
```

Note: The VPN label has not changed from the ISIS-SR case above (***967920*** and ***965242***), reinforcing the fact that the BGP VPN label is orthogonal to the transport label.

The output again includes the RD and IP prefix identifying the route. As seen in the output, we now have the NH set to ***6.6.6.66*** for ***tenant-d***.

```
north-edge# show bgp vpn-ipv4 206.0.0.0/24 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for IPv4 prefix 206.0.0.0/24, Route Distinguisher: 6.6.6.6:64514
 Paths: 1 available
  65010
    6.6.6.66 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:4364
      MPLS label: 967920
north-edge#
north-edge# show bgp vpn-ipv6 2206::/64 detail
BGP routing table information for VRF default
Router identifier 1.1.1.111, local AS number 64512
BGP routing table entry for IPv6 prefix 2206::/64, Route Distinguisher: 6.6.6.6:64514
 Paths: 1 available
  65010
    6.6.6.66 from 2.2.2.222 (2.2.2.222)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:64512:4364
      MPLS label: 965242
north-edge#
```

Note: The VPN label has not changed from the ISIS-SR case above (***967920*** and ***965242***), reinforcing the fact that the BGP VPN label is orthogonal to the transport label.

As displayed in the highlighted route above the label stack, the route are the transport label ***200066*** on top (this is the label path to reach ***NH 6.6.6.66***), with the ***tenant-d*** VPN label ***967920*** next in the stack, and identifying the route as belonging to ***tenant-a***.

```
north-edge# show ip route vrf tenant-d

VRF: tenant-d
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
```

```
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route

Gateway of last resort is not set

 B I    10.255.255.0/30 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8, label 967920
                                  via 192.168.58.12, Ethernet1/1, label 200066
                                  via 192.168.59.12, Ethernet2/1, label 200066
 C      10.255.255.4/30 is directly connected, Ethernet6/1.120
 B E    201.0.0.0/24 [200/0] via 10.255.255.6, Ethernet6/1.120
 B I    206.0.0.0/24 [200/0] via 6.6.6.66/32, BGP LU tunnel index 8, label 967920
                               via 192.168.58.12, Ethernet1/1, label 200066
                               via 192.168.59.12, Ethernet2/1, label 200066

north-edge(config-router-bgp)# show ipv6 route vrf tenant-d

VRF: tenant-d
Displaying 4 of 7 IPv6 routing table entries
Codes: C - connected, S - static, K - kernel, O3 - OSPFv3, B - BGP, R - RIP, A B - BGP Aggregate,
 I L1 -
IS-IS level 1, I L2 - IS-IS level 2, DH - DHCP, NG - Nexthop Group Static Route, M - Martian, DP -
 Dynamic
Policy Route

 B    2010::/126 [200/0]
       via 6.6.6.66/32, BGP LU tunnel index 8, label 965242
          via 192.168.58.12, Ethernet1/1, label 200066
          via 192.168.59.12, Ethernet2/1, label 200066
 C    2010::4/126 [0/0]
       via Ethernet6/1.120, directly connected
 B    2201::/64 [200/0]
       via 2010::6, Ethernet6/1.120
 B    2206::/64 [200/0]
       via 6.6.6.66/32, BGP LU tunnel index 8, label 965242
          via 192.168.58.12, Ethernet1/1, label 200066
          via 192.168.59.12, Ethernet2/1, label 200066
```

A capture of the data-plane on North-Edge matching on the BGP-SR transport label confirms the encapsulated traffic on the wire. ***200066:976920:[Source IP Address][Destination IP Address]***.

```
monitor session 1 source Ethernet1/1 tx
monitor session 1 destination Cpu

north-edge(config-router-bgp)# bash tcpdump -nei mirror0 -q -c 10 mpls 200066
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on mirror0, link-type EN10MB (Ethernet), capture size 262144 bytes
16:37:15.074916 28:99:3a:4d:3e:f1 > 28:99:3a:4d:3a:f3, MPLS unicast, length 122: MPLS (label
 200066, exp 0,
ttl 63) (label 967920, exp 0, [S], ttl 63) 10.255.255.6 > 206.0.0.1: ICMP echo request, id 22573,
 seq 1,
length 80

16:37:15.075088 28:99:3a:4d:3e:f1 > 28:99:3a:4d:3a:f3, MPLS unicast, length 122: MPLS (label
 200066, exp 0,
ttl 63) (label 967920, exp 0, [S], ttl 63) 10.255.255.6 > 206.0.0.1: ICMP echo request, id 22573,
 seq 2, length 80
```

18.16 EVPN and VCS Commands

Global Configuration Mode

- router general

Router BGP Configuration Mode

- next-hop resolution disabled
- redistribute service vxlan
- route-target
- route-target export

- route-target import
- route-target route-map
- vlan (VLAN-AWARE-Bundle configuration mode)
- vni-aware-bundle

Router BGP Address-Family Configuration Mode

- encapsulation vxlan layer-3 set next-hop igp-cost

Router General Configuration Mode

- leak routes

VCS Commands

- redistribute bgp evpn vxlan

EVPN VXLAN Commands

- bfd vtep evpn
- redistribute router-mac next-hop vtep primary

Display Commands

- show bgp evpn
- show ip bgp vrf
- show ip route vrf
- show ipv6 bgp vrf
- show ipv6 route vrf
- show l2rib input all
- show l2Rib input vxlan-control-service
- show l2rib output
- show service vxlan address-table
- show vrf leak flapping
- show vxlan control-plane

18.16.1 bfd vtep evpn

The **`bfd vtep evpn`** command to configure the BGP PIC Edge for EVPN VXLAN routes for remote VTEPs. This command is configured under the VXLAN Tunnel Interface (VTI).

The **`no bfd vtep evpn`** command removes bfd vtep evpn configuration from the ***running-config***.

Command Mode

VXLAN Tunnel Interface (VTI) Mode

Command Syntax

`bfd vtep evpn` interval ***interval-milliseconds*** min-rx ***min-rx-milliseconds*** multiplier ***multiplier-range***

`no bfd vtep evpn` interval ***interval-milliseconds*** min-rx ***min-rx-milliseconds*** multiplier ***multiplier-range***

Parameters

- **interval** Set transmit rate in milliseconds.
  - ***interval-milliseconds*** Rate in milliseconds. Value ranges from ***50*** to ***60000*** milliseconds.
- **min-rx** Set expected minimum incoming rate in milliseconds.

  - ***min-rx-milliseconds*** Rate in milliseconds. Value ranges from ***50*** to ***60000*** milliseconds.
- ***multiplier*** Sets the BFD multiplier.
  - ***multiplier-range*** The value ranges from ***3*** to ***50***.

Example

In this example (assuming symmetric configuration on other PE devices) any BFD for VXLAN session initiated on the VTI would have a detect time of ***300ms*** (interval of 100ms multiplied by 3).

```
switch(config-if-Vx1)# bfd vtep evpn interval 100 min-rx 100 multiplier 3
```

18.16.2 encapsulation vxlan layer-3 set next-hop igp-cost

The **encapsulation vxlan layer-3 set next-hop igp-cost** command configure the underlay IGP metric for the VTEP reachability to be considered for BGP best path selection in the IP VRF that is importing the EVPN route.

The **no encapsulation vxlan layer-3 set next-hop igp-cost** or **default encapsulation vxlan layer-3 set next-hop igp-cost** command removes all the IGP cost for VTEP running configurations on the switch.

Command Mode

BGP Address-Family Configuration

Command Syntax

encapsulation vxlan layer-3 set next-hop igp-cost

no encapsulation vxlan layer-3 set next-hop igp-cost

default encapsulation vxlan layer-3 set next-hop igp-cost

Examples

- The following command configures the IGP cost for VTEP feature on the switch :

```
switch(config-router-bgp)# address-family evpn
switch(config-router-bgp-af)# [no | default] encapsulation vxlan layer-3 set
 next-hop igp-cost
```

18.16.3 leak routes

The **leak routes** command configures an inter-VRF route-leaking policy to allow routes to be leaked from one VRF to another using a route map.

The **no leak routes** and **default leak routes** commands removing the specified route-leaking policy from ***running-config***.

Command Mode

Router General Configuration

Command Syntax

leak routes source-vrf *source_name* **subscribe-policy** ***route_map***

no leak routes source-vrf ***source_name***

default leak routes source-vrf ***source_name***

Example

- These commands configure routes to be leaked from VRF ***VRF1*** to VRF ***VRF2*** according to the policy described in route-map ***RM1***.

```
switch(config)# router general
switch(config-router-general)# vrf VRF2
switch(config-router-general-vrf-VRF2)# leak routes source-vrf VRF1
 subscribe-policy RM1
switch(config-router-general-vrf-VRF2)# exit
switch(config-router-general)# exit
switch(config)#
```

18.16.4 next-hop resolution disabled

The **next-hop resolution disabled** command disables the next-hop resolution in routes received from BGP-EVPN peers.

The **no next-hop resolution disabled** and the **default next-hop resolution disabled** commands enable the next-hop resolution in routes received from BGP-EVPN peers.

Command Mode

Router-BGP Address-Family Configuration

Command Syntax

next-hop resolution disabled

Example

This command disables the next-hop resolution in routes received from BGP-EVPN peers.

```
switch(config)# router bgp 65002
switch(config-router-bgp)# address-family evpn
switch(config-router-bgp-af)# next-hop resolution disabled
switch(config-router-bgp-af)#
```

18.16.5 redistribute bgp evpn vxlan

The **redistribute bgp evpn vxlan** command enables BGP-EVPN routes to be redistributed to VCS which in turn advertises them to all VTEPs within the DC.

The **no redistribute bgp evpn vxlan** and the **default redistribute bgp evpn vxlan** commands disable the redistribution of BGP-EVPN routes to VCS.

Command Mode

CVX-VXLAN Configuration

Command Syntax

redistribute bgp evpn vxlan

Example

This command enables redistribution of BGP-EVPN routes to VCS.

```
switch(config)# cvx
switch(config-cvx)# no shutdown
switch(config-cvx)# service vxlan
switch(config-cvx-vxlan)# no shutdown
switch(config-cvx-vxlan)# redistribute bgp evpn vxlan
```

18.16.6 redistribute router-mac next-hop vtep primary

Use the **redistribute router-mac next-hop vtep primary** command in the MAC VRF configuration mode to advertise an EVPN type-2 route for the VARP MAC with a nexthop of the primary VTEP IP.

Configuration Mode

MAC VRF

Command Syntax

redistribute router-mac next-hop vtep primary

Parameters

- **router-mac** Router Ethernet address.
- **next-hop** Cnfigure the advertised next-hop.
- **vtep** Associate next-hop with a VTEP.
- **primary** Use the primary VTEP IP.

Example

The following example uses the command **redistribute router-mac next-hop vtep primary** in the MAC VRF configuration.

```
switch(config)# router bgp 65003
switch(config-router-bgp-65003)# neighbor 223.255.255.1 remote-as 65001
switch(config-router-bgp-65003)# neighbor 223.255.255.2 remote-as 65002
switch(config-router-bgp-65003)# neighbor 223.255.255.4 remote-as 65003

switch(config)# vlan 100
switch(config-vlan-100)# rd 1:100
switch(config-vlan-100-rd-100)# route-target both 1:100
switch(config-vlan-100-rd-100)# redistribute learned
switch(config-vlan-100-rd-100)# redistribute router-mac next-hop vtep primary

switch(config)# vlan 200
switch(config-vlan-200)# rd 1:200
switch(config-vlan-100-rd-200)# route-target both 1:200
switch(config-vlan-100-rd-200)# redistribute learned
switch(config-vlan-100-rd-200)# redistribute router-mac next-hop vtep primary
```

18.16.7 redistribute service vxlan

The **redistribute service vxlan** command enables BGP to redistribute the Layer 2 bridging information received from VCS.

The **no redistribute service vxlan** and the **default redistribute service vxlan** commands disable the redistribution of the bridging information received from VCS.

Command Mode

Router-BGP VNI Configuration

Command Syntax

redistribute service vxlan

Example

This command enables redistribution of the Layer 2 bridging information received from VCS.

```
switch(config)# router bgp 100
```

```
switch(config-router-bgp)# vni-aware-bundle bundle1
switch(config-macvrf-bundle1)# redistribute service vxlan
```

18.16.8 route-target

The **route-target** command configures a well-known extended community that is used by BGP-EVPN to export routes from or import routes into MAC-VRF.

The **no route-target** and **default route-target** commands delete the route-target configuration.

Command Mode

Router-BGP VNI Configuration

Syntax

route-target {export | import | both} rt

no route-target

default route-target

Parameters

- **export** configures a well-known extended community that is attached to the routes exported by BGP-EVPN.
- **import** configures a well known extended community that identifies the received routes that need to be imported into the MAC-VRF specified by the VNI bundle.
- **both** configures the same extended community for import and export of routes.
- **rt** route-target extended community.

This command configures a well-known extended community for import and export of routes.

```
switch(config)# router bgp 100
switch(config-router-bgp)# vni-aware-bundle bundle1
switch(config-macvrf-bundle1)# route-target both 503:12
switch(config-macvrf-bundle1)#
```

18.16.9 route-target export

The **route-target export** command allows the user to export routes from a VRF to the local VPN table using the route target extended community list.

The **no route-target export** and **default route-target export** commands remove the routes from the VPN table.

Command Mode

Router-BGP VNI Configuration

Syntax

route-target export [***evpn*** | ***vpn-ipv4*** | ***vpn-ipv6***] ***RT***

no route-target export

default route-target export

Parameters

- ***evpn*** EVPN address family.
- ***vpn-ipv4*** MPLS L3 VPN IPv4 unicast address family.
- ***vpn-ipv6*** MPLS L3 VPN IPv6 unicast address family.
- ***RT*** route-target extended community.

Examples

- These commands export routes from ***vrf-red*** to the VPN table.

```
switch(config)# service routing protocols model multi-agent
switch(config)# mpls ip
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-red
switch(config-router-bgp-vrf-vrf-red)# rd 1:1
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv4 10:10
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv6 10:20
```

- These commands export routes from ***vrf-red*** to the EVPN table.

```
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-red
switch(config-router-bgp-vrf-vrf-red)# rd 1:1
switch(config-router-bgp-vrf-vrf-red)# route-target export evpn 10:1
```

18.16.10 route-target import

The **route-target import** command allows the user to import route target extended community lists from the local VPN table to the target VRF.

The **no route-target import** and **default route-target import** commands remove the routes from the VPN table.

Command Mode

Router-BGP VNI Configuration

Syntax

route-target import [***evpn*** | ***vpn-ipv4*** | ***vpn-ipv6***] ***RT***

no route-target import

default route-target import

Parameters

- ***evpn*** EVPN address family.
- ***vpn-ipv4*** MPLS L3 VPN IPv4 unicast address family.
- ***vpn-ipv6*** MPLS L3 VPN IPv6 unicast address family.
- ***RT*** route-target extended community.

Examples

- These commands import routes from the VPN table to ***vrf-blue***.

```
switch(config)# service routing protocols model multi-agent
switch(config)# mpls ip
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 2:2
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv4 10:10
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv6 10:20
```

- These commands import routes from the EVPN table to ***vrf-blue***.

```
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 2:2
```

```
switch(config-router-bgp-vrf-vrf-blue)# route-target import evpn 10:1
```

18.16.11 route-target route-map

The **route-target route-map** command allows the user to export and import route target extended community lists from one VRF to another using route maps.

The **no route-target route-map** and **default route-target route-map** commands remove the routes from the VPN table.

Command Mode

Router-BGP VNI Configuration

Syntax

route-target {import | export} [***evpn*** | ***vpn-ipv4*** | ***vpn-ipv6***] **route-map** ***RM***

no route-target route-map

default route-target route-map

Parameters

- ***evpn*** EVPN address family.
- ***vpn-ipv4*** MPLS L3 VPN IPv4 unicast address family.
- ***vpn-ipv6*** MPLS L3 VPN IPv6 unicast address family.
- ***RM*** route-map extended community.

Examples

- These commands export routes from ***vrf-red*** to the VPN table.

```
switch(config)# service routing protocols model multi-agent
switch(config)# mpls ip
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-red
switch(config-router-bgp-vrf-vrf-red)# rd 1:1
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv4 10:10
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv6 10:20
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv4 route-
map EXPORT_V4_ROUTES_T0_VPN_TABLE
switch(config-router-bgp-vrf-vrf-red)# route-target export vpn-ipv6 route-
map EXPORT_V6_ROUTES_T0_VPN_TABLE
```

- These commands export routes from ***vrf-red*** to the EVPN table.

```
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-red
switch(config-router-bgp-vrf-vrf-red)# rd 1:1
switch(config-router-bgp-vrf-vrf-red)# route-target export evpn 10:1
switch(config-router-bgp-vrf-vrf-red)# route-target export evpn route-map
 EXPORT_ROUTES_T0_EVPN_TABLE
```

- These commands import routes from the VPN table to ***vrf-blue***.

```
switch(config)# service routing protocols model multi-agent
switch(config)# mpls ip
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 1:1
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv4 10:10
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv6 10:20
```

```
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv4 route-
map IMPORT_V4_ROUTES_VPN_TABLE
switch(config-router-bgp-vrf-vrf-blue)# route-target import vpn-ipv6 route-
map IMPORT_V6_ROUTES_VPN_TABLE
```

- These commands import routes from the EVPN table to ***vrf-blue***.

```
switch(config)# router bgp 65001
switch(config-router-bgp)# vrf vrf-blue
switch(config-router-bgp-vrf-vrf-blue)# rd 2:2
switch(config-router-bgp-vrf-vrf-blue)# route-target import evpn 10:1
switch(config-router-bgp-vrf-vrf-blue)# route-target import evpn route-map
 IMPORT_ROUTES_FROM_EVPN_TABLE
```

18.16.12 router general

The **router general** command places the switch in Router-General Configuration Mode for the configuration of protocol-independent routing.

The **no router general** and **default router general** commands remove all protocol-independent routing configuration from ***running-config***.

Command Mode

Global Configuration

Command Syntax

router general

no router general

default router general

Example

- These commands place the switch in Router-General Configuration Mode and display the commands available there.

```
switch(config)# router general
switch(config-router-general)# ?
  command             Configuration command
  control-functions   Routing control functions configuration
  hardware            Configure hardware-specific parameters
  next-hops           Next hop configuration
  rib                 Routing table
  route               Route commands
  route-map           Route-map source configuration
  router-id           Configure a general router ID for all routing processes
  segment-routing     Segment Routing configuration
  software            Software configuration
  vrf                 Enter VRF sub-mode

switch(config-router-general)#
```

18.16.13 show bgp evpn

The **show bgp evpn** command displays information about the BGP-EVPN routes of the switch.

Command Mode

Global Configuration

Command Syntax

show bgp evpn [community | detail | esi esid | extcommunity | host-flap | instance | large-community ***AS:nn:nn*** | next-hop | rt | admin:local-assignment | route-type | summary | vni ***vni_num***]

Parameters

- ***no parameters*** displays all routes of the switch.
- **community** displays routes filtered by the specified community. Options include:
  - **GSHUT** well known GSHUT community.
  - ***aa:nn*** AS and network number, separated by colon. The value ranges from ***1*** to ***4294967295***.
  - **internet** advertises route to the Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise the route to any peer.
  - **no-export** advertises route only within the BGP-EVPN AS boundary.
  - ***comm_num*** community number. Values range from ***1*** to ***4294967040***.
- **detail** displays detailed information of routes.
- **esi esid** displays routes filtered by the specified Ethernet Segment Identifier (ESI).
- **extcommunity** displays routes that match with BGP or VPN extended community list. Options include:
  - **esi-label esid** displays routes filtered by the specified value of ESI label. The value ranges from ***0*** to ***16777215***.
  - **mac-mobility** displays routes filtered by the specified MAC mobility.
  - **rt** displays routes filtered by the specified route target.
  - **tunnel-encap vxlan** displays routes filtered by the VXLAN tunnel encapsulation.
  - **router-mac *H.H.H*** displays routes filtered by the specified router MAC address.
- **host-flap** displays routes that contains MAC addresses that are blacklisted due to duplication.
- **instance** displays routes with EVPN instances.
- **large-community *AS:nn:nn*** displays routes filtered by the specified large community.
- **next-hop** displays routes filtered by next-hop IPv4 or IPv6 addresses of remote VTEP.
- **rd admin:local-assignment** displays routes filtered by the specified Route Distinguisher (RD).
- **route-type** displays routes filtered by NLRI route type.
- **summary** displays summary of routes.
- **vni *vni_num*** displays routes filtered by the specified VXLAN Network Identifier (VNI). Value ranges from ***1*** to ***4294967294***.

Examples

- This command displays BGP-EVPN routes filtered by the ***VNI 3011***.

```
switch(config-router-bgp-af)# show bgp evpn vni 3011
BGP routing table information for VRF default
Router identifier 2.0.2.2, local AS number 65002
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e
 - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local
 Nexthop

         Network                Next Hop            Metric  LocPref Weight  Path
 * >Ec   RD: 3.3.3.1:3011 auto-discovery 0 009a:f13b:53bb:8800:0000
                                1.1.1.1               -       100     0       65999 65001 i
 *  ec   RD: 3.3.3.1:3011 auto-discovery 0 009a:f13b:53bb:8800:0000
                                1.1.1.1               -       100     0       65999 65001 i
 * >     RD: 3.3.3.2:3011 auto-discovery 0 009a:f13b:53bb:8800:0000
                                -                     -       -       0        i
 * >Ec   RD: 3.3.3.1:3011 imet 1.1.1.1
                                1.1.1.1               -       100     0       65999 65001 i
 *  ec   RD: 3.3.3.1:3011 imet 1.1.1.1
                                1.1.1.1               -       100     0       65999 65001 i
 * >     RD: 3.3.3.2:3011 imet 1.1.1.2
                                -                     -       -       0        i
```

```
cvx(config-router-bgp-af)#
```

- This command displays the prefixes that are exported to the respective VPN table, along with the route distinguisher.

```
switch(config)# show bgp evpn
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 65001
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e
 - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local
 Nexthop
         Network                Next Hop            Metric  LocPref Weight  Path
 * >     RD: 400:1 ip-prefix 45.0.0.1/32
                                -                     -       -       0        i
 * >     RD: 400:1 ip-prefix 52.0.0.1/32
                                -                     -       -       0        i
 * >     RD: 400:1 ip-prefix 120.0.0.0/24
                                -                     -       -       0        i
 * >     RD: 400:1 ip-prefix 130.0.0.0/24
                                -                     -       -       0        i
 * >     RD: 400:1 ip-prefix 130.0.1.0/24
```

18.16.14 show ip bgp vrf

The **show ip bgp vrf** command displays the type of VPN from the imported route. It shows an indication that the IPv4 route has been leaked and source VRF information is displayed.

Command Mode

Global Configuration

Command Syntax

show ip bgp vrf {***vrf_name*** | all | default}

Parameters

- ***vrf_name*** name of the VRF.
- **all** displays summary of all VRFs.
- **default** default virtual routing and forwarding instance.

Example

This command displays the leaked and source VRF information.

```
switch(config)# show ip bgp 13.0.0.0/24 vrf vrf-blue
BGP routing table information for VRF vrf-blue
Router identifier 5.0.0.2, local AS number 65001
BGP routing table entry for 130.110.61.0/24
   4.0.0.3 from 4.0.0.3 (52.0.0.1), imported EVPN route, RD 400:1
      Origin IGP, metric -, localpref 100, weight 0, valid, external,best
      Extended Community: Route-Target-AS:4000:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:74:83:ef:0b:70:f3
      Leaked from VRF vrf-red
```

18.16.15 show ip route vrf

The **show ip route vrf** command displays leaked prefixes with the label **L** in the output that indicates that the IPv4 route has been leaked. It also displays information about the source VRF from which these prefixes have been leaked.

Command Mode

Global Configuration

Command Syntax

show ip route vrf {***vrf_name*** | all}

Parameters

- ***vrf_name*** name of the VRF.
- **all** displays summary of all VRFs.

Example

These commands display the OSPF or OSPFV3 leaked routes as **redistribute ospf** and **redistribute ospfv3** are configured on the source VRF ***vrf-red***.

```
switch(config)# show ip route vrf vrf-blue
VRF: vrf-blue
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked
Gateway of last resort is not set
C       5.0.0.2/31 is directly connected, Ethernet14
B L  57.0.0.3/32 [200/0] (source VRF vrf-red) via 4.0.0.3, Ethernet11
B L  45.0.0.1/32 [200/0] (source VRF vrf-red) via 4.0.0.3, Ethernet11
B L  52.0.0.1/32 [200/0] (source VRF vrf-red) via 4.0.0.3, Ethernet11
B L  120.0.0.0/24 [200/0] (source VRF vrf-red) via 4.0.0.3, Ethernet11
B L  130.0.0.0/24 [200/0] (source VRF vrf-red) via 4.0.0.3, Ethernet11
B L  130.0.1.0/24 [200/0] (source VRF vrf-red) via 4.0.0.3, Ethernet11
B L  130.0.2.0/24 [200/0] (source VRF vrf-red) via 4.0.0.3, Ethernet11
B L  130.0.3.0/24 [200/0] (source VRF vrf-red) via 4.0.0.3, Ethernet11
```

18.16.16 show ipv6 bgp vrf

The **show ipv6 bgp vrf** command displays the type of VPN from the imported route. It shows an indication that the IPv6 route has been leaked and source VRF information is displayed.

Command Mode

Global Configuration

Command Syntax

show ipv6 bgp vrf {***vrf_name*** | all | default}

Parameters

- ***vrf_name*** name of the VRF.
- **all** displays summary of all VRFs.
- **default** default virtual routing and forwarding instance.

Example

This command displays the leaked and source VRF information.

```
switch(config)# show ipv6 bgp 2001:10:1:0::102/64 vrf default
BGP routing table information for VRF default
Router identifier 218.218.218.218, local AS number 34
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
```

```
                    S - Stale, c - Contributing to ECMP, b - backup, L -
labeled-unicast
                    % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

         Network                Next Hop              Metric  LocPref Weight  Path
 * >     2000:0:14:120::/64     2001:db8:1111:9000::  -       100     109     i
 *       2000:0:14:120::/64     2001:db8:156:1010::2  -       100     0       i
 *       2000:0:14:120::/64     2001:db8:152:1010::2  -       100     0       i
 *       2000:0:14:120::/64     2001:db8:203:1010::2  -       100     0       i
```

18.16.17 show ipv6 route vrf

The **show ipv6 route vrf** command displays leaked prefixes with the label **L** in the output that indicates that the IPv6 route has been leaked. It also displays information about the source VRF from which these prefixes have been leaked.

Command Mode

Global Configuration

Command Syntax

show ipv6 route vrf{***vrf_name*** | all}

Parameters

- ***vrf_name*** name of the VRF.
- **all** displays summary of all VRFs.

Example

These commands display the OSPF or OSPFV3 leaked routes as **redistribute ospf** and **redistribute ospfv3** are configured on the source VRF ***vrf-red***.

```
switch(config)# show ipv6 route vrf vrf-blue
VRF: vrf-blue
Displaying 802 of 802 IPv6 routing table entries
Codes: C - connected, S - static, K - kernel, O3 - OSPFv3, B - BGP, R - RIP, A
 B - BGP Aggregate,
I L1 - IS-IS level 1, I L2 - IS-IS level 2, DH - DHCP, NG - Nexthop Group
 Static Route, M - Martian,
DP - Dynamic Policy Route, L - VRF Leaked
  B L   18::1/128 [200/0] (source VRF vrf-red)
        via 4::3, Ethernet11
  B  L  6::2/127 [200/0] (source VRF vrf-red)
        via fe80::7683:efff:fe0b:963d, Ethernet11
  B  L  45::1/128 [200/0] (source VRF vrf-red)
        via fe80::7683:efff:fe0b:963d, Ethernet11
  B  L  130::/64 [200/0] (source VRF vrf-red)
        via fe80::7683:efff:fe0b:963d, Ethernet11
  B  L  130:0:0:1::/64 [200/0] (source VRF vrf-red)
        via fe80::7683:efff:fe0b:963d, Ethernet11
  B  L  130:0:0:2::/64 [200/0] (source VRF vrf-red)
        via fe80::7683:efff:fe0b:963d, Ethernet11
  B  L  130:0:0:3::/64 [200/0] (source VRF vrf-red)
```

18.16.18 show l2rib input all

Use the **show l2rib input all** command to display the Layer 2 RIB input tables migration process.

Command Mode

EXEC

Command Syntax

`show l2rib input all` [detail | floodset [vlan | vtep] | interface [Ethernet | Port-Channel] | mac ***H.H.H*** | vlan ***vlan_id*** | vtep [***A.B.C.D*** | ***A:B:C:D:E:F:G:H***]

Parameters

- **detail** Displays a more comprehensive output.
- **floodset** Displays L2Rib floodset from input source.
  - **vlan** Filter by VLAN ID.
  - **vtep** Filter by VTEP.
- **interface** Filter by destination interface.
  - **Ethernet** Ethernet interface.
  - **Port-Channel** Link Aggregation Group (LAG).
- **mac *H.H.H*** Filter by MAC address.
- **vlan *vlan_id*** Filter by VLAN ID.
- **vtep** Filter by VTEP.
  - ***A.B.C.D*** IP address of VTEP.
  - ***A:B:C:D:E:F:G:H*** IP address of VTEP.

18.16.19 show l2Rib input vxlan-control-service

Use the **`show l2Rib input vxlan-control-service`** to display the migration processes for the VXLAN control service.

Command Mode

EXEC

Command Syntax

`show l2Rib input vxlan-control-service` [detail | floodset [vlan | vtep] interface [Ethernet | Port-Channel]| mac ***H.H.H*** | vlan ***vlan_id*** | vtep [***A.B.C.D*** | ***A:B:C:D:E:F:G:H***]]

Parameters

- **detail** Displays a more comprehensive output.
- **floodset** Displays L2Rib floodset from input source.
  - **vlan** Filter by VLAN ID.
  - **vtep** Filter by VTEP.
- **interface** Filter by destination interface.
  - **Ethernet** Ethernet interface.
  - **Port-Channel** Link Aggregation Group (LAG).
- **mac** Filter by MAC address.
  - ***H.H.H*** Ethernet address.
- **vlan *vlan_id*** Filter by VLAN ID.
- **vtep** Filter by VTEP.
  - ***A.B.C.D*** IP address of VTEP.
  - ***A:B:C:D:E:F:G:H*** IP address of VTEP.

18.16.20 show l2rib output

Use the **`show l2rib output`** command to display the the Layer 2 RIB utput tables migration process.

Command Mode

EXEC

Command Syntax

`show l2rib output` [detail | floodset [vlan | vtep] | mac H.H.H | vlan ***vlan_id*** | vtep [***A.B.C.D*** | ***A:B:C:D:E:F:G***]

Parameters

- **detail** Displays a more comprehensive output.
- **floodset** Displays L2Rib floodset from input source.
  - **vlan** Filter by VLAN ID.
  - **vtep** Filter by VTEP.
- **interface** Filter by destination interface.
  - **Ethernet** Ethernet interface.
  - **Port-Channel** Link Aggregation Group (LAG).
- **mac *H.H.H*** Filter by MAC address.
- **vlan *vlan_id*** Filter by VLAN ID.
- **vtep** Filter by VTEP.
  - ***A.B.C.D*** IP address of VTEP.
  - ***A:B:C:D:E:F:G:H*** IP address of VTEP.

18.16.21 show service vxlan address-table

The **`show service vxlan address-table`** command displays route entries in the MAC forwarding table that are added through the CVX.

Command Mode

CVX Global Configuration

Command Syntax

`show service vxlan address-table` {advertised | received}[***address H.H.H*** | evpn | hsc | mss | switch [***Word*** | all] | vni ***vnid*** | vtep ***A.B.C.D***]

Parameters

- **advertised** displays the advertised route entries in the MAC forwarding table.
- **received** displays the received route entries in the MAC forwarding table.
- **address *H.H.H*** displays route entries that are filtered by the specified MAC addresses.
- **evpn** displays route entries filtered by BGP-EVPN.
- **hsc** displays route entries filtered by Hardware Switch Controller (HSC).
- **mss** displays route entries filtered by Macro Segmentation Service (MSS).
- **switch** displays route entries that are filtered by the specified switch or all switches. Options include:
  - ***Word*** Hostname, IP address or ID of the switch.
  - **all** all switches.
- **vni *vnid*** displays route entries filtered by the specified VXLAN Network Identifier (VNI). Value ranges from ***1*** to ***4294967294***.
- **vtep *A.B.C.D*** displays route entries filtered by the specified IP address of the remote Virtual Tunnel End Point (VTEP).

Examples

- This command displays the route entries in MAC forwarding table advertised to BGP-EVPN peers.

```
cvx# show service vxlan address-table advertised evpn
```

```
         Advertised Mac Address Table
-------------------------------------------------------
VNI          Mac Address    VTEP         Moves
-----------  -----------------  --------------  -----
1000         02:01:62:01:00:00 10.0.0.1         1
Total Mac Addresses for this criterion: 1

         Advertised Flood Table
-------------------------------------------------
VNI         Mac Address       VTEP(s)
----------  -----------------  ------------------------------
1000        00:00:00:00:00:00  10.0.0.1     10.0.0.2
Total Mac Addresses for this criterion: 1 cvx#
```

- This command displays the route entries in MAC forwarding table received from BGP-EVPN peers.

```
switch# show service vxlan address-table received evpn

      Received Mac Address Table
-----------------------------------------------------------
Source      VNI      Mac Address        VTEP       Moves
----------- -------- -----------------  ---------- -----
EVPN        1000     02:01:62:02:00:00  10.0.0.3   1
Total Mac Addresses for this criterion: 1

      Received Flood Table
-------------------------------------------------------------
Source          VNI      Mac Address    VTEP
----------------- ----------- ----------------- --------------
EVPN            1000     00:00:00:00:00:00 10.0.0.3
EVPN            1000     00:00:00:00:00:00 10.0.0.4
Total Mac Addresses for this criterion: 2 switch#
```

18.16.22 show vrf leak flapping

The **show vrf leak flapping** command displays the flapping prefixes of the routes leaked from one VRF to another VRF. Routes that are detected as “flapping” are blocked considering the future leaking policy execution.

Command Mode

EXEC

Command Syntax

show vrf leak flapping

Parameters

- **destination** displays flapping prefixes destined to a VRF.
- **prefix** displays flapping routes for a prefix.
- **source** displays flapping prefixes sourced from a VRF.
- **vrf** displays flapping prefixes associated with a VRF.

Example

This command displays the flapping prefixes of the leaked routes.

```
switch# show vrf leak flapping

  Age       Source VRF       Destination VRF       Prefix            Created At
-------- ---------------- --------------------- ----------------- -------------
  141       VRF1             VRF2                 10.0.2.0/24    3357281.40992
```

18.16.23 show vxlan control-plane

To enable the user to quickly view which control planes and sources are importing reachability information into which VLANs, use the following show command:

Command Mode

EXEC

`show vxlan control-plane` [export vlan [***$*** | ***vlan_id***]|import vlan [***$*** | ***vlan_id***]][vlan [***$*** | ***vlan_id***]

Parameters

- **export** Restricts output to VLANs exporting to a control plane.
  - **vlan** Specifies specific VLANs to display control plane information.
    - ***$*** List end.
    - ***vlan_id*** VLAN ID. Range ***1-4094***.
- **import** Restricts output to VLANs importing from a control plane.
  - ***$*** List end.
  - ***vlan_id*** VLAN ID. Range ***1-4094***.
- **vlan** Specifies specific VLANs to display control plane information.
  - ***$*** List end.
  - ***vlan_id*** VLAN ID. Range ***1-4094***.

18.16.24 vlan (VLAN-AWARE-Bundle configuration mode)

The **`vlan add/ remove`** command configures the VLAN range string in a VLAN-AWARE-BUNDLE. When the add and remove options are not used, the currently configured range is replaced by the new one . The add and remove options update the currently configured range-aware bundles by adding or removing the provided VLAN range from the currently configured range respectively.

Command Mode

VLAN-AWARE-BUNDLE Configuration

Command Syntax

`vlan` [add | remove] ***range***

Parameters

range is the range specified.

Example

These commands take an existing vlan-aware-bundle configuration, ***corporate_100***, and modify it with the add and remove options.

```
(Existing configuration for corporate_100)
vlan-aware-bundle corporate_100
  rd 1.1.1.1:100
  route-target both 100:100
  vlan 1-10

switch(config-vlan-aware-bundle)#vlan add 1000

(Modified configuration for corporate_100)
vlan-aware-bundle corporate_100
  rd 1.1.1.1:100
  route-target both 100:100
  vlan 1-10, 1000
```

```
switch(config-vlan-aware-bundle)#vlan remove 5

(Updated configuration for corporate_100)
vlan-aware-bundle corporate_100
  rd 1.1.1.1:100
  route-target both 100:100
  vlan 1-4, 6-10, 1000
```

18.16.25 vni-aware-bundle

The **vni-aware-bundle** command configures a BGP MAC-VRF containing Layer 2 routes from a group of VXLAN Network Identifiers (VNI).

Command Mode

Router BGP Configuration

Command Syntax

vni-aware-bundle ***vni_bundle_name***

Parameter

vni_bundle_name VNI bundle name.

Example

This command configures MAC-VRF BGP to support VNI bundle1.

```
cvx(config)# router bgp 100
cvx(config-router-bgp)# vni-aware-bundle bundle1
cvx(config-macvrf-bundle1)#
```

Chapter 19

Multiprotocol Label Switching (MPLS)

Tunneling protocols encapsulate packets of a different protocol as the payload of a larger frame for delivery within networks utilizing the encapsulating protocol. Tunneling facilitates the delivery of payload over an incompatible delivery network and creates a secure path through an untrusted network. Protocols that this chapter describes include MPLS, Decap Groups, and Nexthop Groups.

Sections in this chapter include:

- MPLS
- BGP/MPLS L3 VPN
- EVPN MPLS Shared ESI Label
- RSVP-TE LSR
- RSVP-TE LER
- LDP Pseudowire
- LDP Entropy Label
- MPLS Commands

19.1 MPLS

These sections describe the Arista MPLS implementation:

- MPLS Description
- MPLS Configuration

19.1.1 MPLS Description

19.1.1.1 MPLS Overview

Multiprotocol Label Switching (MPLS) is a networking process that replaces complete network addresses with short path labels for directing data packets to network nodes. The labels identify virtual links (paths) between distant nodes rather than endpoints. MPLS is scalable and protocol-independent. Data packets are assigned labels, which are used to determine packet forwarding destinations without examining the packet.

Arista switches utilize MPLS to improve efficiency and control from servers through data centers and to the WAN. The MPLS implementation supports static MPLS tunneling that is manually configured on each switch or established over a network by an SDN controller. The configuration is specified by a set of rules that filter packets based on matching criteria. Each rule applies MPLS-related actions to packets that match the rule's criteria. Each rule includes a metric that the switch uses to select an action when multiple rules match a packet.

19.1.1.2 MPLS Implementation

MPLS static rule parameters contain the following:

- A 20-bit value that is compared to the top header label of each MPLS packet. Other rule parameters may be applied to packets whose top label match this value.
- A nexthop location that specifies the packet’s next destination (IPv4 or IPv6) and the interface through which the switch forwards the packet.
- An MPLS label stack management action that is performed on filtered packets:

- **pop-payload**: removes the top label from stack; this terminates an Label-Switched Path (LSP).
- **swap-label**: replaces top label with a specified new label; this passes a packet along an LSP.

- A rule metric that the switch uses to select a rule when multiple rules match an MPLS packet.

Packets that do not match any MPLS rules are dropped.

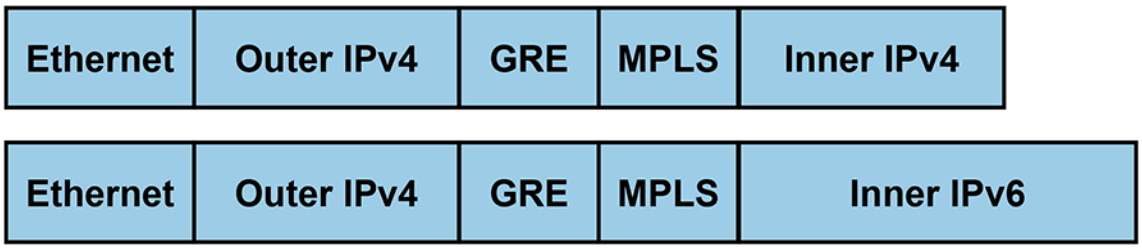
19.1.1.3 MPLSoGRE Filtered Mirroring

In MPLS over Generic Routing Encapsulation (MPLSoGRE) filtered mirroring, IPv4 over MPLS over GRE (IPv4oMPLSoGRE) and IPv6 over MPLS over GRE (IPv6oMPLSoGRE) packets that enter a GRE tunnel endpoint on which MPLS lookup is performed, are selected for mirroring based on the destination IP address field in the inner IPv4 or IPv6 header.

Note: These packets are not selected for mirroring if they are forwarded based on either the L2 or outer L3 header destination address.

the image below shows the header format of the packets that are selected for mirroring.

Figure 19-1: Header Format of Packets



When mirroring to a GRE tunnel, the payload of the outgoing GRE packet contains the payload of the incoming source packet starting from the MPLS header. L2 and outer L3 headers are stripped from the mirror copy. When the MPLS lookup fails, the packet is still eligible for mirroring based on the selection criteria defined in the ACL.

19.1.1.4 Mixed TTL/DSCP in MPLS Tunnel Termination

A Label Edge Router (LER) decapsulating an MPLS packet may choose to source the TTL and DSCP fields from either the MPLS header or the original inner IPv4 header. However, some applications need the TTL field value from the MPLS header, while retaining the DSCP from the original IPv4 header. Mixed TTL/DSCP in MPLS Tunnel Termination enables tunnel termination to select this specific mix of two modes.

This section contains the following topics:

- Configuration
- Packet Support for IPv6

19.1.1.4.1 Configuration

Mixed TTL/DSCP in MPLS tunnel termination is intended to be used with MPLS VRF decapsulation. Therefore, a static VRF-to-MPLS-label map is expected to be provided, similar to the following:

```
switch(config)# mpls ip
switch(config)# mpls static vrf-label mpls-label vrf vrf-name
switch(config)# ip routing vrf vrf-name
```

TTL=uniform and DSCP=pipe options have been added to the existing **mpls tunnel termination** command:

```
switch(config)# mpls tunnel termination model ttl uniform dscp pipe
```

Also, a new TCAM profile packet type has been introduced:

```
switch(config)# packet mpls ipv4 forwarding routed decap
```

Configuration occurs under the “qos ip” feature section of a user defined TCAM profile:

```
switch(config)# hardware tcam
# May use any profile that provides "qos ip"
switch(config-tcam)# profile name copy default
switch(config-tcam-profile-name)# feature qos ip
switch(config-tcam-profile-name-feature-qos-ip)# packet mpls ipv4 forwarding
 routed decap

switch(config-tcam)# system profile name
```

19.1.1.4.1.1Packet Support for IPv6

RFE 391109 brings mixed TTL/DSCP support for IPv6 over MPLS packets as well. The TCAM profile configuration is similar to the above IPv4 configuration:

```
switch(config)# hardware tcam
   # May use any profile that provides "qos ipv6"
switch(config-tcam)# profile name copy default
switch(config-tcam-profile-name)# feature qos ipv6
switch(config-tcam-profile-name-ipv4)# packet mpls ipv6 forwarding routed decap
switch(config-tcam)# system profile name
```

19.1.1.5 Support for MPLS Packets in IP ACLs

Ingress and egress IPv4 and IPv6 access-lists do not automatically match the inner IP header when processing IP over MPLS traffic.

Support is now avaiable to allow matching of the inner IP header in an access-list by adding the proper packet types to the corresponding TCAM features.

19.1.1.6 DSCP-to-TC Maps for MPLS Traffic to L3 VRFs

MPLS-labeled traffic, both IPv4 and IPv6, can be decapsulated and routed using a specific VRF, either the default or a named VRF. The MPLS label can be statically assigned to a specific VRF, or dynamically through L3 EVPN MPLS. In this configuration, a traffic class (TC) label based on a packet's ingress DSCP is added to the decapsulated MPLS packet if it is directed to a specific VRF. This is an MPLS tunneling technique called Short Pipe Mode.

The mapping of DSCP labels to TC is steered by a QoS map. Up to 10 custom maps can be applied. Such a map will only use hardware resources when attached to at least one VRF. When a map is first created, the DSCP-to-TC map is copied from the global map; subsequent changes to the global map do not have any effect on custom maps.

19.1.2 MPLS Configuration

MPLS routing is enabled through the **`mpls ip`** command.

This command enables MPLS routing.

```
switch(config)# mpls ip
switch(config)# show running-config mpls ip
!

end
```

```
switch(config)#
```

MPLS rules are created by the **mpls static** command. MPLS static rules identify a set of MPLS packets by a common top label and defines the method of handling these packets.

These commands create an MPLS rule that matches packets with a top label value of ***3400*** and causes the removal of the top label from the header stack. The nexthop destination of the IPv4 payload is IP address ***10.14.4.4*** through ***interface ethernet 3/3/3***. This rule has a metric value of ***100***.

```
switch(config)# mpls static top-label 3400 ethernet 3/3/3 10.14.4.4
 pop payload-type ipv4
switch(config)# show running-config

!
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type
 ipv4
!

end
switch(config)#
```

These commands create a backup rule that forwards the packet through ***interface ethernet 4/3***. This rule's metric value of ***150*** assigns it backup status prior to the first rule.

```
switch(config)# mpls static top-label 3400 ethernet 4/3 10.14.4.4 pop
 payload-type ipv4 metric 150
switch(config)# show running-config

!
mpls static top-label 3400 Ethernet4/3 10.14.4.4 pop payload-type ipv4
 metric 150
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type
 ipv4
!

end
switch(config)#
```

These commands create an MPLS rule that forwards the packet to the nexthop address through any interface.

```
switch(config)# mpls static top-label 4400 10.15.46.45 pop payload-
type ipv4
switch(config)# show running-config

!
mpls static top-label 3400 Ethernet4/3 10.14.4.4 pop payload-type ipv4
 metric 150
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type
 ipv4
mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
!
```

```
end
switch(config)#
```

This command configures a static tunnel for the tunnel endpoint ***64.0.0.1*** and pushes a label ***11111*** to it.

```
switch(config)# mpls static STATIC 64.0.0.1/32 54.0.0.1 Port-Channel7
 label-stack 11111
```

This example shows MPLS swap route configuration. The following CLI swaps a packet with MPLS top label ***3400*** to ***5600*** for forwarding to ***10.14.4.4***.

```
switch(config)# mpls static top-label 3400 10.14.4.4 swap 5600
switch(config)#
```

This example shows MPLS next-hop groups configuration for MPLS. The following CLI shows the specification of a next hop group called ***TestGrp1*** with two entries.

```
switch(config)# nexthop-group TestGrp1 type MPLS
switch(config-nexthop-group-TestGrp1)# size 2
switch(config-nexthop-group-TestGrp1)# entry 0 push label-stack 70
 nexthop 10.20.30.5
switch(config-nexthop-group-TestGrp1)# entry 1 push label-stack 71
 nexthop 10.20.30.6
switch(config-nexthop-group-TestGrp1)# exit
switch(config)#
```

The label-stack keyword can take only one label. Tunnel-source and ttl config commands do not apply to MPLS NexthopGroups and are disabled. The nexthop entry is recursively resolved when the specified entry is a remote nexthop. The show command indicates the resolved, directly-attached nexthop.

The switch's MPLS static rule configuration for specified routes and rules is displayed by **show mpls route**.

This command displays the MPLS rule configuration.

```
switch> show mpls config route
In-Label  Out-Label  Metric  Payload  NextHop
3400      pop        100     ipv4     10.14.4.4,Et3/3/3
3400      pop        150     ipv4     10.14.4.4,Et4/3 switch>
```

Statistics about the configuration and implementation of MPLS rules are displayed by the **show mpls route summary** command.

This command displays a summary of MPLS rule implementation.

```
switch> show mpls route summary
Number of Labels: 1 (1 unprogrammed)
Number of adjacencies in hardware: 0
Number of backup adjacencies: 2 switch>
```

The **show mpls lfib route** command displays the Label Forward Information Base of the switch.

```
switch(config)# show mpls lfib route
MPLS forwarding table (Label [metric] Vias) - 1 routes
MPLS next-hop resolution allow default route: False
Via Type Codes:
          M - MPLS via, P - Pseudowire via,
          I - IP lookup via, V - VLAN via,
          VA - EVPN VLAN aware via, ES - EVPN ethernet segment via,
          VF - EVPN VLAN flood via, AF - EVPN VLAN aware flood via,
          NG - Nexthop group via
Source Codes:
          G - gRIBI, S - Static MPLS route,
          B2 - BGP L2 EVPN, B3 - BGP L3 VPN,
          R - RSVP, LP - LDP pseudowire,
          L - LDP, M - MLDP,
          IP - IS-IS SR prefix segment, IA - IS-IS SR adjacency
 segment,
          IL - IS-IS SR segment to LDP, LI - LDP to IS-IS SR segment,
          BL - BGP LU, ST - SR TE policy,
          DE - Debug LFIB

 S   300      [100]
                via M, 192.0.2.2, swap 500
                 payload mpls, bypass egress-acl
                 interface Ethernet3/1
switch(config)#
```

The **show ip route** command displays all IP routes along with static MPLS push routes and Nexthop Group routes.

```
switch(config)# show ip route
VRF name: default
 Codes: C - connected, S - static, K - kernel,
 O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
 E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
 N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
 R - RIP, I L1 - ISIS level 1, I L2 - ISIS level 2,
 O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
 NG - Nexthop Group Static Route, V - VXLAN Control Service

Gateway of last resort is not set
C 192.0.2.0/24 is directly connected, Ethernet1/1
 S 3.3.3.0/24 [1/0] via 192.0.2.2, Ethernet1/1 label 200
 S 10.80.0.0/13 [1/0] via 172.24.0.1, Management1
 S 10.95.0.0/20 [1/0] via 172.24.0.1, Management1
 C 172.24.0.0/18 is directly connected, Management1
 S 172.16.0.0/12 [1/0] via 172.24.0.1, Management1
```

```
switch(config)#
```

The MPLS push routes are of type static so the **show ip route static** command also displays the output for an MPLS Nexthop Group assuming there is a route pointing to the NexthopGroup and the entries in the NexthopGroup are resolved. The output for **show nexthop-group TestGrp1** for the test group configured earlier is shown below.

```
switch(config)# show nexthop-group TestGrp1
TestGrp1
  Id 2
  Type MPLS
  Size 2
  Entries
   0 push label-stack 70 nexthop 10.20.30.5
   Tunnel destination directly connected, Ethernet2
   52:14:a3:a6:50:e8, Ethernet2
   1 push label-stack 71 nexthop 10.20.30.6
   Tunnel destination directly connected, Ethernet2
   52:14:a3:a6:50:e8, Ethernet2
switch(config)#
```

19.1.2.1 LSP Ping/Traceroute for MPLS Nexthop Group Tunnels

The **ping/traceroute mpls tunnel nexthop-group** command checks the liveness of Nexthop Group tunnel endpoint for an MPLS Nexthop Group.

The following displays the output for the **ping** command where the endpoint is specified.

```
switch(config)# rtrmpls1
switch(config-rtrmpls1)# ping mpls tunnel nexthop-group 100.0.116.1/32
LSP ping to nexthop-group tunnel 100.0.116.1/32

100.0.116.1/32: nexthop-group tunnel index 1 (nexthop-group name: nhg-100)
Entry 0
 Via 10.0.16.2
 Reply from 10.0.108.1: seq=1, time=507.546ms
Entry 1
 Via 10.0.16.8
 Reply from 10.0.113.1: seq=1, time=516.131ms

--- nexthop-group tunnel index 1, nexthop-group nhg-100: lspping statistics
        ---
Entry 0
 Via 10.0.16.2
 1 packets transmitted, 1 received, 0% packet loss, time 652ms
 1 received from 10.0.108.1, rtt min/max/avg 507.546/507.546/507.546 ms

Entry 1
 Via 10.0.16.8
 1 packets transmitted, 1 received, 0% packet loss, time 652ms
   1 received from 10.0.113.1, rtt min/max/avg 516.131/516.131/516.131ms
```

The following displays the output where the endpoint is specified for **traceroute** command.

```
switch(config)# rtrmpls1
switch(config-rtrmpls1)# traceroute mpls tunnel nexthop-group 100.0.116.1/32
LSP traceroute to nexthop-group tunnel 100.0.116.1/32
```

```
Traceroute over nexthop-group tunnel index 1, nexthop-group nhg-100 Entry 1
Entry 1
  1  10.0.225.1        32.571ms
     label stack (top label first): 89
  2  10.0.227.1        42.866ms
     label stack (top label first): 109
  3  10.0.229.1        54.893ms
     label stack (top label first): 110
  4  10.0.231.1        15.946ms
     label stack (top label first): 111
  5  10.0.233.1        27.72ms
     label stack (top label first): 112
  6  10.0.113.1        36.383ms
```

The following displays the output for the **ping** command where the static route resolves over Nexthop Group tunnel(s).

```
switch(config-rtrmpls1)# ping mpls static ip 100.0.77.0/24
LSP ping to static MPLS push label route 100.0.77.0/24

100.0.77.0/24: nexthop-group tunnel index 1 (nexthop-group name: nhg-46)
Entry 0
   Via 10.0.25.5
   Reply from 10.0.51.1: seq=1, time=112.171ms

100.0.77.0/24: nexthop-group tunnel index 2 (nexthop-group name: nhg-64)
Entry 0
   Via 10.0.25.4
   Reply from 10.0.68.1: seq=1, time=121.809ms
Entry 1
   Via 10.0.25.9
   Reply from 10.0.76.1: seq=1, time=133.668ms

--- nexthop-group tunnel index 1, nexthop-group nhg-46: lspping statistics ---
Entry 0
   Via 10.0.25.5
   1 packets transmitted, 1 received, 0% packet loss, time 251ms
   1 received from 10.0.51.1, rtt min/max/avg 112.171/112.171/112.171 ms


--- nexthop-group tunnel index 2, nexthop-group nhg-64: lspping statistics ---
Entry 0
   Via 10.0.25.4
   1 packets transmitted, 1 received, 0% packet loss, time 251ms
   1 received from 10.0.68.1, rtt min/max/avg 121.809/121.809/121.809 ms

Entry 1
   Via 10.0.25.9
   1 packets transmitted, 1 received, 0% packet loss, time 251ms
   1 received from 10.0.76.1, rtt min/max/avg 133.668/133.668/133.668 ms
```

The following displays the output for the **traceroute** command where the static route resolves over Nexthop Group tunnel(s). The command randomly selects a tunnel.

```
switch(config-rtrmpls1)# traceroute mpls static ip 100.0.77.0/24LSP traceroute
 to 100.0.77.0/24
100.0.77.0/24: nexthop-group tunnel index 1 (nexthop-group name: nhg-46)
100.0.77.0/24: nexthop-group tunnel index 2 (nexthop-group name: nhg-64)
Traceroute over nexthop-group tunnel index 1, nexthop-group nhg-46 Entry 0
Entry 0
  1  10.0.83.1         382.798ms
     label stack (top label first): 35
```

```
  2  10.0.85.1         42.7ms
     label stack (top label first): 47
  3  10.0.87.1         55.815ms
     label stack (top label first): 48
  4  10.0.89.1         17.728ms
     label stack (top label first): 49
  5  10.0.91.1         29.452ms
     label stack (top label first): 50
  6  10.0.51.1         38.686ms
```

19.1.2.2 Egress IPv4/IPv6 over MPLS ACLs

IPv4/IPv6 over MPLS packets are now eligible for ACLs at the egress stage by default, applicable only to IPv4/IPv6 over MPLS packets that are MPLS label popped (such as if the label is at the bottom of stack). The user can override this behavior if required, thereby disabling egress ACLs for certain MPLS labels by configuration. No special configuration is required to enable egress ACLs on IPv4/IPv6 over MPLS packets.

Examples

- This command disables egress ACLs for MPLS top-label ***12000*** on the egress interface ***120.1.1.1*** nexthop address.

```
switch(config)# no mpls static top-label 12000 120.1.1.1 pop
 payload-type ipv6
switch(config)#
```

- This command enables egress ACLs for MPLS top-label ***12000*** on the egress interface ***120.1.1.1*** nexthop address.

```
switch(config)# mpls static top-label 12000 120.1.1.1 pop payload-
type ipv6
switch(config)#
```

19.1.2.3 Configuring MPLSoGRE Filtered Mirroring

The filtered mirroring of terminated MPLSoGRE packets is configured by creating an IPv4 access-list, and then attaching the IPv4 access-list to a monitor session source where a tunnel decap group has been configured. This IPv4 access-list has rules that match to either inner IPv4 or IPv6 destination addresses.

Enabling the TC-Counters TCAM Profile

The following limitations are applicable to MPLSoGRE filtered mirroring in tc-counters TCAM profile:

- Security ACLs are not enforced on IPv4oMPLSoGRE and IPv6oMPLSoGRE terminated packets.
- The rules of a mirroring-ACL are set to match either inner IPv4 or inner IPv6 header fields, but not both.

The ACLs containing rules to match both inner IPv4 and inner IPv6 header fields are not applicable to a single source interface in multiple mirroring sessions. In other words, all ACLs applied to a shared source interface must contain either inner IPv4 rules or inner IPv6 rules.

The commands below switch to the tc-counters TCAM profile in the running configuration.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# system profile tc-counters
switch(config-hw-tcam)# exit
```

Defining Two IPv4 Access-Lists

The **ip access-list** command places the switch in ACL configuration mode, which is a group change mode that modifies an IPv4 access control list. The command specifies the name of the IPv4 ACL that

subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending till the end of the session.

The **permit (Role)** command configures one access-list to match the inner IPv4 address, and the other access-list to match the inner IPv6 address.

```
switch(config)# ip access-list dIPv4
switch(config)# 10 permit ip any any inner ip any host 5.5.5.5
switch(config)# exit

switch(config)# ip access-list dIPv6
switch(config)# 10 permit ip any any inner ipv6 any host 55::55
switch(config)# exit
```

Attaching Access-Lists

The **monitor session source** and **monitor session destination** commands allow to attach two access-lists to two different monitor session sources.

```
switch(config)# monitor session sess1 source et1 rx ip access-group dIPv4
switch(config)# monitor session sess1 destination tunnel mode gre source 1.1.1.1 destination
 2.2.2.2
switch(config)# monitor session sess2 source et2 rx ip access-group dIPv6
switch(config)# monitor session sess2 destination tunnel mode gre source 3.3.3.3 destination
 4.4.4.4
switch(config)# show monitor session

Session sess1
-------------------------

Source Ports:

  Rx Only:     Et1(IP ACL: dIPv4)

Destination Ports:

        status    source    dest     TTL   DSCP  proto    VRF      fwd-drop
Gre1 :  active    1.1.1.1   2.2.2.2  128   0     0x88be   default  no


Session sess2
-------------------------

Source Ports:

  Rx Only:     Et2(IP ACL: dIPv6), Et5(IP ACL: dIPv6)

Destination Ports:

        status   source    dest     TTL   DSCP  proto    VRF        fwd-drop
Gre2 :  active   3.3.3.3   4.4.4.4  128   0     0x88be   default    no

switch(config)#
```

19.1.2.4 Configurations to Support IP ACLs for MPLS Packets

19.1.2.4.1 Ingress IP Access-List Support

To facilitate inner IP header matching, there are two TCAM profile packet types:

```
packet mpls ipv4 forwarding bridged
packet mpls ipv4 forwarding mpls
```

The ingress inner IP header matching creates a user-defined profile with these new packet types. Although MAC access-lists do not match on the inner IP header, you must add this packet type in order to generate a lookup for MPLS packets (if using MAC ACLs).

```
hardware tcam
   #May use any profile that provides ACL features
   profile <name> copy default
      feature acl port ip
         packet mpls ipv4 forwarding bridged
         packet mpls ipv4 forwarding mpls
      feature acl port ipv6
         packet mpls ipv6 forwarding bridged
         packet mpls ipv6 forwarding mpls
      feature acl port mac
         packet mpls ipv4 forwarding bridged
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding bridged
         packet mpls ipv6 forwarding mpls
   system profile <name>
```

Starting in Release ***EOS 4.23.1***, ingress on inner IP header matching in MPLS packets is also supported on PBR ACLs. In order to enable this, the following must be configured in a user-defined TCAM profile:

```
hardware tcam
   #May use any profile that provides ACL features
   profile <name> copy default
      feature pbr ip
         packet mpls ipv4 forwarding mpls
      feature pbr ipv6
         packet mpls ipv6 forwarding mpls
```

19.1.2.4.2 Egress IPv4/IPv6 over MPLS ACLs

IPv4/IPv6 over MPLS packets are now eligible for ACLs at the egress stage by default, applicable only to IPv4/IPv6 over MPLS packets that are MPLS label popped (such as if the label is at the bottom of stack). The user can override this behavior if required, thereby disabling egress ACLs for certain MPLS labels by configuration. No special configuration is required to enable egress ACLs on IPv4/IPv6 over MPLS packets.

Examples

- This command disables egress ACLs for MPLS top-label ***12000*** on the egress interface ***120.1.1.1*** nexthop address.

  ```
  switch(config)# no mpls static top-label 12000 120.1.1.1 pop
   payload-type ipv6
  switch(config)#
  ```

- This command enables egress ACLs for MPLS top-label ***12000*** on the egress interface ***120.1.1.1*** nexthop address.

  ```
  switch(config)# mpls static top-label 12000 120.1.1.1 pop payload-
  type ipv6
  switch(config)#
  ```

19.1.2.4.3 MPLS Pop Terminated Packets

To enable the use of these qualifiers on MPLS-terminated packets by last label pop, this feature must be applied to the current TCAM profile, as illustrated below:

```
feature acl port ip egress mpls-tunnelled-match
```

No functionality is lost, however, this feature is not enabled by default on all system profiles. Some TCAM profiles may also use the resources required by this feature. Verification is required to ensure the current user-defined TCAM profile supports this feature.

```
hardware tcam
   #May use any profile that provides ACL features
   profile <name> copy <some-system-profile>
      feature acl port ip egress mpls-tunnelled-match
   system profile <name>
```

19.1.2.4.4 MPLS VRF-Decapsulated Packets

To enable the use of these new qualifiers on MPLS-terminated packets by last label VRF-label mapping, the packet type **`mpls ipv4 forwarding routed decap`** must be applied to at least one feature in the TCAM profile. For example:

```
hardware tcam
   profile <name> copy default
      feature qos ip
         packet mpls ipv4 forwarding routed decap
   system profile <name>
```

19.1.2.5 Supporting Traffic Policy on Interfaces

Access Control Lists (ACL) configures the action for packets which are going through the packet processor pipeline based on different fields of packets. Usually TCAM is used to match packets with multiple entries matching the list of IP addresses. TCAM is also a limited resource, so traffic-policy performs transformation of the fields of the packet, that summarize them in terms of the relevant rules this field matches by using command **`interface traffic-policy`**.

Custom TCAM Profile

The following commands enable custom TCAM profile to support traffic policy.

```
hardware tcam
   profile traffic-policy
      feature acl port mac
         sequence 55
         key size limit 160
         key field dst-mac ether-type src-mac
         action count drop
         packet ipv4 forwarding bridged
         packet ipv4 forwarding routed
         packet ipv4 forwarding routed multicast
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet ipv4 non-vxlan forwarding routed decap
         packet ipv4 vxlan forwarding bridged decap
         packet ipv6 forwarding bridged
         packet ipv6 forwarding routed
         packet ipv6 forwarding routed decap
         packet ipv6 forwarding routed multicast
```

```
         packet ipv6 ipv6 forwarding routed decap
         packet mpls forwarding bridged decap
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding mpls
         packet mpls non-ip forwarding mpls
         packet non-ip forwarding bridged
      feature forwarding-destination mpls
         sequence 100
      feature mirror ip
         sequence 80
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops l4-src-
port src-ip tcp-control
         action count mirror set-policer
         packet ipv4 forwarding bridged
         packet ipv4 forwarding routed
         packet ipv4 forwarding routed multicast
         packet ipv4 non-vxlan forwarding routed decap
      feature mpls
         sequence 5
         key size limit 160
         action drop redirect set-ecn
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding mpls
         packet mpls non-ip forwarding mpls
      feature pbr ip
         sequence 60
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops-18b l4-
src-port src-ip tcp-control
         action count redirect
         packet ipv4 forwarding routed
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet ipv4 non-vxlan forwarding routed decap
         packet ipv4 vxlan forwarding bridged decap
      feature pbr ipv6
         sequence 30
         key field dst-ipv6 ipv6-next-header l4-dst-port l4-src-port src-ipv6-
high src-ipv6-low tcp-control
         action count redirect
         packet ipv6 forwarding routed
      feature pbr mpls
         sequence 65
         key size limit 160
         key field mpls-inner-ip-tos
         action count drop redirect
         packet mpls ipv4 forwarding mpls
         packet mpls ipv6 forwarding mpls
         packet mpls non-ip forwarding mpls
      feature qos ip
         sequence 75
         key size limit 160
         key field dscp dst-ip ip-frag ip-protocol l4-dst-port l4-ops l4-src-
port src-ip tckp-control
         action set-dscp set-policer set-tc
         packet ipv4 forwarding routed
         packet ipv4 forwarding routed multicast
         packet ipv4 mpls ipv4 forwarding mpls decap
         packet ipv4 mpls ipv6 forwarding mpls decap
         packet ipv4 non-vxlan forwarding routed decap
      feature qos ipv6
```

```
         sequence 70
         key field dst-ipv6 ipv6-next-header ipv6-traffic-class l4-dst-port l4-
src-port src-ipv6-high src-ipv6-low
         action set-dscp set-policer set-tc
         packet ipv6 forwarding routed
      feature traffic-policy port ipv4
         sequence 45
         key size limit 160
         key field dscp dst-ip-label icmp-type-code ip-frag ip-fragment-offset
 ip-length ip-protocol l4-dst-port
         l4-src-port src-ip-label tcp-control ttl
         action count drop log set-dscp set-tc
         packet ipv4 forwarding routed
      feature traffic-policy port ipv6
         sequence 25
         key field dst-ipv6-label hop-limit icmp-type-code ipv6-length ipv6-
next-header ipv6-traffic-class l4-dst-port
         l4-src-port src-ipv6-label tcp-control
         action count drop log set-dscp set-tc
         packet ipv6 forwarding routed
      feature tunnel vxlan
         sequence 50
         key size limit 160
         packet ipv4 vxlan eth ipv4 forwarding routed decap
         packet ipv4 vxlan forwarding bridged decap
```

19.2 BGP/MPLS L3 VPN

Border Gateway Protocol/ Multiprotocol Label Switching (BGP/MPLS) L3 Virtual rivate Network (VPN) allows a Service Provider (SP) or an Enterprise to provide the service of interconnecting geographically dispersed customer sites. This type of service can be provided to multiple customers over the common network backbone infrastructure of the Service Provider, while:

- Maintaining privacy of each customer.
- Allowing for overlapping IP addresses among customers.
- Having constrained route distribution – such as, only those routers in the Service Provider's network that need the customer routes, have them.

Achieve the above through the extensions to BGP as defined in ***RFC 4364*** for IPv4 and ***RFC 4659*** for IPv6, and the use of VPN Routing and Forwarding Tables (VRFs), Route Distinguishers (RDs), and Route Targets (RTs).

BGP/MPLS L3 VPN is available when configuring BGP in the multi-agent routing protocol model.

- Operation
- Configuration
- Show Commands
- Limitations

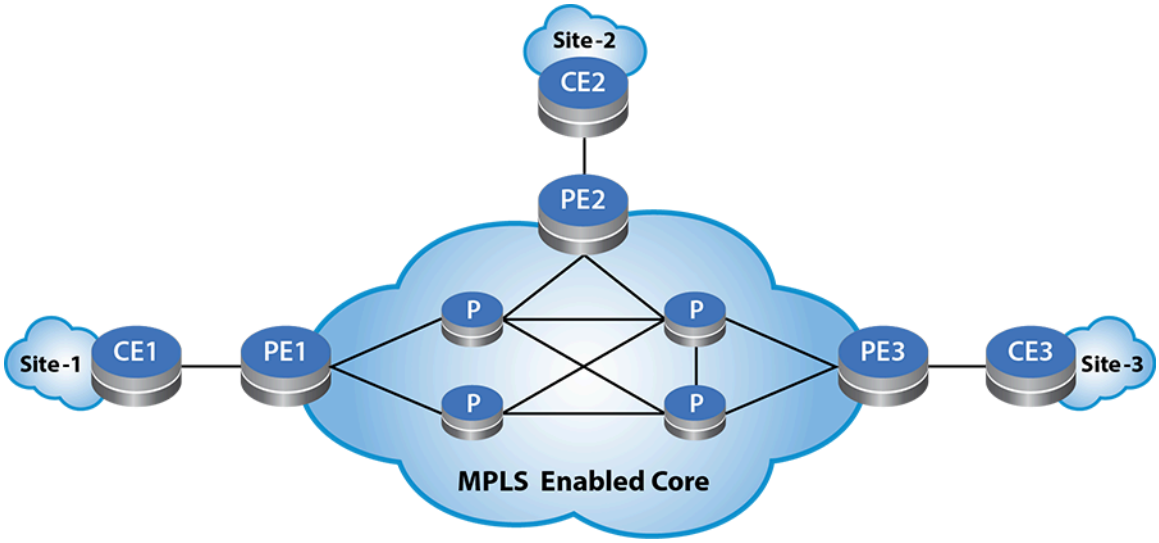
19.2.1 Operation

A Virtual Private Network, or VPN, is a set of geographically dispersed sites attached to the Service Provider's (SP) backbone, with IP interconnectivity amongst the sites. Using this scenario, the customer can obtain "VPN service" from the SP. The SP will provide a VPN service to multiple customers, using this common backbone network infrastructure. The sample VPN topology diagram below illustrates three sites where a customer is being interconnected over the SP backbone network.

At each site, the Customer Edge router (CE) attaches to the Provider's Edge router (PE). The CE can attach to more than one PE, and in these cases the CE is said to be "multi-homed". The routers in the SP core

network, those which do not attach to any CE, are referred to as “P” routers. The “P” routers do not need to know about the customer routes.

Figure 19-2: MPLS Enabled Core Schematic



The CE attaches to PE in a VRF. The routes learned from the CE are programmed in the corresponding VRF and the PE then distributes the routes to other PEs as “VPN routes” using MP-BGP. This is done by using two new BGP address families, VPN-IPv4 (AFI=1, SAFI=128) and VPN-IPv6 (AFI=2, SAFI=128).

On the PEs, corresponding to the VPN, VRFs configure with the RD and one or more import and export RTs. The RD attaches to the customer route to create a VPN route. By picking a unique RD for each VRF and attaching it to the customer route, the VPN route is unique. This allows for customers with overlapping IP addresses to be managed by the Service Provider. This unique VPN route is advertised to other PEs along with the configured export RT and the VPN label. The PE router allocates a VPN label per VRF and address family. The PE router programs its Label FIB (LFIB) with this label information. When the PE router receives an incoming MPLS packet with the VPN label as the topmost label, it pops the label and does an IP lookup in the associated VRF. The PE router also maintains a mapping of import RTs to the corresponding VRFs. This mapping is used later when deciding into which VRFs a received VPN route should be imported into.

The PE learns customer routes from the CE through the PE-CE routing protocol, which could be Static routing, EBGP, OSPF, or ISIS. The PE will install those customer routes in the associated VRF, with the CE as the nexthop. The customer routes in the VRF are exported into the BGP VPN table as VPN routes, along with the VPN label and the configured export RTs as BGP path attributes. These VPN routes are then advertised to other PEs which have been activated for the VPN address-families (VPN-IPv4/VPN-IPv6). The PE then sets itself as the nexthop while advertising the VPN routes. The PE which receives those VPN routes strips out the RD and import them as IPv4 or IPv6 routes into the VRFs. The list of VRFs into which the route is imported is determined based on the mapping of import RTs to VRFs. These routes will be programmed into the FIB along with the VPN label and the remote PE as the nexthop. Once these routes are installed in the VRF, the CE connected to that VRF will learn those routes (based on the PE-CE routing protocol) and make it available it in the customer network that it is attached to. This way the geographically dispersed customer sites learn of each other’s networks and IP reachability is established between the them.

19.2.1.1 Forwarding

In the Service Provider’s core network there should be MPLS LSPs between the PEs. That is, the connection to the VPN next-hop should be over an MPLS tunnel. The MPLS LSPs in the core could be setup using RSVP-TE or LDP in conjunction with OSPF/ISIS or ISIS-SR.

When the PE receives an IP packet from the CE destined to the remote site, it does an IP lookup in the VRF to which the CE is connected. This lookup provides the VPN label to be used and the nexthop, which would be the remote-PE. The VPN label is imposed on the IP packet and the resulting MPLS packet is

then tunneled through the MPLS LSP to the remote PE. As result of penultimate hop popping, the MPLS packet arrives at the remote PE with the VPN label as the topmost label. The label lookup results in the VPN label being popped and an IP lookup will be done in the associated VRF. Note that the PE would have programmed this label action when it allocated the label to start with. That IP lookup will result in the IP packet being forwarded out to the CE.

19.2.2 Configuration

Configuring BGP/MPLS L3 VPN involves enabling the SP core for MPLS and then configuring the PEs with the required BGP configuration.

It is assumed that the SP core network is enabled for MPLS. This involves configuring an IGP (OSPF/ISIS) followed by a label distribution protocol such as LDP, RSVP-TE or ISIS-SR. Typically, loopback interfaces are configured on all the PE and the P routers and the IGPs exchange reachability to those loopback interfaces. And then the MPLS Label Distribution Protocol will set up MPLS LSPs/tunnels between all those loopbacks.

This section includes the following topics:

- LDP Hello Redundancy
- Enabling BGP to Exchange the Routing Tables with the Peer
- Configuring the VRF Information
- Configuring the Route Distinguisher
- Configuring Route Targets
- Configuring Import / Export Route-maps
- VPN Next-hop

Enabling MPLS and LDP on the PE involves the following steps:

1. Configure terminal.

```
switch# configure terminal
```

2. Enter the ***Loopback0*** interface config mode.

```
switch(config)# interface Loopback0
```

3. Enter destination IP address.

```
switch(config)# ip address 11.0.0.1/32
```

4. Enable MPLS routing.

```
switch(config)# mpls ip
```

5. Configure MPLS LDP.

```
switch(config)# mpls ldp
```

6. Configure the router ID interface.

```
switch(config-mpls-ldp)# router-id interface Loopback0
```

7. Configure for no shutdown

```
switch(config-mpls-ldp)# no shutdown
```

Example:

```
switch# configure terminal
switch(config)# interface Loopback0
switch(config-if)# ip address 11.0.0.1/32
```

```
switch(config)# mpls ip
switch(config)# mpls ldp
switch(config-mpls-ldp)# router-id interface Loopback0
switch(config-mpls-ldp)# no shutdown
```

19.2.2.1 LDP Hello Redundancy

LDP Hello Redundancy establishes a redundant target Hello adjacency for each neighbor discovered through the Basic Discovery Mechanism using the LDP Extended Discovery mechanism. For the mechanism to work, the following must be true:

- Devices must have a loopback interface configured to serve as the transport address interface, which must be routable through all interfaces on the device.
- Devices must be reachable via a redundant path through other devices on the network.
- Both devices must have Hello Redundancy configured or the targeted Hello messages will be ignored.

After a Hello adjacency is established using LDP Basic Discovery, devices with Hello Redundancy will start sending Targeted Hello messages to the Transport Address found in the received Link Hello message of Basic Discovery. The Targeted Hello adjacency can support the session established between peers even when all Link Hello adjacencies have timed out. The FEC label bindings between two peers with no Link Hello adjacency will not be active because the Interior Gateway Protocol will not use the other peer as the next hop. Maintaining the FEC label bindings and the session between the two peers can save significant time when the Link Hello adjacency is reestablished.

The **`neighbor hello-redundancy`** command configures Hello Redundancy on all platforms under the LDP configuration mode. If a Link Hello adjacency is restored within 600 seconds of being lost, the Target Hello adjacency and the session associated with it will be dropped. The timeout can be configured using the ***duration*** option of the command. An infinite value for the ***duration*** disables the timeout.

Configuration for LDP Hello Redundancy

The following example is applicable to all platforms.

These commands enable Targeted Hello redundancy with a duration of 300 seconds.

```
switch(config-mpls-ldp)# neighbor hello-redundancy
switch(config-mpls-ldp)# neighbor hello-redundancy duration 300
```

Either of the following commands in the example below disable Hello Redundancy.

```
switch(config-mpls-ldp)# neighbor hello-redundancy none
switch(config-mpls-ldp)# default neighbor hello-redundancy
```

The following command shows the Targeted Hello adjacencies established.

```
switch(config)# show mpls ldp discovery detail
LDP MD5 Password Not Set
Local LDP Identifier: 2.2.2.2:0
Discovery Sources:
   Interfaces:
      Ethernet1 (ldp):
         Hello interval: 5 sec; Source IP addr: 192.168.2.1
         LDP ID: 3.3.3.3:0
            Source IP addr: 192.168.2.2; Transport IP addr: 3.3.3.3
            Hold time: 15 sec; Proposed local/peer: 15/15 sec; Expires in:
 11.96 sec
      Ethernet2 (ldp):
         Hello interval: 5 sec; Source IP addr: 192.168.1.2
         LDP ID: 1.1.1.1:0
            Source IP addr: 192.168.1.1; Transport IP addr: 1.1.1.1
```

```
            Hold time: 15 sec; Proposed local/peer: 15/15 sec; Expires in:
12.00 sec
  Targeted Hellos:
     Targeted neighbor 1.1.1.1:
        Hello interval: 15 sec; Source IP addr: 2.2.2.2
        LDP ID: 1.1.1.1:0
           Source IP addr: 1.1.1.1; Transport IP addr: 1.1.1.1
           Hold time: 45 sec; Proposed local/peer: 45/45 sec; Expires in:
40.78 sec
           Target configuration source: Hello Redundancy
     Targeted neighbor 3.3.3.3:
        Hello interval: 15 sec; Source IP addr: 2.2.2.2
        LDP ID: 3.3.3.3:0
           Source IP addr: 3.3.3.3; Transport IP addr: 3.3.3.3
           Hold time: 45 sec; Proposed local/peer: 45/45 sec; Expires in:
35.89 sec
           Target configuration source: Hello Redundancy
```

19.2.2.2 Enabling BGP to Exchange the Routing Tables with the Peer

Enabling the send-community extended knob on the neighbor. This essentially enables BGP to exchange the RTs with the peer.

- Activating the peer (the remote PE) under the **`address-family VPN-IPv4`** and **`address-family VPN-IPv6`** modes enables BGP to negotiate the MPLS L3 VPN address families.
- Specify the address for the VPN next-hop using the command, **`neighbor default encapsulation mpls next-hop-self source-interface Loopback0.`**

In the preceding configuration example, the system uses the address ***11.0.0.1*** from interface ***Loopback0*** as the nexthop in the VPN route advertisements.

19.2.2.3 Configuring the VRF Information

First, the VRF must be configured in the ***global*** mode and IPv4 and IPv6 routing must be enabled in the VRF. After that, under the ***router bgp*** mode, we need to configure the VRF and provide the information related to RD and import and export RTs.

1. Configure terminal.

```
switch# configure terminal
```

2. Enable IP routing.

```
switch(config)# ip routing
```

3. Enable IP routing for the VRF.

```
switch(config)# ip routing vrf vrf1
```

4. Enable IPv6 routing for the VRF.

```
switch(config)# ipv6 unicast-routing vrf vrf1
```

```
switch# configure terminal
switch(config)# ip routing
switch(config)# ip routing vrf vrf1
switch(config)# ipv6 unicast-routing vrf vrf1
```

The VRF configuration under router BGP mode is shown below.

```
switch(config-router-bgp-vrf-vrf1)# show active
router bgp 300
  vrf vrf1
    rd 11.0.0.1:0
    route-target import vpn-ipv6 300:0
    route-target import vpn-ipv4 300:0
    route-target import vpn-ipv4 300:0
    route-target import vpn-ipv4 300:0
    route-target export vpn-ipv6 300:0
    route-target export vpn-ipv4 300:0
    redistribute connected
    redistribute static
switch(config-router-bgp-vrf-vrf1)#
```

19.2.2.4 Configuring the Route Distinguisher

The Route Distinguisher (RD) is structured such that it can be easily configured and managed. It is configured by specifying two fields separated by a colon, as in administrative-subfield: assigned-number-subfield.

The administrative subfield could contain either an IP address, as shown in the example (the PEs loopback address), or the AS number. The assigned-number-subfield can be any number which is determined by the SP. The primary requirement of the RD is that it must be unique per VRF. It is possible to have overlapping address space between VRFs, the intent of the RD is to ensure that a VPN route can be uniquely identified as it is received by a remote PE. However, the identification of which VRF(s) the received VPN route should be imported into is handled by the import RTs configured on the remote PE.

19.2.2.5 Configuring Route Targets

Next is the import and export Route Target configuration. The RTs are structured similar to the RDs. They too are made up of administrative-subfield: **assigned-number-subfield**. In the example shown, the AS number has been used as the administrative subfield. And the value ***0*** has been used for the assigned number subfield. It is the RT that plays an important role in identifying the VPN.

- Received VPN routes with the import RT as path attributes will be imported into the VRF will have RTs has extended-community BGP path attributes. They will be imported into VRFs which import RT configuration for RTs in the received VPN route.
- And while advertising the routes from the VRF as VPN routes, the export RT will be attached to the route as an extended-community BGP attribute.

1. In the example, the connected and static routes in the VRF are redistributed into BGP. And these routes will be exported as VPN routes.
2. It is also possible to have a BGP session with the CE. In that case, routes received over that session are exported as VPN routes. And imported routes are advertised to the CE.

19.2.2.6 Configuring Import / Export Route-maps

The routes that are imported into a VRF can be further controlled by applying an import route-map. And the routes that are exported from the VRF can be controlled by applying an export route-map.

Complete the following steps:

1. Configure terminal.

```
switch# configure terminal
```

2. Configure BGP. In this example, BGP is configured AS ***4274781899.***

```
switch(config)# router bgp 4274781899
```

3. Configure the VRF under router bgp mode.

```
switch(config-router-bgp)# vrf vrf1
```

4. Select the BGP route distinguisher. In this example, ***36351:268450419*** is selected.

```
switch(config-router-bgp-vrf-vrf1)# rd 36351:268450419
```

5. Configure the import route-target VPN-IPv4, unicast address family. In this example, ***36351:1001*** is selected.

```
switch(config-router-bgp-vrf-vrf1)# route-target import vpn-ipv4 36351:1001
```

6. Configure another import route-target for VPN-IPv4 unicast address family. In this example, ***36351:268450419*** is selected.

```
switch(config-router-bgp-vrf-vrf1)# route-target import vpn-ipv4
 36351:268450419
```

7. Configure the export route-target for VPN-IPv4 unicast address family. In this example, ***36351:268450419*** is selected.

```
switch(config-router-bgp-vrf-vrf1)# route-target export vpn-ipv4
 36351:268450419
```

8. Configure the import route-map. In this example, the import router-map name is **BGP-IMPORT-VRF-SERVICES**.

```
switch(config-router-bgp-vrf-vrf1)# route-target import vpn-ipv4 route-map
 BGP-IMPORT-VRF-SERVICES
```

9. Configure the export route-map. In this example, the export route-map name is **BGP-EXPORT-VRF-VRF1**.

```
switch(config-router-bgp-vrf-vrf1)# route-target export vpn-ipv4 route-map
 BGP-EXPORT-VRF-VRF1
```

10. Optionally, redistribute connected routes in the VRF into BGP, with the associated route-map **BGP-ANNOUNCE-CONNECTED**.

```
switch(config-router-bgp-vrf-vrf1)# redistribute connected route-map BGP-
ANNOUNCE-CONNECTED
```

11. Optionally, redistribute static routes in the vrf into BGP, with associated route-map **BGP-ANNOUNCE-STATIC**.

```
switch(config-router-bgp-vrf-vrf1)# redistribute static route-map BGP-
ANNOUNCE-STATIC
```

12. The import and export route-maps should be separately configured. The configuration snippet below shows the configuration of the same route-map **BGP-IMPORT-VRF-SERVICES**.

```
switch(config-router-bgp-vrf-vrf1)# route-map BGP-IMPORT-VRF-SERVICES permit
 10
switch(config-route-map-BGP-IMPORT-VRF-SERVICES)# match extcommunity
 SERVICES
```

```
switch(config-route-map-BGP-IMPORT-VRF-SERVICES)# match ip address prefix-
list SERVICES
```

Example:

```
switch# configure terminal
switch(config)# router bgp 4274781899
switch(config-router-bgp)# vrf vrf1
switch(config-router-bgp-vrf-vrf1)# rd 36351:268450419
switch(config-router-bgp-vrf-vrf1)# route-target import vpn-ipv4
 36351:1001
switch(config-router-bgp-vrf-vrf1)# route-target import vpn-ipv4
 36351:268450419
switch(config-router-bgp-vrf-vrf1)# route-target export vpn-ipv4
 36351:268450419
switch(config-router-bgp-vrf-vrf1)# route-target import vpn-ipv4
 route-map BGP-IMPORT-VRF-SERVICES
switch(config-router-bgp-vrf-vrf1)# route-target export vpn-ipv4
 route-map BGP-EXPORT-VRF-VRF1
switch(config-router-bgp-vrf-vrf1)# redistribute connected route-map
 BGP-ANNOUNCE-CONNECTED
switch(config-router-bgp-vrf-vrf1)# redistribute static route-map BGP-
ANNOUNCE-STATIC
switch(config-router-bgp-vrf-vrf1)# route-map BGP-IMPORT-VRF-SERVICES
 permit 10
switch(config-route-map-BGP-IMPORT-VRF-SERVICES)# match extcommunity
 SERVICES
switch(config-route-map-BGP-IMPORT-VRF-SERVICES)# match ip address
 prefix-list SERVICES
```

19.2.2.7 VPN Next-hop

By default, the system uses the source address of the BGP session as the nexthop in the VPN advertisements. This source address is determined by the system while establishing the TCP session between the PEs. If the SP want to use a different address as the VPN next-hop, then an interface with that address must be specified under the BGP address-family VPN-IPv4 or address-family VPN-IPv6 configuration modes. For example,

```
switch(config-router-bgp)# address-family vpn-ipv4
switch(config-router-bgp-af)# neighbor 10.0.0.2 activate
switch(config-router-bgp-af)# neighbor default encapsulation mpls next-hop-self
 source-interface Loopback0


switch(config-router-bgp)# address-family vpn-ipv6
switch(config-router-bgp-af)# neighbor 10.0.0.2 activate
switch(config-router-bgp-af)# neighbor default encapsulation mpls next-hop-self
 source-interface Loopback0
```

In the previous example, the interface ***Loopback 0*** has been specified. The system picks the VPN next-hop address from the interface based on the following rules:

- For VPN-IPv4, the IPv4 address from the interface is picked.
- For VPN-IPv6,
  - For an IPv4 peer,
    - Pick the IPv4 address from the interface.

    - If the interface does not have a IPv4 address, pick the IPv6 address.
- For an IPv6 peer,
    - Pick the IPv6 address from the interface.
    - If the interface does not have an IPv6 address, pick the IPv4 address.

For VPN-IPv6, when an IPv4 address is picked as the next-hop, it is encoded as a IPv4-mapped IPv6 address in the VPN route advertisement. This is as per ***RFC 4659***.

19.2.3 Show Commands

Use the **`show bgp instance vrf vrf1`** command to show the BGP instance status for a specific VRF to verify the route-targets and import/export route-maps being used. The command also displays the locally allocated MPLS label that the system has allocated for IPv4 and IPv6.

```
switch# show bgp instance vrf vrf1
BGP instance information for VRF vrf1
BGP Local AS: 4274781899, Router ID: 169.254.156.10
Total peers:                        0
 Static peers:          0
 Dynamic peers:         0
 Disabled peers:        0
 Established peers:     0
Four Octet ASN mode enabled
Graceful restart helper mode disabled
Graceful restart mode disabled
Graceful restart timer timeout: 00:05:00
End of rib timer timeout: 00:05:00
Attributes of the reflected routes are not preserved
UCMP mode: disabled
Peer mac resolution timeout: 00:00:00
BGP IPv4 Listen Port Status: listening on port 179
BGP IPv6 Listen Port Status: listening on port 179
BGP Convergence information:
   BGP has converged:   yes,   Time taken to converge: 00:00:31
   Outstanding EORs:    0,     Outstanding Keepalives: 0
   Convergence timeout: 00:05:00
BGP Convergence timer is inactive
BGP Convergence based update synchronization is disabled
BGP Convergence slow-peer timeout: 00:01:30
Address-family IPv4 Unicast:
 Redistributed routes into BGP:
   Static
   Connected
 Route Distinguisher: 36351:268450419
  Route targets to import:
   VPN-IPv4:
     36351:1001
     36351:268450419
 Route targets to export:
   VPN-IPv4:
     36351:268450419
 Route maps to apply on import:
   VPN-IPv4: BGP-IMPORT-VRF-SERVICES
 Route maps to apply on export:
   VPN-IPv4: BGP-EXPORT-VRF-DI-0056
 Local IP lookup MPLS VRF label: 135275
 Additional-paths installation is disabled
 Extended next-hop capability is disabled
Address-family IPv6 Unicast:
  Redistributed routes into BGP:
    Static
```

```
    Connected
  Route Distinguisher: 36351:268450419
  Route targets to import:
    VPN-IPv6:
      36351:1001
  Route targets to export:
    VPN-IPv6:
      36351:268450419
  Local IP lookup MPLS VRF label: 135896
  Additional-paths installation is disabled
```

Use the **show bgp neighbors** command to verify that the VPN address families have negotiated with the neighbor.

```
switch# show bgp neighbors
BGP neighbor is 10.0.0.2, remote AS 300, internal link
  BGP version 4, remote router ID 0.0.1.1, VRF default
  Last read 00:00:15, last write 00:00:31
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Hold timer is active, time left: 00:02:02
  Keepalive timer is active, time left: 00:00:16
  Connect timer is inactive
  Idle-restart timer is inactive
  BGP state is Established, up for 00:44:18
  Number of transitions to established: 1
  Last state was OpenConfirm
  Last event was HoldTime
  Neighbor Capabilities:
    Multiprotocol IPv4 Unicast: advertised and received and negotiated
    Multiprotocol VPN-IPv4: advertised and received and negotiated
    Multiprotocol VPN-IPv6: advertised and received and negotiated
    Four Octet ASN: advertised and received and negotiated
    Route Refresh: advertised and received and negotiated
    Send End-of-RIB messages: advertised and received and negotiated
    Additional-paths recv capability:
      IPv4 Unicast: advertised
      VPN-IPv4: advertised
      VPN-IPv6: advertised
    Additional-paths send capability:
      IPv4 Unicast: received
      VPN-IPv4: received
      VPN-IPv6: received
  Restart timer is inactive
  End of rib timer is inactive
    IPv4 Unicast End-of-RIB received: Yes
    VPN-IPv4 End-of-RIB received: Yes
    VPN-IPv6 End-of-RIB received: Yes
  Message Statistics:
                         Sent      Rcvd
    Opens:                  1         1
    Notifications:          0         0
    Updates:                6         6
    Keepalives:            53        54
    Route-Refresh:          0         0
    Total messages:        60        61
  Prefix Statistics:
                         Sent      Rcvd
    IPv4 Unicast:           1         1
    IPv6 Unicast:           0         0
  Configured maximum total number of routes is 12000
  Inbound updates dropped by reason:
    AS path loop detection: 0
```

```
    Malformed MPBGP routes: 0
    Originator ID matches local router ID: 0
    Nexthop matches local IP address: 0
Local AS is 300, local router ID 0.0.0.1
Local TCP address is 10.0.0.1, local port is 179
Remote TCP address is 10.0.0.2, remote port is 47400
```

Use the **show bgp vpn-ipv4 summary** command to show the status of VPN-IPv4 peers. While the examples below are with respect to VPN-IPv4, the same set of commands are applicable to VPN-IPv6.

```
switch# show bgp vpn-ipv4 summary
BGP summary information for VRF default
Router identifier 0.0.0.1, local AS number 300
Neighbor Status Codes: m - Under maintenance
Neighbor  V AS  MsgRcvd MsgSent InQ OutQ Up/Down State  PfxRcd PfxAcc
10.0.0.2  4 300 3379    60      0   0    1d23h   Estab  1      1
```

Use the **show bgp vpn-ipv4** command to show how the VPN-IPv4 routes sent and received.

```
switch# show bgp vpn-ipv4
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
 -
ECMP head, e - ECMP
                   S - Stale, c - Contributing to ECMP, b - backup
                   % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes:Or-ID - Originator ID,C-LST - Cluster List,LL Nexthop-Link
Local Nexthop
        Network             Next Hop         Metric  LocPref Weight Path
 * >     RD: 11.0.0.1:0 IPv4 prefix 20.0.0.0/24
                            -                -       -       0       i
 * >     RD: 11.0.1.1:0 IPv4 prefix 20.0.1.0/24
           11.0.1.1         -                100     0               i
```

Each entry in the output represent a VPN path in the VPN table. For each VPN path, the RD and actual prefix along with the nexthop information is shown. Paths in the VPN table are either received from other VPN-IPv4 peers (other PEs) or exported from local VRFs.

In the above output, ***20.0.0.0/24*** is a local route that has been exported. Notice that it has been prepended with the RD ***11.0.0.1:0*** to make it a VPN-IPv4 route. And the prefix of ***20.0.1.0/24*** has been received from another PE. It has the RD of ***11.0.1.1:0*** with a nexthop of ***11.0.1.1***. Looking at each of those prefixes in detail:

20.0.0.0/24 is a local route from one of the VRFs that has been exported. Notice that along with the prefix, the RD, the export RT and the MPLS label information is displayed. In this case, the MPLS label is a locally allocated label.

```
switch# show bgp vpn-ipv4 20.0.1.0/24
BGP routing table information for VRF default
Router identifier 0.0.0.1, local AS number 300
BGP routing table entry for IPv4 prefix 20.0.1.0/24, Route Distinguisher:
11.0.1.1:0
 Paths: 1 available
  Local
    11.0.1.1 from 10.0.0.2 (0.0.1.1)
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:300:0
      MPLS label: 100123
```

20.0.1.0/24 is a prefix that has been received from the VPN-IPv4 peer, ***10.0.0.2***. The next-hop in this case is ***11.0.1.1***. This VPN route is imported into a VRF based on the import RT configuration matching the RT received in the VPN route (***300:0***).

Note: Route-Distinguishers for the non-default VRFs must be configured under the **router bgp** mode. Route-Distinguisher configured under the VRF definition mode are ignored.

The route is installed in the VRF only when the VPN next-hop is reachable through an MPLS tunnel. The presence of such an MPLS tunnel can be verified using the **`show tunnel fib`** command. The output below shows that there is an MPLS tunnel setup by LDP to the VPN nexthop ***11.0.1.1***.

```
switch# show tunnel fib
Type 'LDP', index 1, endpoint 11.0.1.1/32, forwarding None
   via 10.0.0.2, 'Ethernet6'
      label stack 3
```

Use the **`show ip bgp vrf vrf1`** command to show the BGP table for the VRF which contains the imported VPN-IPv4 route.

```
switch# show ip bgp vrf vrf1
BGP routing table information for VRF vrf1
Router identifier 11.0.0.1, local AS number 300
Route status codes:s-suppressed,*-valid,>-active,#-not installed, E-ECMP
 head,e-ECMP
       S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
       % - Pending BGP convergence
Origin codes: i-IGP, e-EGP, ?-incomplete
AS Path Attributes: Or-ID-Originator ID, C-LST-Cluster List, LL Nexthop-Link
 Local Nexthop

         Network          Next Hop       Metric  LocPref Weight Path
 * >     20.0.0.0/24      -              -       -       0      i
 * >     20.0.1.0/24      11.0.1.1       -       100     0      i
```

Each entry in the table represents a path either locally redistributed/received into the VRF (from a BGP peer) or imported from the VPN table.

Use the **`show ip bgp 20.0.1.0/24 vrf vrf1`** command for a more detailed view of the imported IP prefix ***20.0.1.0/24***:

```
switch# show ip bgp 20.0.1.0/24 vrf vrf1
BGP routing table information for VRF vrf1
Router identifier 11.0.0.1, local AS number 300
BGP routing table entry for 20.0.1.0/24
 Paths: 1 available
  Local
    11.0.1.1 from 10.0.0.2 (0.0.1.1), imported VPN-IPv4 route, RD 11.0.1.1:0
      Origin IGP, metric -, localpref 100, weight 0, valid, internal, best
      Extended Community: Route-Target-AS:300:0
      Remote MPLS label: 100123
```

Use the **`show ip route vrf vrf1`** command to view the prefix installed in route table of the VRF:

```
switch# show ip route vrf vrf1
VRF: vrf1
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
```

```
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route
Gateway of last resort is not set
 C     20.0.0.0/24 is directly connected, Ethernet5
 B I   20.0.1.0/24 [200/0] via 11.0.1.1/32,LDP tunnel index 1,label 100123
                           via 10.0.0.2,Ethernet3,label imp-null(3)
```

The output displays both the VPN label, as well as the underlay tunnel (LDP) information.

19.2.4 Limitations

- While configuring a BGP Route-Reflector for the VPN-IPv4/VPN-IPv6 address families, the route-reflector must have transport MPLS LSPs to reach the PE nexthop addresses. Even though the route-reflector may not be in the data path and does not use the transport LSPs to forward traffic, the LSPs are required in order for the BGP nexthops to be considered as reachable valid candidates in the bestpath computation.
- When configuring eBGP peers, which receive routes over an eBGP session and re-advertise the same to a different eBGP peer, next-hop-unchanged knob must be configured, so that the original nexthop is retained.
- With iBGP route-reflector topology, next-hop-self knob must not be configured.
- Even with directly connected PEs, the VPN nexthops should be the loopback on the directly connected PEs with an MPLS tunnel between them.
  - For a VPN route to be installed in the VRF, the VPN nexthop must resolve over a MPLS tunnel.
- With OSPF as the PE-CE protocol, ***RFC 4576***, setting of the DN bit in the LSA, is not supported.
- Internal BGP (iBGP) as the PE-CE protocol, ***RFC 6368***, is not supported.

19.3 EVPN MPLS Shared ESI Label

In a multihomed EVPN MPLS configuration, BUM packets sent from a Non-Designated Forwarder (Non-DF) PE to a Designated Forwarder (DF) PE must carry the ESI label advertised by the egress DF PE. When the egress DF PE receives a packet with the ESI label that it has advertised, it does not forward the packet on the Ethernet Segment (ES) corresponding to that ESI label. This avoids sending the packet back to the same ES from which the BUM packet originated.

However, when a DF election is triggered, a PE may change its role from being Non-DF to DF. Since the DF election is run on each PE distributively, some PE(s) may not have updated their view of the new DF PE and add no ESI label in the BUM packet. In such a case, the BUM packets in transit may get sent back to the same ES where they originated. To overcome this, ESI labels are included to Non-DF PEs as well. This shared ESI label provides a mechanism to achieve sending ESI label to Non-DF PEs on platforms which cannot support sending different ESI labels to its multihomed peers.

19.3.1 Configuring EVPN MPLS Shared ESI Label

An Ethernet-Segment (ES) must be configured with a shared-index to allocate an ESI label value based on the shared-index configuration.

Example

```
switch(config)# interface Ethernet4
switch(config-if-Et1)# switchport access vlan 1000
switch(config-if-Et1)# evpn ethernet-segment
switch(config-evpn-es)# identifier 0022:2222:2222:2222:2222
switch(config-evpn-es)# mpls shared index 100
switch(config-evpn-es)# route-target import 00:02:00:02:00:02
```

- If all PEs which are provisioned with the same ES carry the same shared-index value, then they will allocate the same ESI label value for that ethernet-segment.

- Having this configuration consistent on all PEs will enable the hardware to encapsulate the ESI label on all BUM packets towards its multi-homed peers. This is essential to implement split-horizon filtering during DF role change events.
- The shared index value is used as an offset into the reserved **`l2evpn ethernet-segment`** range where the label value is derived as follows:
  - ESI label = label-range base value + shared-index value – 1

The ESI labels are allocated from **`l2evpn ethernet-segment`** label range which has a **default configuration** as shown below.

```
switch#show running-config all
mpls label range bgp-sr 900000 65536
mpls label range dynamic 100000 262144
mpls label range isis-sr 900000 65536
mpls label range l2evpn 1036288 12288
mpls label range l2evpn ethernet-segment 1031072 1024
mpls label range srlb 965536 65536
mpls label range static 16 99984
```

19.3.2 EVPN MPLS Shared ESI Label Show Commands

The following show command displays the operational state on whether the ESI label is shared amongst all the PEs in that Ethernet-Segment:

Example

```
switch# show bgp evpn instance
EVPN instance: VLAN 1000
  Route distinguisher: 100.0.0.1:1000
  Route target import: Route-Target-AS:64500:1000
  Route target export: Route-Target-AS:64500:1000
  Service interface: VLAN-based
  Local IP address: 100.0.0.1
  Encapsulation type: MPLS
  Label allocation mode: per-instance
  MAC route label: 1040210
  IMET route label: 1042201
  AD route label: 1040210
  Local ethernet segment:
    ESI: 0022:2222:2222:2222:2222
      Interface: Ethernet4
      Mode: all-active
      State: up
      ESI label: 1031171
      Shared ESI label: on
      ES-Import RT: 00:02:00:02:00:02
      Designated forwarder: 100.0.0.1
      Non-Designated forwarder: 100.0.0.2
```

- Shared ESI label state for an ES is **ON** when the following conditions are satisfied.
  - ES is configured with shared-index i.e, **`mpls shared index <value>`**.
  - All PEs in that ES carry the same ESI label.
- **`l2evpn shared ethernet-segment`** label range space can be looked up using the command below:

```
switch# show mpls label ranges
Start     End       Size      Usage
-----------------------------------------------
0         15        16        reserved
16        99999     99984     static mpls
```

```
100000    116383    16384     ldp (dynamic)
116384    132767    16384     bgp (dynamic)
132768    362143    229376    free (dynamic)
362144    899999    537856    unassigned
900000    965535    65536     isis-sr
900000    965535    65536     bgp-sr
965536    1031071   65536     srlb
1031072   1032095   1024      l2evpn shared ethernet-segment
1032096   1036287   4192      unassigned
1036288   1048575   12288     l2evpn
```

19.3.3 Limitations

When the PE switch is multihomed to more than six PEs and each one of those PEs have an ethernet segment in shared ESI configuration, the PE switch may not be able to push the ESI label into the MPLS label stack of BUM packets ingressing on every ethernet segment. In such a case, the syslog **EVPN-3-MPLS_SHARED_ESI_HW_RESOURCE_FULL** is emitted.

19.4 RSVP-TE LSR

RSVP-TE applies the Resource Reservation Protocol (RSVP) for Traffic Engineering (TE), that is, to distribute MPLS labels for steering traffic and reserving bandwidth.

The EOS implementation supports:

- RSVP-TE core protocol (***RFC 2205***, ***RFC 3209***), transit role
- Hello messages (***RFC 3209***)
- Refresh overhead reduction (***RFC 2961***)
- Cryptographic authentication (***RFC 2747***)
- Fast Reroute: facility backup, link-protection/NHOP (***RFC 4090***)
- LSP Ping/Traceroute (***RFC 4379***)
- Fast Reroute node-protection (Starting in ***EOS Release 4.22.1F***)
- Bandwidth management (Starting in ***EOS Release 4.22.1F***)
- SRLG for Fast Reroute (Starting in ***EOS Release 4.23.1F***)
- MTU signaling (Starting in ***EOS Release 4.23.1F***)
- Soft preemption (Starting in ***EOS Release 4.23.1F***)
- Support for OSPFv2 as the IGP (Starting in ***EOS Release 4.24.2F***)
- Support for LDP entropy label (Starting in ***EOS Release 4.26.0F, RFC 6790***)

19.4.1 RSVP-TE LSR Configuration

There is a dedicated configuration sub-mode for RSVP:

```
(config)# mpls rsvp
(config-mpls-rsvp)#
```

19.4.1.1 Enabling RSVP-TE

Enable RSVP-TE globally by issuing **no shutdown** in the configuration sub-mode. This is the only mandatory setting for RSVP-TE to work. This also globally enables MPLS.

```
(config-mpls-rsvp)# no shutdown
```

To disable RSVP-TE, use the **shutdown** command, which is the default.

```
(config-mpls-rsvp)# shutdown
```

There is no per-interface knob to enable and disable RSVP. However, you can only enable RSVP on interfaces that MPLS is also enabled.

19.4.1.2 State Refresh Parameter

You can configure the state refresh interval. It describes parameter R in seconds, leading to a state refresh (Path and Resv messages) every ***0.5*R*** to ***1.5*R*** seconds (randomly chosen). The default value is ***30*** seconds.

```
(config-mpls-rsvp)# refresh interval 30
```

19.4.1.3 Hello Messages

To configure the Hello message interval (seconds) and timeout multiplier (integer), refer to the following example.

In this example, hello messages are sent to all known neighbors every ***10*** seconds. If no hello responses are received from a neighbor for 4*10=40 seconds, communication is considered to be lost and the neighbor is reset.

```
(config-mpls-rsvp)# hello interval 10 multiplier 4
```

The default of ***10*** seconds with multiplier ***4*** can be reset with:

```
(config-mpls-rsvp)# default hello interval
```

To explicitly disable Hello messages, use the following command:

```
(config-mpls-rsvp)# no hello interval
```

19.4.1.4 Refresh Overhead Reduction

Setting the refresh method to **bundled** enables the Refresh Overhead Reduction (***RFC 2961***) that supports the sending of message IDs and refreshing state with refresh messages.

```
(config-mpls-rsvp)# refresh method bundled
```

This is also the default setting. To turn off refresh overhead reduction, use **explicit**.

```
(config-mpls-rsvp)# refresh method explicit
```

19.4.1.5 Cryptographic Authentication Extension

Cryptographic Authentication (***RFC 2747***) is enabled by setting the **authentication type** to **md5** and configuring an active password.

```
(config-mpls-rsvp)# authentication type md5
```

The default is **none**, which disables cryptographic authentication.

Authentication secrets are configured with an index. One of the indices should be chosen as the actively used password:

```
(config-mpls-rsvp)# authentication index 1 password s3cr3t
(config-mpls-rsvp)# authentication index 1 active
```

The **active** password is used to authenticate outgoing messages. All configured passwords are accepted for authentication of incoming packets, which allows smooth key rollover.

Password obfuscation is available:

```
(config-mpls-rsvp)# authentication index 1 password 7 07092E43
```

The size of the sequence number reorder window can be changed to accommodate a larger number of out-of-order packets. A value of *N* means that a packet is accepted if all earlier received packets with a higher sequence number are within the preceding *N-1* packets.

```
(config-mpls-rsvp)# authentication sequence-number window 5
```

The default value is ***5***. A value of ***1*** effectively turns off support for packet reordering.

19.4.1.6 Fast Reroute Extension

Support for Fast ReRoute (FRR) link protection/NHOP (***RFC 4090***) is enabled by setting the Fast Reroute mode to **link-protection**.

```
(config-mpls-rsvp)# fast-reroute mode link-protection
```

To turn off FRR, change the mode to the default setting **none**.

```
(config-mpls-rsvp)# fast-reroute mode none
```

This setting only applies to Point of Local Repair (PLR) behavior, for example, the router is the upstream node relative to a to-be-protected link over which an LSP is routed. Merge Point (MP) behavior is always enabled and not affected by this setting.

Support for Node Protection is present starting in ***EOS Release 4.22.1F***, so the above CLI also accepts an additional mode **node-protection**.

You can change the revertive behavior of the FRR from the **global** revertive mode to the **local** revertive mode. In the **global** revertive mode, an LSR that is re-routed over a bypass tunnel because its downstream link is dead keeps using the bypass tunnel even after the link has recovered. This expects the headend router to set up a new LSP upon notification that a link is not available anymore. In the **local** revertive mode, the LSR switches back to using the primary link after recovery.

```
(config-mpls-rsvp)# fast-reroute reversion local
```

The default for reversion is **global**.

```
(config-mpls-rsvp)# fast-reroute reversion global
```

19.4.1.7 Shared Risk Link Group

Starting in ***EOS Release 4.23.1F***, there is an additional config command to configure SRLG mode.

srlg [strict]

no srlg [strict]

`default srlg` [strict]

This command specifies if link SRLGs of a primary LSP are to be considered as constraints while creating a fast-reroute bypass tunnel with either link or node protection. When **`srlg`** is specified with the **strict** keyword, then if a path for a bypass tunnel excluding SRLGs of the next-hop interface of primary LSP can not be found, RSVP does not setup the bypass tunnel. When **`srlg`** is specified without a **strict** keyword, then a bypass tunnel is setup with as many links as possible that exclude the SRLGs of the next-hop interface of primary LSP and where such links are not available, links that have the least number of SRLGs which are to be excluded are used.

When this CLI is not configured, the behavior remains the same as before, which is to turn off SRLG processing. Therefore, the **`no`** and **`default`** versions of the command take you back to the default of SRLG processing being turned off.

The SRLGs of an interface can be configured using the following traffic-engineering CLI as below:

```
(config)# interface Et1
(config-if-Et1)# traffic-engineering srlg 100

(config)# interface Et2
(config-if-Et2)# traffic-engineering srlg 200
```

Note: Although a large number of SRLGs can be configured on an interface (ranging from ***0*** to ***2^32 -1***), CSPF only takes into account a maximum of ***16*** SRLGs for finding a path with SRLG constraints.

19.4.1.8 CSPF Configuration Command

RSVP-TE uses CSPF to compute the FRR backup path. CSPF can be throttled to avoid frequent path changes when there are frequent network events. The following CLI command is used under the **`router traffic-engineering`** mode to specify how frequently CSPF is to run after a network event by specifying the initial wait interval, back-off interval, and maximum wait interval for CSPF.

`cspf delay` [initial *n* back-off ***n***] max ***n***

All three values are in milliseconds and default to ***100***, ***200***, and ***1000*** milliseconds, respectively.

19.4.1.9 Traffic-engineering Configuration Commands

For CSPF to compute the FRR backup path to a particular destination, enable traffic-engineering on all the IGP enabled interfaces, and enable globally the traffic engineering extensions. For IS-IS, enable under ***router isis*** mode, and for OSPFv2, enable under ***router ospf***. This is required for the IGP to start exchanging TE-related attributes with peers and build its topology database.

Also, CSPF needs some notion of router ID to uniquely identify a router, so that it can find a path to it. CSPF also uses self router-id as a source for running SPF. So a router-id is required to be configured under ***router traffic-engineering*** mode for CSPF to function properly.

19.4.1.10 MTU Signaling

Starting in ***EOS Release 4.23.1F***, MTU along a path can be signaled using RSVP.

Example

`mtu signaling`

`no mtu signaling`

`default mtu signaling`

RSVP can discover the lowest MTU used along an LSP by evaluating and updating, at each hop, the composed MTU, stored in the General Parameters fragment of an AdSpec object in RSVP Path messages.

At each RSVP hop, when MTU signaling is enabled, the value is updated by taking the minimum between the downstream interface for an LSP and the composed MTU value from the incoming Path message.

When the feature is disabled, the composed MTU is not updated.

19.4.1.11 Soft Preemption

Starting in ***EOS Release 4.23.1F***, soft preemption is supported, which enables deferred failure of RSVP-TE LSPs on link oversubscription. A preemption timer value can be used to configure a delay on a transit router to support LSPs signaled with soft preemption enabled by the headend (***RFC 5712***).

`preemption method` [hard | soft] timer *t*

`no preemption method` [hard | soft] timer *t*

`default preemption method` [hard | soft] timer *t*

The default preemption method is soft preemption with a timer value of ***30*** seconds.

In this example, the preemption method is soft preemption with a timer value of ***10*** seconds.

```
(config-mpls-rsvp)# preemption method soft timer 10
```

Setting the preemption method to hard preemption results in a timer value of ***0*** seconds and disables the feature.

```
(config-mpls-rsvp)# preemption method hard
```

Sample Configuration

```
switch(config)# ip routing
!
switch(config)# mpls ip
!
switch(config)# interface Ethernet1
switch(config-if-Et1)# no switchport
switch(config-if-Et1)# ip address 10.0.0.1/24
switch(config-if-Et1)# isis enable isis
switch(config-if-Et1)# traffic-engineering
switch(config-if-Et1)# traffic-engineering bandwidth 100 percent
switch(config-if-Et1)# traffic-engineering metric 5
!
switch(config)# router traffic-engineering
switch(config-te)# router-id ipv4 0.1.1.1
!
switch(config)# router isis isis
switch(config-router-isis)# net 49.0000.0000.0000.1111.00
switch(config-router-isis)# is-type level-2
   !
switch(config-router-isis)# address-family ipv4 unicast
switch(config-router-isis-af)# maximum-paths 32
   !
switch(config-router-isis-af-te)# traffic-engineering
switch(config-router-isis-af-te)# no shutdown
switch(config-router-isis-af-te)# is-type level-2
!
switch(config)# mpls rsvp
switch(config-mpls-rsvp)# no shutdown
!
```

The following displays the overall state of RSVP.

```
switch> show mpls rsvp
Administrative state: enabled
Operational state: up
Refresh interval: 30 seconds
Refresh reduction: enabled
Hello messages: enabled
   Hello interval: 10 seconds
   Hello multiplier: 4
Fast Re-Route: disabled
   Mode: none
   Hierarchical FECs: enabled
Cryptographic authentication: disabled
MTU signaling: disabled
Number of sessions: 1
   Ingress/Transit/Egress: 0/1/0
Number of LSPs: 1
   Operational: 1
   Ingress/Transit/Egress: 0/1/0
   Currently using bypass tunnels: 0
Number of bypass tunnels: 0
Number of neighbors: 2
Number of interfaces: 2
```

The following displays the RSVP neighbors.

```
switch> show mpls rsvp neighbor
Neighbor 10.0.1.1
   Upstream for
      Session #1 LSP #1
   Downstream for
   Neighbor uptime: 00:01:24
   Authentication type: disabled
   Last hello received: 1 seconds ago
   Last hello sent: 1 seconds ago
   Bypass tunnel: not requested
Neighbor 10.0.2.2
   Upstream for
   Downstream for
   Session #1 LSP #1
   Neighbor uptime: 00:01:24
   Authentication type: disabled
   Last hello received: -
   Last hello sent: 31 seconds ago
   Bypass tunnel: not requested
```

The following displays a summary of RSVP neighbors by IP address.

```
switch> show mpls rsvp neighbor summary
Neighbor                                Role    Sessions LSPs
======================================= ========== ======== ========
10.0.1.1                                Upstream   1     1
10.0.2.2                                Downstream 1     1
```

The following displays RSVP sessions.

```
switch> show mpls rsvp session
Session #1
   Destination address: 0.4.4.4
   Tunnel ID: 0
   Extended Tunnel ID: 0.1.1.1
```

```
   Role: transit
   LSP #1
        State: up
        Type: primary
        Source address: 0.1.1.1
        LSP ID: 1
        LSP uptime: 00:02:38
        Session name: Session1
        Local label: 100000
        Downstream label: 100000
        Upstream neighbor: 10.0.1.1
           Last refresh received: 17 seconds ago
           Last refresh sent: 10 seconds ago
        Downstream neighbor: 10.0.2.2
           Last refresh received: 7 seconds ago
           Last refresh sent: 9 seconds ago
        Bypass tunnel: not requested
```

The following displays a summary of RSVP sessions.

```
switch> show mpls rsvp session summary
Session  Destination   LSP      Name         Role    Bypass State
======== ============= ======== =========== ======= ====== ======
1         0.4.4.4       1        Session1    transit n/req  up
```

The following displays details of RSVP sessions.

```
switch> show mpls rsvp session detail
Session #1
   Destination address: 0.4.4.4
   Tunnel ID: 0
   Extended Tunnel ID: 0.1.1.1
   Role: transit
   LSP #1
        State: up
        [...]
        MTU Signaling: enabled
           Received Path MTU: 1800 bytes
           Sent Path MTU: 1500 bytes
  [...]
```

The following displays RSVP message counters per interface.

```
switch> show mpls rsvp counters
Received Messages:

Interface  Path  PathTear  PathErr  Resv  ResvTear  ResvErr  Srefresh  Other  Errors
---------  ----  --------  -------  ----  --------  -------  --------  -----  ------
Ethernet1  5     0         0        0     0         0        8         51     1
Ethernet2  0     0         0        14    0         0        0         0      0

Sent Messages:

Interface  Path  PathTear  PathErr  Resv  ResvTear  ResvErr  Srefresh  Other  Errors
---------  ----  --------  -------  ----  --------  -------  --------  -----  ------
Ethernet1  0     0         0        4     0         0        9         49     0
Ethernet2  13    0         0        0     0         0        0         4      0
```

19.4.1.12 Path Specifications

Path specifications can be **explicit** and **dynamic** depending on Constrained Shortest Path First (CSPF) search procedure to find a path in the network topology.

Explicit Path Specifications

The operator provides all hops in the path explicitly. The given path is used directly as the Explicit Route Object (ERO) in RSVP Path messages. All hops are implicitly **strict** hops. Explicit loose hops are not supported. The submode to configure explicit paths is entered by specifying the name.

```
switch(config-te-rsvp)# path MyPath explicit
switch(config-te-rsvp-path-expl-MyPath)#
```

Each hop is specified explicitly in order.

```
switch(config-te-rsvp-path-expl-MyPath)# hop 10.0.12.2
(config-te-rsvp-path-expl-MyPath)# hop 10.0.34.4
```

Adding hops with **before** and **after** create a unique internal ordering in **`show running-config`**.

```
switch(config-te-rsvp-path-expl-MyPath)# hop 10.0.23.3 before 10.0.34.4
switch(config-te-rsvp-path-expl-MyPath)# hop 10.0.45.5 after 10.0.34.4
```

`no hop` command removes the hop.

```
switch(config-te-rsvp-path-expl-MyPath)# no hop 10.0.12.2
```

`show active` command retrieves the currently active configuration. The pending configuration that becomes active upon exiting the submode is retrieved with **`show pending`** command; the difference between these two can be retrieved with **`show diff`** command.

```
switch(config-te-rsvp-path-expl-MyPath)# show active
router traffic-engineering
   rsvp
      path MyPath explicit
         hop 10.0.23.3
         hop 10.0.34.4
         hop 10.0.45.5
(config-te-rsvp-path-expl-MyPath)# show pending
   hop 10.0.23.3
   hop 10.0.44.4
   hop 10.0.45.5
(config-te-rsvp-path-expl-MyPath)# show diff
 hop 10.0.23.3
-hop 10.0.34.4
+hop 10.0.44.4
 hop 10.0.45.5
(config-te-rsvp-path-expl-MyPath)#
```

`exit` command keeps the changes in the submode.

```
switch(config-te-rsvp-path-expl-MyPath)# exit
switch(config-te-rsvp)#
```

`abort` command discards the changes in the submode.

```
switch(config-te-rsvp-path-expl-MyPath)# abort
switch(config-te-rsvp)#
```

Dynamic Path Specifications

In a dynamic path specification, the operator provides constraints with which a CSPF procedure finds a path in the network topology. The CSPF result is a list of strict hops which forms the ERO.

The submode to configure dynamic paths is entered by specifying the name with **dynamic**.

```
switch(config-te-rsvp)# path MyPath dynamic
switch(config-te-rsvp-path-dyn-MyPath)#
```

Exclude hop constraints specify that CSPF must not choose the specified address on the path. Note that other interfaces on the same node may be used.

```
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.56.6 exclude
```

Include hop constraints specify these hops included in the computed path in a certain order. A hop can be loose which allows other hops to be filled by the CSPF procedure. The keywords **before** and **after** work in the submode for explicit paths.

```
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.23.3
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.45.5 loose
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.12.2 before 10.0.23.3
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.67.7 loose after 10.0.45.5
```

19.4.1.13 Tunnel Specifications

Tunnel specification has its own sub-mode.

```
switch(config-te-rsvp)# tunnel MyTunnel
switch(config-te-rsvp-tunnel-MyTunnel)#
```

Each tunnel has a tunnel destination IP.

```
switch(config-te-rsvp-tunnel-MyTunnel)# destination ip 10.2.2.2
```

A tunnel specifies for path for its LSP(s). It configures the **primary** LSP.

```
switch(config-te-rsvp-tunnel-MyTunnel)# path MyPath
```

A **secondary** LSP is configured, when the **primary** LSP is not available. Only one secondary path can be configured per tunnel.

```
switch(config-te-rsvp-tunnel-MyTunnel)#path MyOtherPath secondary pre-signaled
```

A tunnel reserves bandwidth along the path. The bandwidth can be configured explicitly.

```
switch(config-te-rsvp-tunnel-myTunnel)# bandwidth 10 mbps
```

Auto bandwidth specifies the minimum and maximum bandwidth used for the tunnel and adjusts bandwidth with an **adjustment period** based on the observed traffic going over the tunnel. The adjustment period is specified in seconds.

```
switch(config-te-rsvp-tunnel-myTunnel)# bandwidth auto min 1 mbps max 5 mbps
 adjustment-period 60
```

Setup and hold priorities from ***0*** to ***7*** can be configured for the tunnel, where ***0*** is ***most preferred*** and ***7*** the ***least preferred***.

```
switch(config-te-rsvp-tunnel-myTunnel)# priority setup 5 hold 3
```

By default, a tunnel is not enabled.

```
switch(config-te-rsvp-tunnel-myTunnel)# no shutdown
```

Changes in the sub-mode only take effect when the sub-mode is exited normally for **show active**, **show pending** and **show diff** commands.

```
switch(config-te-rsvp-tunnel-myTunnel)# exit
switch(config-te-rsvp)#
```

Changes are discarded when sub-mode is aborted.

```
switch(config-te-rsvp-tunnel-myTunnel)# abort
switch(config-te-rsvp)#
```

Periodic tunnel optimization is configured globally as well as individually for a specific tunnel. For global periodic tunnel optimization, the optimization interval is configured in ***config-te-rsvp*** mode, and this configuration automatically gets applied to all the RSVP tunnels.

```
switch(config-te-rsvp)# optimization interval 3600 seconds
switch(config-te-rsvp)#
```

The optimization interval can also be individually configured on a tunnel by entering the command in ***tunnel*** sub-mode. The optimization interval configured in the ***tunnel*** sub-mode overrides the optimization interval configured globally.

```
switch(config-te-rsvp)# tunnel myTunnel
switch(config-te-rsvp-tunnel-myTunnel)# optimization interval 3600 seconds
```

Optimization can also be disabled on a specific tunnel. Optimization is enabled for all the tunnels with a common interval but a specific tunnel has optimization disabled.

```
switch(config-te-rsvp)# optimization interval 3600 seconds
switch(config-te-rsvp)# tunnel myTunnel
switch(config-te-rsvp-tunnel-myTunnel)# optimization disabled
```

alias IPv4 and IPv6 endpoints per tunnel allow these additional next hops to resolve using the existing tunnel. The command allows a maximum of ***15*** alias endpoints to be configured per tunnel.

```
switch(config-te-rsvp-tunnel-myTunnel)# alias endpoint 5.5.5.5
switch(config-te-rsvp-tunnel-myTunnel)# alias endpoint 2001::10
```

The **split-tunnel** command allows splitting the bandwidth for a tunnel between multiple sessions. With this enabled, RSVP automatically creates multiple LSPs with smaller bandwidth reservations for a single tunnel.

```
switch(config-te-rsvp-tunnel-myTunnel)# split-tunnel quantum 10 kbps
switch(config-te-rsvp-tunnel-myTunnel)# split-tunnel quantum 10 kbps sub-
tunnels limit 20
```

IGP Shortcut in IS-IS

IGP shortcuts enable traffic to get forwarded along paths computed to take advantage of traffic-engineered paths setup using RSVP using a modified SPF algorithm. This enables operators to take advantage of TE capabilities of RSVP tunnels which traverse over links satisfying bandwidth, latency or fast reroute considerations.

IGP shortcuts are enabled on a RSVP Label Edge Router (LER) selectively on specific RSVP tunnels. When IGP shortcut is enabled, IP routes resolving over RSVP tunnels are installed in FIB. As a result all IP traffic including control plane traffic is forwarded through IGP shortcut tunnels. All protocols relying on FIB for nexthop resolution such as Static routes or BGP is also resolved over IGP shortcuts.

Configuring IGP Shortcut on RSVP Tunnel

On a RSVP LER, IGP shortcut is enabled individually on each RSVP tunnel that is intended for them. No configuration changes are needed on any of the RSVP Label Switch Routers.

```
switch(config)# router traffic-engineering
switch(config-te)# rsvp
switch(config-te-rsvp)# tunnel R3b
switch(config-te-rsvp-tunnel-T1)# igp shortcut
switch(config-te-rsvp-tunnel-T1)# exit
```

IGP shortcuts is enabled by default in IS-IS, following command disables this under the IS-IS ***address-family ipv4*** mode.

```
switch# conf terminal
switch(config)# router isis inst1
switch(config-router-isis)# address-family ipv4
switch(config-router-isis-af)# igp shortcut disabled
```

Limitations

- IGP shortcut is only supported for IPv4.
- IGP shortcuts can only be computed over RSVP tunnels'
- IS-IS must be enabled on the loopback interface whose primary IP address is used as the RSVP tunnel end point.
- This is only available in the multi-agent routing protocol model.
- TI-LFA protection is not enabled for Segment Routing destinations reachable via IGP shortcuts.

Show Commands

The **`show traffic-engineering rsvp tunnel`** command gives the information about the paths for its LSP(s).

```
switch# show traffic-engineering rsvp tunnel
Tunnel TestTunnel
   Source: 10.1.1.1
   Destination: 10.4.4.4
   State: up
   Bandwidth: 0.0 bps, mode explicit
   LSPs: 2
   Active path: primary
   Primary path: Path1to4
      State: up, in use
      Path (dynamic):
         10.0.12.2
         10.0.23.3
         10.0.34.4
   Secondary path: Path1to4detour
      State: up
      Path (explicit):
         10.0.16.6
         10.0.67.7
         10.0.37.3
```

```
         10.0.34.4
```

The **show traffic-engineering rsvp tunnel lsp** command gives the details of the paths for its LSP(s).

```
switch(config)# show traffic-engineering rsvp tunnel lsp
Tunnel TestTunnel
   Source: 10.1.1.1
   Destination: 10.4.4.4
   State: up
   Bandwidth: 0.0 bps, mode explicit
   LSPs: 2
   Active path: primary
   LSP 1:
      Path specification: Path1to4, primary
      CSPF Path ID: 10001
      Bandwidth: 0.0 bps
      State: up, in use
      Path (dynamic):
         10.0.12.2
         10.0.23.3
         10.0.34.4
   LSP 2:
      Path specification: Path1to4detour, secondary
      Bandwidth: 0.0 bps
      State: up
      Path (explicit):
         10.0.16.6
         10.0.67.7
         10.0.37.3
         10.0.34.4
```

The **show traffic-engineering rsvp tunnel detail** command gives the details of the tunnel.

```
switch# show traffic-engineering rsvp tunnel detail
Tunnel TestTunnel
   Source: 10.1.1.1
   Destination: 10.4.4.4
   Additional endpoints:
      5.5.5.5
      2001::10
   State: up
   Bandwidth: 0.0 bps, mode explicit
      Setup priority: 7
      Hold priority: 0
   MTU signaling: disabled
Periodic optimization: disabled
   Session #4
   Tunnel index: 1
   LSPs: 2
LDP tunneling: enabled
   IGP shortcut: enabled
   Active path: primary
   Primary path: Path1to4
      State: up, in use
      CSPF Path ID: 10001
      Path (dynamic):
         10.0.12.2
         10.0.23.3
         10.0.34.4
   Secondary path: Path1to4detour
      State: up
```

```
      Path (explicit):
         10.0.16.6
         10.0.67.7
         10.0.37.3
         10.0.34.4
```

The **show traffic-engineering rsvp tunnel history** command gives the history for the paths.

```
switch# show traffic-engineering rsvp tunnel history
Tunnel TestTunnel
Mon 2020-07-13 07:04:44 CSPF query on primary path
Mon 2020-07-13 07:04:44 State change: down
Mon 2020-07-13 07:04:44 LSP #1 added
Mon 2020-07-13 07:04:44 CSPF reply for primary path, path found
Mon 2020-07-13 07:04:44 LSP #2 added
Mon 2020-07-13 07:04:46 State change: up using primary path
LSP #1
Mon 2020-07-13 07:04:44 LSP created
Mon 2020-07-13 07:04:44 State change: establishing
Mon 2020-07-13 07:04:46 State change: up
LSP #2
Mon 2020-07-13 07:04:44 LSP created
Mon 2020-07-13 07:04:44 State change: establishing
Mon 2020-07-13 07:04:47 State change: up
```

The **show isis summary** command shows the status of IGP shortcut configuration if IGP shortcut is enabled for IS-IS.

```
switch# show isis summary
IS-IS Instance: inst1 VRF: default
  Instance ID: 0
  System ID: 1111.1111.1001, administratively enabled
  Router ID: IPv4: 1.0.5.1
  Multi Topology disabled, not attached
  IPv4 Preference: Level 1: 115, Level 2: 115
  IPv6 Preference: Level 1: 115, Level 2: 115
  IS-Type: Level 2, Number active interfaces: 3
  Routes IPv4 only
  LSP size maximum: Level 1: 1492, Level 2: 1492
  …
  Shortcut SPF for IGP: Enabled
  …
```

The **show isis network topology** command verifies the best path details computed after IS-IS SPF. RSVP tunnel details are shown if the destination is reachable through a IGP shortcut.

```
switch# show isis network topology
IS-IS Instance: inst1 VRF: default
 IS-IS paths to level-2 routers
  System Id        Metric   IA Metric Next-Hop         Interface
  SNPA
  1111.1111.1003   10    0         1111.1111.1003   RSVP LER tunnel index 5
 IGP Shortcut
```

19.4.1.14 Show Commands

19.4.1.14.1 RSVP Show Commands

Use the **show mpls rsvp** command to display the overall state of the RSVP.

```
switch> show mpls rsvp
Administrative state: enabled
Operational state: up
Refresh interval: 30 seconds
Refresh reduction: enabled
Hello messages: enabled
   Hello interval: 10 seconds
   Hello multiplier: 4
Fast Re-Route: disabled
   Mode: none
   Hierarchical FECs: enabled
Cryptographic authentication: disabled
MTU signaling: disabled
Number of sessions: 1
   Ingress/Transit/Egress: 0/1/0
Number of LSPs: 1
   Operational: 1
   Ingress/Transit/Egress: 0/1/0
   Currently using bypass tunnels: 0
Number of bypass tunnels: 0
Number of neighbors: 2
Number of interfaces: 2
```

Use the **show mpls rsvp neighbor** command to display RSVP neighbors.

```
switch> show mpls rsvp neighbor
Neighbor 10.0.1.1
   Upstream for
      Session #1 LSP #1
   Downstream for
   Neighbor uptime: 00:01:24
   Authentication type: disabled
   Last hello received: 1 seconds ago
   Last hello sent: 1 seconds ago
   Bypass tunnel: not requested
Neighbor 10.0.2.2
   Upstream for
   Downstream for
   Session #1 LSP #1
   Neighbor uptime: 00:01:24
   Authentication type: disabled
   Last hello received: -
   Last hello sent: 31 seconds ago
   Bypass tunnel: not requested
```

Use the **show mpls rsvp neighbor summary** command to display neighbors filtered by IP address.

```
switch> show mpls rsvp neighbor summary
Neighbor             Role       Sessions LSPs
==================== ========== ======== ========
10.0.1.1             Upstream   1        1
10.0.2.2             Downstream 1        1
```

or use the **show mpls rsvp session** command.

```
switch> show mpls rsvp session
Session #1
   Destination address: 0.4.4.4
   Tunnel ID: 0
   Extended Tunnel ID: 0.1.1.1
   Role: transit
   LSP #1
        State: up
        Type: primary
        Source address: 0.1.1.1
        LSP ID: 1
        LSP uptime: 00:02:38
        Session name: Session1
        Local label: 100000
        Downstream label: 100000
        Upstream neighbor: 10.0.1.1
           Last refresh received: 17 seconds ago
           Last refresh sent: 10 seconds ago
        Downstream neighbor: 10.0.2.2
           Last refresh received: 7 seconds ago
           Last refresh sent: 9 seconds ago
        Bypass tunnel: not requested
```

Session #ID and LSP #ID are internal values that are locally significant. Use these to filter sessions in the **show mpls rsvp session summary** command and in LSP ping and traceroute commands (see below). Sessions can further be filtered by name, destination, router role (transit/ingress/egress), and state.

```
switch> show mpls rsvp session summary
Session  Destination        LSP      Name         Role    Bypass State
======== ================== ======== ============ ======= ====== =========
1        0.4.4.4             1        Session1    transit n/req  up
```

Beginning with ***EOS Release 4.23.1F***, use the **show mpls rsvp session detail** command to display detailed information of RSVP sessions.

```
switch> show mpls rsvp session detail
Session #1
   Destination address: 0.4.4.4
   Tunnel ID: 0
   Extended Tunnel ID: 0.1.1.1
   Role: transit
   LSP #1
        State: up
        [...]
        MTU Signaling: enabled
           Received Path MTU: 1800 bytes
           Sent Path MTU: 1500 bytes
  [...]
```

Use the **show mpls rsvp counters** command to display RSVP message counters by interface.

```
switch> show mpls rsvp counters
Received Messages:

Interface  Path  PathTear  PathErr  Resv  ResvTear  ResvErr  Srefresh  Other
 Errors
---------  ----  --------  -------  ----  --------  -------  --------  -----
 ------
Ethernet1  5     0         0        0     0         0        8         51     1
```

```
Ethernet2  0     0         0        14    0         0        0         0      0


Sent Messages:

Interface  Path  PathTear  PathErr  Resv  ResvTear  ResvErr  Srefresh  Other
 Errors
---------  ----  --------  -------  ----  --------  -------  --------  -----
 ------
Ethernet1  0     0         0        4     0         0        9         49     0

Ethernet2  13    0         0        0     0         0        0         4
 0
```

19.4.1.14.2 CSPF Show Commands

RSVP-TE uses CSPF to compute the FRR backup path. Starting from ***EOS Release 4.23.1F***, if **srlg** is configured in RSVPconfig to exclude SRLG, then details about the SRLG related constraint attributes are also shown as in the following show commands.

Use the **`show traffic-engineering cspf path`** command to display all the paths computed by CSPF. In the following example, path ***20.0.0.1*** is selected for display.

```
switch> show traffic-engineering cspf path 20.0.0.1 20.0.0.1

Destination     Constraint                          Path
20.0.0.1        exclude Ethernet1                   0.1.1.1
                exclude SRLG of Ethernet1           0.1.1.2
                                                    0.2.2.2
                                                    3.3.3.2
                exclude Ethernet2                   0.1.1.1
                                                    0.1.1.2
                                                    0.2.2.2
                                                    3.3.3.2
```

Using the **`show traffic-engineering cspf path detail`** command, path ***20.0.0.1*** is displayed in detail.

```
switch> show traffic-engineering cspf path 20.0.0.1 detail

Destination: 20.0.0.1
   Path Constraint: exclude Ethernet1
                    exclude SRLG of Ethernet1: orange-link (500),
                       green-link (400), 100, red-link (200), 600
   Request Sequence number: 1
   Response Sequence number: 1
   Number of times path updated: 2
   Last updated: 00:01:58
   Reoptimize: Always

   Path:
   0.1.1.1
   0.1.1.2
   0.2.2.2
   3.3.3.2

      Path Constraint: exclude Ethernet2
      Request Sequence number: 2
      Response Sequence number: 2
      Number of times path updated: 3
      Last updated: 00:00:38
      Reoptimize: Always
```

```
      Path:
   0.1.1.1
   0.1.1.2
   0.2.2.2
   3.3.3.2
```

19.4.1.14.2.1 Displaying the Traffic-engineering Database

You can display the topology used for CSPF computations by using the **`show traffic-engineering database`** command. Starting from ***EOS Release 4.23.1F***, the SRLG group details of a neighbor are shown if it is advertised.

Beginning with ***EOS Release 4.24.2F***, information for the OSPFv2 topology, if configured, displays.

```
switch# show traffic-engineering database

TE Router-ID: 1.0.0.2
  Source: IS-IS Level-1 IPv4 Topology Database
    IS-IS System-ID: 1111.1111.1001
      Number of Links: 2
        Network type: P2P
          Neighbor: 1111.1111.1002
   Administrative group (Color): 0x123a
            TE Metric: 30
            IPv4 Interface Addresses:
              20.20.20.1
              192.168.20.1
            IPv4 Neighbor Addresses:
              20.20.20.2
              192.168.20.2
            Maximum link BW: 25.00 Gbps
            Maximum reservable link BW: 10.00 Mbps
            Unreserved BW:
              TE class 0: 9.00 Mbps     TE class 1: 9.00 Mbps
              TE class 2: 8.5.00 Mbps   TE class 3: 8.00 Mbps
              TE class 4: 7.00 Mbps     TE class 5: 7.50 Mbps
              TE class 6: 6.00 Mbps     TE class 7: 6.00 Mbps

        Network Type: LAN
          Neighbor: 1111.1111.1003.02
            TE Metric: 30
         Administrative Group: 0x12
            IPv4 Local Addresses:
              30.30.30.1

            Maximum Link BW: 10.00 Gbps

            Maximum Reservable Link BW: 10.50 Gbps

            Unreserved BW:

              TE-Class 0: 8.50 Gbps       TE-Class 1: 8.70 Gbps

              TE-Class 2: 7.50 Gbps       TE-Class 3: 7.25 Gbps

              TE-Class 4: 6.50 Gbps       TE-Class 5: 7.30 Gbps

              TE-Class 6: 3.50 Gbps       TE-Class 7: 7.20 Gbps
  Source: IS-IS Level-2 IPv4 Topology Database
    IS-IS System-ID: 1111.1111.1003
      Number of Links: 1
        Network Type: LAN
```

```
         Neighbor: 1111.1111.1003.16
           IPv4 Local Addresses:
             40.40.40.1
           Maximum link BW: 10.00 Gbps
           Maximum reservable link BW: 5.00 Gbps
           Unreserved BW:
             TE class 0: 4.00 Gbps     TE class 1: 4.00 Gbps
             TE class 2: 4.00 Gbps     TE class 3: 4.00 Gbps
             TE class 4: 3.00 Gbps     TE class 5: 3.00 Gbps
             TE class 6: 3.00 Gbps     TE class 7: 3.00 Gbps
  Source: OSPFv2 Instance ID 33 Area-ID 0.0.0.0 Topology Database
    OSPFv2 Router-ID: 1.2.3.4
      Number of Links: 2
        Network type: P2P
          Neighbor: 3.4.5.6
            Administrative group (Color): 0x123a
            TE metric: 30
            IPv4 Interface Addresses:
              20.20.20.1
              192.168.20.1
            IPv4 Neighbor Addresses:
              20.20.20.2
              192.168.20.1
            Maximum link BW: 25.00 Gbps
            Maximum reservable link BW: 10.00 Mbps
            Unreserved BW:
              TE class 0: 9.00 Mbps    TE class 1: 9.00 Mbps
              TE class 2: 8.50 Mbps    TE class 3: 8.00 Mbps
              TE class 4: 7.00 Mbps    TE class 5: 7.50 Mbps
              TE class 6: 6.00 Mbps    TE class 7: 6.00 Mbps
        Network type: LAN
          Neighbor: 2.3.4.5
            Administrative group (Color): 0x12
            TE metric: 30
            IPv4 Interface Addresses:
              30.30.30.1
            IPv4 Neighbor Addresses:
              0.0.0.0
            Maximum link BW: 10.00 Gbps
            Maximum reservable link BW: 10.5 Gbps
            Unreserved BW:
              TE class 0: 8.50 Gbps    TE class 1: 8.70 Gbps
              TE class 2: 7.50 Gbps    TE class 3: 7.25 Gbps
              TE class 4: 6.50 Gbps    TE class 5: 7.30 Gbps
              TE class 6: 3.50 Gbps    TE class 7: 7.20 Gbps

TE Router-ID: 1.0.0.3
  Source: IS-IS Level-1 IPv4 Topology Database
    IS-IS System-ID: 1111.1111.1004
      Number of Links: 2
        Network type: P2P
          Neighbor: 1111.1111.1002
          IPv4 Interface Addresses:
              1.0.5.1
            IPv4 Neighbor Addresses:
              1.0.5.2
            Maximum link BW: 50.00 Gbps
            Maximum reservable link BW: 10.00 Gbps
            Unreserved BW:
              TE class 0: 8.00 Gbps     TE class 1: 8.00 Gbps
              TE class 2: 8.00 Gbps     TE class 3: 8.00 Gbps
              TE class 4: 7.00 Gbps     TE class 5: 7.00 Gbps
              TE class 6: 7.00 Gbps     TE class 7: 7.00 Gbps
            Shared Risk Link Group:
```

```
                Group: 100
                Group: green-link (150)
```

19.4.1.15 Limitations

RSVP-TE LSR contains the following limitations:

- Supports only IPv4.
- Supports only the default VRF.
- Supports only strict EROs with host hops (***/32***).
- Supports only transit role functionality.
- Traffic Engineering links with secondary IP addresses are not supported.
- The maximum number of supported RSVP sessions and LSPs is ***5000***.
- For FRR, the maximum number of supported LSPs per interface is ***2000***.
- Changing the RSVP FRR mode while a bypass tunnel is already in use can bring down both primary and bypass tunnels which cause traffic loss.
- Changing the RSVP SRLG mode while a bypass tunnel is already in use could bring down both primary and bypass tunnels which cause traffic loss.
- The maximum number of IS-IS/OSPFv2 routers supported by CSPF on a single broadcast network is ***30***.
- The maximum number of IS-IS/OSPFv2 adjacencies supported by CSPF for a single router is ***500***.
- CSPF only takes into account a maximum of ***16*** SRLGs per TE link, when excluding SRLGs while computing a path.

19.5 RSVP-TE LER

RSVP-TE, the Resource Reservation Protocol (RSVP) for Traffic Engineering (TE), is used to distribute MPLS labels for steering traffic and reserving bandwidth. The Label Edge Router (LER) feature implements the headend functionality, such as, RSVP-TE tunnels can originate at an LER which is used to steer traffic into the tunnel.

To Configure RSVP-TE LER

To configure RSVP-TE LER, use the traffic-engineering mode to configure the **`rsvp`** command.

Example

```
switch(config)# router traffic-engineering
switch(config-te)# rsvp
switch(config-te-rsvp)
```

In the RSVP-TE submode, the configuration has three components.

- Global configuration
- Path specifications
- Tunnel specifications

Global Configuration

In global configuration, all settings apply to all configured tunnels. Path specifications describe a set of constraints for paths. These are referenced from tunnel specifications to describe which path each tunnel can take.

Path Specifications

Path specifications are **explicit** and **dynamic** which differ in whether they involve a Constrained Shortest Path First (CSPF) search procedure to find a path in the network topology known to the headend. A path specification of a certain name is only either **explicit** or **dynamic**, such as, they share a namespace.

Explicit Path Specifications

In an explicit path specification, the operator provides all hops in the path explicitly. The given path is used directly as the Explicit Route Object (ERO) in RSVP Path messages. All hops are implicitly strict hops. Explicit loose hops are not supported.

Example

The submode to configure explicit paths is entered by specifying the name and the **explicit** keyword.

```
switch(config-te-rsvp)# path MyPath explicit
switch(config-te-rsvp-path-expl-MyPath)#
```

To explicitly specify the order of each hop, use the RSVP-TE explicit path configuration mode. In this example, hop 10.0.12.2. then hop 10.0.34.4 is specied.

Example

```
switch(config-te-rsvp-path-expl-MyPath)# hop 10.0.12.2
switch(config-te-rsvp-path-expl-MyPath)# hop 10.0.34.4
```

Adding hops with **before** and **after** keywords creates a unique internal ordering which will be represented in a canonical form using neither before nor after in the **`show running-config`**.

Example

Using the keyword **before**, the configuration places hop ***10.0.23.3*** before ***10.0.34.4***. Then, using the keyword **after** the configuration places hop ***10.0.45.5*** after ***10.0.34.4***.

```
switch(config-te-rsvp-path-expl-MyPath)# hop 10.0.23.3 before 10.0.34.4
switch(config-te-rsvp-path-expl-MyPath)# hop 10.0.45.5 after 10.0.34.4
```

When it is no longer necessary to have a hop order use the **`no hop`** command.

Example

Use the **`no hop 10.0.12.2`** command when removing hops.

```
switch(config-te-rsvp-path-expl-MyPath)# no hop 10.0.12.2
```

The **`show active`** command explains what is currently happening with the configuration

Example

The currently active configuration is retrieved with the **`show active`** command.

```
switch(config-te-rsvp-path-expl-MyPath)# show active
router traffic-engineering
   rsvp
      path MyPath explicit
         hop 10.0.23.3
         hop 10.0.34.4
         hop 10.0.45.5
```

The **`show pending`** command explains what happens when exiting the submode.

Example

The pending configuration that becomes active upon exiting the submode is retrieved with the **show pending** command.

```
switch(config-te-rsvp-path-expl-MyPath)# show pending
   hop 10.0.23.3
   hop 10.0.44.4
   hop 10.0.45.5
```

The **show diff** command displays the differences between the **show active** and **show pending** commands.

Example

The difference between these two commands is that they are retrieved with the **show diff** command.

```
switch(config-te-rsvp-path-expl-MyPath)# show diff
 hop 10.0.23.3
-hop 10.0.34.4
+hop 10.0.44.4
 hop 10.0.45.5
switch(config-te-rsvp-path-expl-MyPath)#
```

When you want to keep all your saved changes, use the **exit** command.

Example

The change in the submode only takes effect when the submode is exited normally.

```
switch(config-te-rsvp-path-expl-MyPath)# exit
switch(config-te-rsvp)#
```

when you do not want to save any changes, use the **abort** command.

Example

Changes are discarded when the submode is aborted.

```
switch(config-te-rsvp-path-expl-MyPath)# abort
switch(config-te-rsvp)#
```

Dynamic Path Specifications

In a dynamic path specification, the operator provides constraints with which a Constained Shortest Path First (CSPF) procedure finds a path in the network topology. Effectively, the path specification serves as a template to get instantiated together with other tunnel constraints (like bandwidth requirements). The CSPF result is a list of strict hops which form the ERO.

Example

The submode to configure dynamic paths is entered by specifying the name of the path, in this example MyPath, and use the **dynamic** keyword.

```
switch(config-te-rsvp)# path MyPath dynamic
switch(config-te-rsvp-path-dyn-MyPath)#
```

Exclude hop constraints specifies that CSPF must not choose the selected address on the path. Each excluding hops expresses that neither end of a link in the path may have the specified address. Note that other interfaces on the same node may be used.

Example

This example excludes hop ***10.0.56.6***.

```
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.56.6 exclude
```

Include hop constraints are supported and specify that these hops have to be included in the computed path in a certain order. By default, a hop is strict, meaning that in the computed path it has to appear directly after the previously specified hop. when a hop is loose it allows other hops to be filled by the CSPF procedure. The keywords before and after work as in the submode for explicit paths.

Example

In the following example hop ***10.0.23.3*** must be included in the path.

```
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.23.3
```

In the following example hop ***10.0.45.5*** loose shows the hop does not have to be in any paticular order.

```
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.45.5 loose
```

The following example shows hop ***10.0.12.2*** comes before ***10.0.23.3***.

```
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.12.2 before 10.0.23.3
```

The following example shows hop ***10.0.67.7*** comes loose after ***10.0.45.5***.

```
switch(config-te-rsvp-path-dyn-MyPath)# hop 10.0.67.7 loose after 10.0.45.5
```

Administrative group constraints are specified to restrict CSPF path computation to links that match a set of admin groups. Their IDs are globally significant and the specification contains lists and ranges of admin groups to include or exclude. In particular, every chosen link for a dynamic path must be in all of the admin groups specified in the ***include all*** range, must be in one of the admin groups specified in the ***include any*** range, and must not be in any of those in the ***exclude*** range. The admin group range starts at ***0*** and ends at ***31***. On the wire, the admin-group of ***0*** is translated to ***0x1***, ***1*** is translated to ***0x2***.

Example

In this example include ***1*** and anything inbetween ***2-4***, and exclude ***7*** and ***9***.

```
switch(config-te-rsvp-path-dyn-MyPath)# administrative-group include all 1
 include any 2-4 exclude 7,9
```

Administrative group constraints are specified using a name as an alias mapped to a numerical value. The mapping is configured under the global TE mode.

These names are directly used to configure administrative group constraints in addition to the existing numerical format. The admin group constraints for the dynamic path are configured using the following RSVP LER CLI.

Example

In this example the administartive-group includes all blue and anything between ***2-4*** red, and excludes ***7*** green.

```
switch(config-te-rsvp-path-dyn-MyPath)# administrative-group include all blue
 include any 2-4,red exclude green,7
```

The list of administrative groups is provided as a comma-separated input without spaces.

The submode for dynamic paths has the same commit/abort semantics and show commands as the submode for explicit paths.

Local Interface

Use the **`local-interface Loopback`** command to derive the source IP address for RSVP-TE tunnels. This is a mandatory setting.

Example

```
switch(config-te-rsvp)# local-interface Loopback 0
```

A few settings are taken from the global ***mpls rsvp*** configuration like Fast-Reroute (FRR) mode and soft preemption. The SRLG setting from the global ***mpls rsvp*** configuration is also used to specify if the secondary path of a tunnel is set up by excluding SRLGs of all the links in the primary path as additional constraints, therefore allowing the secondary path to be disjoint from the primary path.

Tunnel Specifications

A tunnel needs a specification of which tunnel to go to.

Example

In this example, MyTunnel is the name of the submode tunnel to use.

```
switch(config-te-rsvp)# tunnel MyTunnel
switch(config-te-rsvp-tunnel-MyTunnel)#
```

Basic Tunnel Configuration

Each tunnel must have a tunnel destination IP.

Example

The this example the selected destination IP is ***10.2.2.2***.

```
switch(config-te-rsvp-tunnel-MyTunnel)# destination ip 10.2.2.2
```

Adding Path Specifications

A tunnel needs to specify along the path in which its LSPs are established. Therefore, a path specification is referenced by its name and configures the **primary** LSP.

Example

The specified name for the path is ***MyPath***.

```
switch(config-te-rsvp-tunnel-MyTunnel)# path MyPath
```

A secondary LSP is specified which provides a fallback in case the primary LSP is not available. The secondary LSP is either established on-demand cold standby once the primary is not available or pre-signaled hot standby regardless. Configuring a secondary path is optional.

Example

To configure the secondary path, in this example, the name ***MyOtherPath*** is selected as a seconday path and is configured to be pre-signaled.

```
switch(config-te-rsvp-tunnel-MyTunnel)# path MyOtherPath secondary pre-signaled
```

Only one secondary path can be configured, per tunnel.

Bandwidth Specification

Use the **bandwidth** command to reserve bandwidth along the path. The bandwidth is explicitly configured.

Example

In this example, a bandwidth of ***10*** mbps is selected as the reserve bandwidth for the ***myTunnel*** tunnel path.

```
switch(config-te-rsvp-tunnel-myTunnel)# bandwidth 10 mbps
```

Available units are bps, kbps, mbps, and gbps. By default, no bandwidth reservation is signaled, such as a bandwidth of***0*** bps.

An alternative to **explicit** bandwidth configuration is to use the **autobandwidth** feature which specifies the minimum and maximum bandwidth to be used for the tunnel and otherwise adjusts bandwidth with an **adjustment period** based on the observed traffic going over the tunnel. The adjustment period is an optional parameter measured in seconds. The default adjustment period is ***0*** . A special value meaning that the algorithm decides when to adjust bandwidth based on statistical measurements (the actual adjustment cannot happen more frequently than the sampling, which occurs every ***30*** seconds).

Example

In this example the bandwidth is set to auto with a minimum of ***1*** mbps and a maximum of ***5*** mbps, and has an adjustment-period of ***60*** seconds.

```
switch(config-te-rsvp-tunnel-myTunnel)# bandwidth auto min 1 mbps max 5 mbps
 adjustment-period 60
```

The adjustment period is specified in seconds.

Tunnel Priorities

Setup and hold priorities from ***0*** to ***7*** are configured for the tunnel, where ***0*** means most preferred, and ***7*** means least preferred.

Example

In this example, the setup priority is configured of ***5*** and the hold priority is configured of ***3***.

```
switch(config-te-rsvp-tunnel-myTunnel)# priority setup 5 hold 3
```

The CLI does not impose restrictions on the priority values. Therefore, hold priority should be more preferred than setup priority to avoid cycles where an LSP is continuously established and immediately preempted.

Enabling The Tunnel

By default, a tunnel is not enabled, so it needs to be explicitly enabled.

Example

Use the **`no shutdown`** command to explicitly enable the tunnel.

```
switch(config-te-rsvp-tunnel-MyTunnel)# no shutdown
```

As for path specifications, the submode provides **`show active`**, **`show pending`** and **`show diff`** commands. It has abort/commit semantics.

When you want to keep all your saved changes, use the **`exit`** command.

Example

The change in the submode only takes effect when the submode is exited normally.

```
switch(config-te-rsvp-tunnel-MyTunnel)# exit
switch(config-te-rsvp)#
```

when you do not want to save any changes, use the **`abort`** command.

Example

Changes are discarded if the submode is aborted.

```
switch(config-te-rsvp-tunnel-MyTunnel)# abort
switch(config-te-rsvp)#
```

19.6 LDP Pseudowire

LDP pseudowire provides support for emulating Ethernet connections over a Multiprotocol Label Switching (MPLS) network, and controlled using the extension of the MPLS Label Distribution Protocol (LDP) specified in RFC4447.

The **patch panel** command allows "patching" a local interface "connector" to an LDP pseudowire "connector", terminating on the local switch. The LDP pseudowire itself is defined under the **pseudowires** configuration mode, under the ***mpls ldp*** configuration mode.

LDP pseudowire also supports locally patching traffic between two interfaces and is configured under the **patch panel** command.

19.6.1 Configuring LDP Pseudowire

19.6.1.1 Local-Remote Patch Configuration

To configure an LDP pseudowire, complete the following task.

1. Enter the pseudowire configuration section under the ***mpls ldp*** configuration section, and specify an LDP pseudowire name using the **pseudowire** command. In this example, the selected LDP pseudowire name is ***ldppw***.

   ```
   switch(config)# mpls ldp
   switch(config-mpls-ldp)# pseudowire
   switch(config-mpls-ldp-pw)# pseudowire ldppw
   ```

2. Once an LDP pseudowire has been configured, configure the neighbor IP address, pseudowire ID, and pseudowire MTU using the **neighbor**, **pseudowire-id**, and **mtu** configuration commands.

   ```
   switch(config-mpls-ldp-pw-ldppw)# neighbor 1.2.3.4
   switch(config-mpls-ldp-pw-ldppw)# pseudowire-id 1
   switch(config-mpls-ldp-pw-ldppw)# mtu 9000
   ```

3. To configure a cross-connection between two connectors, configure a pseudowire patch by specifying a patch name in the **patch panel** configuration section.

   ```
   switch(config)# patch panel
   switch(config-patch)# patch example
   ```

4. After entering a ***patch*** configuration mode, configure a type 4 LDP pseudowire by specifying a VLAN tagged interface connector and an LDP pseudowire connector using the **connector** command.

   ```
   switch(config-patch-example)# connector interface ethernet1 dot1q vlan 100
   switch(config-patch-example)# connector pseudowire ldp ldppw
   ```

5. After entering a ***patch*** configuration mode, configure a type 5 LDP pseudowire by specifying an interface connector and an LDP pseudowire connector using the **connector** command.

   ```
   switch(config-patch-example)# connector interface ethernet1
   switch(config-patch-example)# connector pseudowire ldp ldppw
   ```

Note: IP routing must be enabled and the interface used for the local connector must be routed.

```
switch(config)# ip routing
switch(config)# interface ethernet1
switch(config-if-Et1)# no switchport
```

19.6.1.2 Local-Local Patch Configuration

The patch can also consist of two interface connectors or two VLAN tagged interface connectors to configure local stitching.

```
switch(config-patch)# patch example
switch(config-patch-example)# connector conn1 ethernet1
switch(config-patch-example)# connector conn2 ethernet2
switch(config-patch)# patch example2
switch(config-patch-example2)# connector conn1 ethernet1 dot1q vlan 100
switch(config-patch-example2)# connector conn2 ethernet2 dot1q vlan 200
```

Note: IP routing must be enabled and both interfaces used for the local connectors must be routed.

```
switch(config)# ip routing
switch(config)# interface ethernet1
switch(config-if-Et1)# no switchport
switch(config)# interface ethernet2
switch(config-if-Et2)# no switchport
```

19.6.1.3 Additional Patch Configuration Commands

The **shutdown** configuration command can be used to disable a pseudowire patch.

Example

```
switch(config-patch-example)# shutdown
```

19.6.1.4 Forwarding Behavior For Layer 2 Control Packets

There are differences in forwarding behavior for reserved MACs between the Ethernet Tagged Mode and the Ethernet Raw Mode. Some Layer 2 control packets are terminated and consumed at the Provider Edge, while others are encapsulated and forwarded on the pseudowire. Below is a list of the common Layer 2 control packets, and a summary of their forwarding behavior.

	Ethernet Tagged Mode (type 4)	Ethernet Raw Mode (type 5)
STP	When STP is enabled, STP packets are terminated. Otherwise, they are forwarded.	Forwarded
LLDP	Terminated	Forwarded
LACP	Terminated	Terminated
MACsec	Not supported	Forwarded

In addition, custom Layer 2 Control Packet forwarding behavior for type 5 pseudowire can be configured using the Layer 2 Protocol Forwarding functionality.

19.6.2 LDP Pseudowire Limitations

When a subinterface has been configured with the same VLAN tag as a pseudowire connector on the same parent interface, or vice versa, only the first configured feature will be active.

When a port-based connector is used for a pseudowire or local cross-connect patch, dot1q-tagged packets will have the outer tag incorrectly removed on ingress/encap if any one of several other features are configured on the switch (on any interface) including Layer 2 subinterfaces, Dot1q Tunnel, VLAN Mapping, and VXLAN. The only workaround is to unconfigure these other features.

When a VLAN-based connector (or subinterface with dot1q encapsulation) is used for a pseudowire or local cross-connect patch, tagged packets will have incorrect tags after forwarding if any one of several other features are configured on the switch (on any interface) including Layer 2 subinterfaces, Dot1q Tunnel, VLAN Mapping, and VXLAN. The only workaround is to unconfigure these other features.

19.6.3 LDP Pseudowire Show Commands

Use the **`show patch panel`** command to display the overall status of the configured pseudowire patches.

Example

```
switch# show patch panel
Patch     Connector                              Status
--------- -------------------------------------- ----------
example   1: Ethernet3                           Up
          2: Port-Channel2
example2  1: Ethernet1 802.1Q VLAN 100           Up
          2: LDP neighbor 1.1.1.1 PW ID 1111
example3  1: LDP neighbor 4.4.4.4 PW ID 1040     Up
          2: Ethernet2
```

Use the **`show patch panel detail`** command to display the detailed status of the configured pseudowire patches.

Example

```
switch# show patch panel detail
PW Fault Legend:
   ET-IN - Ethernet receive fault
   ET-OUT - Ethernet transmit fault
   TUN-IN - Tunnel receive fault
   TUN-OUT - Tunnel transmit fault
   NF - Pseudowire not forwarding (other reason)
Patch: example, Status: Up
   Connector 1: Ethernet3
      Status: Up
   Connector 2: Port-Channel2
      Status: Up
Patch: example2, Status: Up
   Connector 1: Ethernet1 802.1Q VLAN 100
      Status: Up
   Connector 2: LDP neighbor 1.1.1.1 PW ID 1111
      Status: Up
      Local MPLS label: 100032, Group ID: 0
         MTU: 9000, 802.1Q VLAN request sent: -
```

```
      Neighbor MPLS label: 900000, Group ID: 0x0
         MTU: 9000, 802.1Q VLAN request received: -
      PW type: 4 (tagged), Control word: N
      Tunnel type: LDP, Tunnel index: 198
Patch: example3, Status: Up
   Connector 1: LDP neighbor 4.4.4.4 PW ID 1040
      Status: Up
      Local MPLS label: 100002, Group ID: 0x0
         MTU: 1500, 802.1Q VLAN request sent: -
      Neighbor MPLS label: 400000, Group ID: 0x0
         MTU: 9213, 802.1Q VLAN request received: -
      PW type: 5 (raw), Control word: N
      Tunnel type: LDP, Tunnel index: 2
   Connector 2: Ethernet2
      Status: Up
```

Use the **`show patch panel forwarding`** command to display the forwarding information for configured pseudowire connectors.

Example

```
switch# show patch panel forwarding
Legend:
   Type - Pseudowire type: 4 (tagged)
                           5 (raw)
   CW - Control word used

In/Out                         Type CW VLAN Status Patch
------------------------------ ---- -- ---- ------ ------------
Et1 802.1Q VLAN 100               4         Up     example2
   Label 900000, LDP Tun 198
Et2                               5         Up     example3
   Label 400000, LDP Tun 2
Et3                                         Up     example
   Po2
Po2                                         Up     example
   Et3

PW Label Out                    Source Type CW Status Patch
-------- ---------------------- ------ ---- -- ------ ----------
100002   Et2                    LDP       5    Up     example3
100032   Et1 802.1Q VLAN 100    LDP       4    Up     example2
```

19.7 LDP Entropy Label

If a network device uses deep packet inspection for load balancing, it is recommended to use entropy label in LDP to improve load balancing in MPLS networks by providing sufficient entropy in the label stack itself.

19.7.1 Configuring LDP Entropy Label

Use the **`entropy-label`** command to enable the MPLS LDP entropy labelling on the switch. Use the **no** form of the command to disable the MPLS LDP entropy labelling. By default, the entropy label knob is disabled.

```
switch(config)# mpls ldp
```

```
switch(config-mpls-ldp)# entropy-label
switch(config-mpls-ldp)# exit
```

19.7.2 Show Commands

The **show running-config all** command displays the Entropy Label Signaling status.

```
switch# show running-config all
!
mpls ldp
   ...
   no entropy-label
   ...
!
```

The **show mpls ldp bindings detail** command displays whether the ELC is advertised or not for a given FEC for both local/remote bindings.

```
switch# show mpls ldp bindings 1.1.1.1/32 detail
1.1.1.1/32
     Local binding:    Label: imp-null, ELC: False
          Uptime:   1:51:14
          Advertised to: 2.2.2.2:0, 3.3.3.3:0
     Remote binding: Peer ID: 2.2.2.2:0, Label: 100001, ELC: True
          Uptime:   1:51:57
     Remote binding: Peer ID: 3.3.3.3:0, Label: 100002, ELC: True
          Uptime:   1:52:05
```

Following debug commands are modified to incorporate Entropy label and Entropy label indicator information in label stack for debugging on ingress router.

```
switch> show platform fap fec <idx>
Tunnel Type: Mpop(mpls pop), Mpush(mpls push), Mswap(mpls swap),
             MoG(mpls-over-gre), T(IPv4 tunnels GRE/VXLAN)
 'CW  - Control word',
 'FL  - Flow label',
 'EL  - Entropy label',
 'ELI - Entropy label indicator',
 'D   - ECMP is divergent across switching chips',
 ' ------------------------------------------------------------------------------',
 '|                                FEC Entry                                     |',
 '|------------------------------------------------------------------------------',
 '|     |      |     |                   |     |       |                  |',
 '| ECMP|  FEC |     |                   |     |       |                  | Tunnel',
 '|Index| Index| Cmd |     Destination   | VID |Outlif |   MAC / CPU Code  |T Value',
 ' ------------------------------------------------------------------------------',
 '|  -  |353896|ROUTE| Et1/1             |0    |20480  | 00:00:00:00:00:01 |Mpush 501 401 301 ELI EL 201'

B. show platform fap ip route
Tunnel Type: M(mpls), G(gre), MoG(mpls-over-gre),
             vxlan-o(vxlan outer-rewrite info), vxlan-i(vxlan inner-rewrite info)
 'CW  - Control word',
 'FL  - Flow label',
 'EL  - Entropy label',
 'ELI - Entropy label indicator',
 '*   - Routes in LEM',
 'D   - ECMP is divergent across switching chips',
 ' ------------------------------------------------------------------------------------------------------------',
 '|                                Routing Table                                                 |             |',
 '|-------------------------------------------------------------------------------------------------------------',
 '|VRF|   Destination    |     |                   |     |       |                  | ECMP|  FEC | Tunnel',
 '| ID|      Subnet      | Cmd |     Destination   | VID |Outlif |   MAC / CPU Code  |Index| Index|T Value',
 ' ------------------------------------------------------------------------------------------------------------',
 '|0  |192.168.1.3/32    |ROUTE| Et1/2             |100  |8188   | 00:00:00:00:00:02 |  -  |353897|   -   ',
 '|0  |200.0.0.0/24      |ROUTE| Et1/2             |100  |20484  | 00:00:00:00:00:02 |  -  |353898|M 501 401
 301 ELI EL 201',
 '|0  |0.0.0.0/0         |TRAP | CoppSystemL3LpmOver|0    |  -    | SlowReceive       |  -  |576815|   -   ',
```

19.7.3 Limitations

- When the Transit LSR is acting as a PHP hop for a given LDP tunnel with entropy label pushed onto it by an Ingress LSR, only the LDP tunnel label will be popped at PHP whenever the Egress LSR has signaled implicit-null in LDP.
- Egress LSR should be capable of handling ELI as top label in the label stack like [ELI, EL, VPN, Payload].

19.8 MPLS Commands

MPLS Commands

- **MPLS Commands**
  - entropy-label
  - mpls ip
  - mpls shared index
  - mpls static
  - mpls static vrf-label
  - mpls tunnel termination
  - mpls tunnel termination (vrf qos map)
  - show mpls route
  - show mpls route summary
  - show mpls tunnel termination qos maps
  - vrf (MPLS tunnel termination)
- **RSVP-TE LSR**
  - authentication
  - cspf delay
  - fast-reroute
  - fast-reroute reversion
  - hello interval
  - ping mpls rsvp session
  - preemption method
  - refresh method
  - shutdown
  - srlg
- **RSVP-TE LSR Show Commands**
  - show mpls rsvp
  - show mpls rsvp counters
  - show mpls rsvp neighbor
  - show mpls rsvp session
  - show mpls rsvp session detail
  - show mpls rsvp session summary
  - show traffic-engineering cspf path
  - show traffic-engineering database

19.8.1 authentication

Use the **`authentication`** command to enable cryptographic authentication. The **`no`** form and the **`default`** form of the command removes crytographic authentication.

Command Mode

MPLS RSVP sub-mode (mpls-rsvp)

Command Syntax

authentication [index ***1-4294967295*** [active | password]|[sequence-number window ***1-255***]|[type [md5 | none]]]

no authentication [index ***1-4294967295*** [active | password]|[sequence-number window ***1-255***]|[type [md5 | none]]]

default authentication [index ***1-4294967295*** [active | password]|[sequence-number window ***1-255***]|[type [md5 | none]]]

Parameters

- **index** ***1-4294967295*** Password index.
  - **active** Use index as the active password.
  - **password** password.
- **sequence-number** Index in the sequesnce.
  - **window** ***1-255*** Reorder window size. The default value is ***5***. A value of ***1*** turns off support for packet reordering.
- **type** Authentication mechanism.
  - **md5** MD5 hash.
  - **none** No authentication mechanism. Disables cryptographic authentication.

Examples

- Cryptographic Authentication (***RFC 2747***) is enabled by setting the **authentication type** to **md5** and configuring an active password.

```
(config-mpls-rsvp)# authentication type md5
```

- Authentication secrets are configured with an index. One of the indices should be chosen as the actively used password:

```
(config-mpls-rsvp)# authentication index 1 password s3cr3t
(config-mpls-rsvp)# authentication index 1 active
```

  The **active** password is used to authenticate outgoing messages. All configured passwords are accepted for authentication of incoming packets, which allows smooth key rollover.
- Password obfuscation is available:

```
(config-mpls-rsvp)# authentication index 1 password 7 07092E43
```

- The size of the sequence number reorder window can be changed to accommodate a larger number of out-of-order packets. A value of *N* means that a packet is accepted if all earlier received packets with a higher sequence number are within the preceding *N-1* packets.

```
(config-mpls-rsvp)# authentication sequence-number window 5
```

  The default value is ***5***. A value of ***1*** effectively turns off support for packet reordering.

19.8.2 cspf delay

RSVP-TE uses CSPF to compute the FRR backup path. CSPF can be throttled to avoid frequent path changes when there are frequent network events. Use the **cspf delay** command to specify how frequently CSPF is to run after a network event by specifying the initial wait interval, back-off interval, and maximum wait interval for CSPF. The **no** form and the **default** form of the command removes the CSPF delay.

Command Mode

Router traffic engineering mode

Command Syntax

`cspf delay` [initial *n* back-off ***n***] max ***n***

`no cspf delay` [initial *n* back-off ***n***] max ***n***

`default cspf delay` [initial *n* back-off ***n***] max ***n***

Parameters

- **initial** Specify initial wait interval for CSPF.
  - ***n*** Time in milliseconds. ***1-300000***.
    - **back-off** specify back-off interval for CSPF.
      - ***n*** Time in milliseconds. ***1-300000***.
- **max** Specify maximum wait interval for CSPF.
  - ***n*** Time in milliseconds. ***1-300000***.

19.8.3 entropy-label

The **`entropy-label`** command enables the MPLS LDP Entropy Label.

The **`no entropy-label`** command removes the MPLS LDP Entropy Label configurations from the ***running-config***.

Command Mode

MPLS LDP Configuration Mode

Command Syntax

`entropy-label`

`no entropy-label`

Example

This command enables the MPLS LDP Entropy Label.

```
switch(config)# mpls ldp
switch(config-mpls-ldp)# entropy-label
switch(config-mpls-ldp)# exit
```

19.8.4 fast-reroute

Use the **`fast-reroute`** command to support Fast Reroute (FRR) link protection/NHOP (***RFC 4090***). You can enable FRR protection by setting the Fast Reroute mode to **link-protection**.

Command Mode

MPLS RSVP sub-mode (mpls-rsvp)

Command Syntax

`fast-reroute` [mode [link-protection | node-protection | none]|[reversion [global | local]]

Parameters

- **mode** Fast reroute mode.
  - **link-protection** Protects against failure of the next link.
  - **node-protection** protects against failure of the next node.

- **none** Disables fast reroute.
- **reversion** Select reversion behavior.
  - **global** Global revertive repair (default).
  - **local** Local revertive repair.

Examples

- Support for Fast ReRoute (FRR) link protection/NHOP (***RFC 4090***) is enabled by setting the Fast Reroute mode to **link-protection**.

```
(config-mpls-rsvp)# fast-reroute mode link-protection
```

- To turn off FRR, change the mode to the default setting **none**.

```
(config-mpls-rsvp)# fast-reroute mode none
```

- You can change the revertive behavior of the FRR from the **global** revertive mode to the **local** revertive mode. In the **global** revertive mode, an LSR that is re-routed over a bypass tunnel because its downstream link is d keeps using the bypass tunnel even after the link has recovered. This expects the headend router to set up a new LSP upon notification that a link is not available anymore. In the **local** revertive mode, the LSR switches back to using the primary link after recovery

```
(config-mpls-rsvp)# fast-reroute reversion local
```

- The default for reversion is **global**.

```
(config-mpls-rsvp)# fast-reroute reversion global
```

19.8.5 fast-reroute reversion

You can change the revertive behavior of the FRR from the **global** revertive mode to the **local** revertive mode. In the **global** revertive mode, an LSR that is re-routed over a bypass tunnel because its downstream link is dead keeps using the bypass tunnel even after the link has recovered. This expects the headend router to set up a new LSP upon notification that a link is not available anymore. In the **local** revertive mode, the LSR switches back to using the primary link after recovery.

The default for reversion is **global**.

Command Mode

Configuration sub-mode for RSVP

Command syntax

fast-reroute reversion [global | local]

Parameters

- **global** Global revertive repair.
- **local** Local revertive repair.

Examples

- Setting the fast-reroute reversion to local:

```
(config-mpls-rsvp)# fast-reroute reversion local
```

- The default for reversion is **global**.

```
(config-mpls-rsvp)# fast-reroute reversion global
```

19.8.6 hello interval

The **hello interval** command sets the time between the hello packets. The **no** form of the command explicitly disables the Hello messages, while the **default** form of the command resets to the default setting of 10 seconds with a multiplier of 4.

Command Mode

Configuration sub-mode for RSVP.

Command Syntax

hello interval [interval [***sec***]] [multiplier [***num***]]

no hello interval [interval [***sec***]][multiplier [***num***]]

default hello interval [interval [***sec***]][multiplier [***num***]]

Parameters

- **interval *sec*** The interval in units of seconds.
  - **multiplier *num*** The number of missed hellos after which the neighbor is expired.

Examples

- In this example, hello messages are sent to all known neighbors every ***10*** seconds. If no hello responses are received from a neighbor for 4*10=40 seconds, communication is considered to be lost and the neighbor is reset.

```
(config-mpls-rsvp)# hello interval 10 multiplier 4
```

- In this example, the default of ***10*** seconds with multiplier ***4*** is reset.

```
(config-mpls-rsvp)# default hello interval
```

- in this example, the command to explicitly disables the Hello messages.

```
(config-mpls-rsvp)# no hello interval
```

19.8.7 mpls ip

The **mpls ip** command enables MPLS routing. Multiprotocol Label Switching (MPLS) is a networking process that avoids complex lookups in a routing table by replacing complete network addresses with short path labels for directing data packets to network nodes. MPLS data paths are serviced through a tunnel encapsulation data structure that adds four-byte label headers to packets.

The **no mpls ip** and **default mpls ip** commands disable MPLS routing by removing the **mpls ip** command from ***running-config***. When MPLS routing is disabled, routed MPLS packets are dropped and all MPLS routes and adjacencies are removed. MPLS routing is disabled by default.

Command Mode

Global Configuration

Command Syntax

mpls ip

no mpls ip

default mpls ip

Examples

- This command enables MPLS routing. Previous commands enabled IP routing and configured MPLS static routes.

```
switch(config)# mpls ip
switch(config)# show running-config

! Command: show running-config

!
ip routing
!
mpls ip
!
mpls static top-label 3400 10.14.4.4 pop payload-type ipv4
mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
!

!
end
switch(config)#
```

- This command disables MPLS routing.

```
switch(config)# no mpls ip
switch(config)# show running-config


! Command: show running-config
                 <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip routing
!
mpls static top-label 3400 10.14.4.4 pop payload-type ipv4
mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
!

!
end
switch(config)#
```

19.8.8 mpls shared index

The **`mpls shared index`** command configures an Ethernet-Segment (ES) shared-index to allocate an ESI label value based on the shared-index configuration.

The **`no mpls shared index`** command removes mpls shared index configuration from the ***running-config***.

Command Mode

EVPN Ethernet-Segment Mode

Command Syntax

`mpls shared index` ***index-value***

`no mpls shared index` ***index-value***

Parameter

- ***index-value*** Label index value. Value ranges from ***1*** to ***1024***.

Example

- These commands place the switch on Ethernet-Segment (ES) configuration mode and configure an MPLS shared index value of ***100***.

```
switch(config)# interface Ethernet4
switch(config-if-Et1)# switchport access vlan 1000
switch(config-if-Et1)# evpn ethernet-segment
switch(config-evpn-es)# identifier 0022:2222:2222:2222:2222
switch(config-evpn-es)# mpls shared index 100
switch(config-evpn-es)# route-target import 00:02:00:02:00:02
```

19.8.9 mpls static

The **mpls static** command creates an MPLS rule that specifies the method of handling of inbound MPLS traffic. Multiprotocol Label Switching (MPLS) is a networking process that replaces complete network addresses with short path labels for directing data packets to network nodes.

Static rules specify these parameters:

- **MPLS filter:** The top-label parameter specifies the 20-bit value that the MPLS packet's top header label must match to be handled by the rule.
- **Nexthop location:** Specifies the destination nexthop address (IPv4 or IPv6) and the interface through which the switch forwards the packet.
- **MPLS action:** Specifies the MPLS label stack management action performed on the packet:
  - **pop-payload:** removes the top label from stack; this terminates an LSP (label-switched path).
  - **swap-label:** replaces top label with a specified new label; this passes a packet along an LSP.
- **Rule priority:** Specifies the rule to be used when an MPLS packet matches multiple rules.

The **no mpls static** and **default mpls static** commands delete the specified MPLS rule from ***running-config***.

- Commands that include only a top label tag remove all MPLS rules with the matching top label.
- Commands with no **PRIORITY** parameter remove all matching routes of every metric value.

Command Mode

Global Configuration

Command Syntax

mpls static top-label ***top_tag***[**bgp peer** [***peer IP***]] [**DEST_INTF**] **NEXTHOP_ADDR ACTION** [**PRIORITY**]

no mpls static top-label ***top_tag***

no mpls static top-label ***top_tag*** [**DEST_INTF**] **NEXTHOP_ADDR ACTION** [**PRIORITY**]

default mpls static top-label ***top_tag***

default mpls static top-label ***top_tag*** [**DEST_INTF**] **NEXTHOP_ADDR ACTION** [**PRIORITY**]

Parameters

- ***top_tag*** Top header's label field contents. Value ranges from ***0*** to ***1048575*** (20 bits).
- BGP peer ***peer IP*** The BGP peer identifier.
- **DEST_INTF** Specifies interface through which ***NEXTHOP_ADDR*** is accessed. Options include:
  - ***no parameter*** Any interface.
  - **ethernet *e_num*** Ethernet interface specified by ***e_num***.
  - **loopback *l_num*** Loopback interface specified by ***l_num***.
  - **management *m_num*** Management interface specified by ***m_num***.
  - **port-channel *p_num*** Port-channel interface specified by ***p_num***.
  - **vlan *v_num*** VLAN interface specified by ***v_num***.

    - **vxlan** ***vx_num*** VXLAN interface specified by ***vx_num***.
- ***NEXTHOP_ADDR*** Nexthop address for MPLS for filtered MPLS packets. Options include:
    - ***ipv4_addr*** IPv4 address.
    - ***ipv6_addr*** IPv6 address.
- **ACTION** MPLS header stack management action performed on packet. Options include:
    - **pop payload-type ipv4** Removes top layer from stack. Payload is handled as IPv4 packet.
    - **pop payload-type ipv6** Removes top layer from stack. Payload is handled as IPv6 packet.
    - **swap-label** ***0*** to ***1048575*** Replaces header label with specified label value (20 bits).
- **PRIORITY** Specifies rule priority when multiple rules match a packet. Options include:
    - ***no parameter*** Assigns a metric value of 100 to the rule.
    - **metric** ***1*** to ***255*** Lower values denote higher priority. Value ranges from ***1*** to ***255***.

The **`mpls static`** command does not support push label actions.

Examples

- These commands create an MPLS rule that matches packets with a top label value of ***3400*** and causes the removal of the top label from the header stack. The nexthop destination of the IPv4 payload is IP address ***10.14.4.4*** through Ethernet interface ***3/3/3***. This rule has a metric value of ***100***.

```
switch(config)# mpls static top-label 3400 ethernet 3/3/3 10.14.4.4 pop
 payload-type ipv4
switch(config)# show running-config

!
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
!

end
switch(config)#
```

- These commands create a backup rule that forwards the packet through Ethernet interface ***4/3***. This rule's metric value of ***150*** assigns it backup status prior to the first rule.

```
switch(config)# mpls static top-label 3400 ethernet 4/3 10.14.4.4 pop
 payload-type ipv4 metric 150
switch(config)# show running-config

!
mpls static top-label 3400 Ethernet4/3 10.14.4.4 pop payload-type ipv4
 metric 150
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
!

                    <-------OUTPUT OMITTED FROM EXAMPLE-------->
end
switch(config)#
```

- These commands create an MPLS rule that forwards the packet to the nexthop address through any interface.

```
switch(config)# mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
switch(config)# show running-config

                     <-------OUTPUT OMITTED FROM EXAMPLE-------->

!
mpls static top-label 3400 Ethernet4/3 10.14.4.4 pop payload-type ipv4
 metric 150
```

```
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
!

end
switch(config)#
```

- This command uses the BGP peer option to designate the nexthop..

```
switch(config)#mpls static top-label 1001 100.1.1.1 bgp peer pop payload-
type ipv4
```

- When the ***peer IP*** is not specified, the next-hop is considered as the peer.

```
switch(config)# mpls static top-label 1002 100.1.1.2 bgp peer 100.1.1.1 pop
 payload-type ipv4
switch(config)# mpls static top-label 1002 100.1.1.3 bgp peer 100.1.1.1 pop
 payload-type ipv4
```

- Multiple nexthops can be associated with the same BGP peer by explicitly specifying the BGP peer IP. This is useful in the case of third-party nexthops.

```
switch(config)# mpls static top-label 1000 nexthop-group mygroup bgp peer
 100.1.1.1 pop payload-type ipv4
```

19.8.10 mpls static vrf-label

Use the **mpls static vrf-label** command to configure a static VRF label route. The **no** and **default** versions of the command removes the configuration.

Command Mode

Global configuration mode

Command Syntax

mpls static vrf-label ***mpls-label*** [vrf ***vrf-name***]

no mpls static vrf-label ***mpls-label*** [vrf ***vrf-name***]

default mpls static vrf-label ***mpls-label*** [vrf ***vrf-name***]

Parameters

- ***mpls-label*** Value of the MPLS label.
- **vrf** ***vrf-name*** VRF instance name.

Example

```
switch(config)# mpls static vrf-label 100 vrf default
switch(config)# mpls static vrf-label 200 vrf v1
```

19.8.11 mpls tunnel termination

Command Mode

Global configuration mode

Command Syntax

mpls tunnel termination [model [php model][ttl [[pipe | uniform] dscp]| pipe | uniform]

no mpls tunnel termination [model [php model][ttl [[pipe | uniform] dscp]| pipe | uniform]

default mpls tunnel termination [model [php model][ttl [[pipe | uniform] dscp]| pipe | uniform]

Parameters

- **model** Tunnel termination TTL and DSCP model.
- **php** Penultimate hop popping configuration.
- **ttl** Model for TTL.
- **pipe** Preserve the inner value.
- **uniform** Propagate value from outer header.
- **dscp** mode for DSCP.

Example

```
switch(config)# mpls tunnel termination model ttl uniform dscp pipe
```

19.8.12 mpls tunnel termination (vrf qos map)

The **mpls tunnel termination (vrf qos map)** command changes configuration mode to allow the attachment of DSCP-to-traffic-class maps to specific VRFs. This is a group change command, so no changes are made to the running config until the **exit** command is issued.

MPLS tunnel termination mode has one subcommand, **vrf**. **vrf** is also a group change command, with the subcommand **qos map dscp to traffic-class**.

Command Mode

Global Configuration

Command Syntax

mpls tunnel termination

Subcommands

vrf

Example

These commands enter MPLS Tunnel Termination Configuration mode and attach the DSCP-to-traffic-class map ***map1*** to the VRFs ***newVRF1*** and ***newVRF2***.

```
switch(config)#mpls tunnel termination
switch(config-mpls-tunnel-termination)#vrf newVRF1
switch(config-mpls-tunnel-termination-vrf-newVRF1)#qos map dscp to traffic-
class map1
switch(config-mpls-tunnel-termination-vrf-newVRF1)#exit
switch(config-mpls-tunnel-termination)#vrf newVRF2
switch(config-mpls-tunnel-termination-vrf-newVRF2)#qos map dscp to traffic-
class map1
switch(config-mpls-tunnel-termination-vrf-newVRF2)#exit
switch(config-mpls-tunnel-termination)#exit
switch(config)#exit
```

19.8.13 ping mpls rsvp session

LSP Ping allows the user to check if the remote endpoint of an RSVP session is reachable through the LSP. Running ping on an RSVP LSP creates an LSP Ping Request packet with the label programmed by RSVP and that packet will follow the MPLS path until it reaches the end of the tunnel. If the node that receives this Request is the intended destination, it replies with an RSVP Ping Reply through normal IP routing. When the source receives the Reply it indicates that there are no apparent data plane failures and that the endpoint of that LSP is reachable.

Command Mode

Configuration sub-mode for RSVP

Command Syntax

`ping mpls rsvp session` [id ***num*** | name ***word***] [lsp | pad-reply | repeat | source | standard | tos | tos]

Parameters

- **id** Specifies the session by ID
  - ***num*** RSVP session ID.
    - **lsp** Specifies LSP
    - **pad-reply** Indicates that the reply should copy the pad TLV.
    - **repeat** Specifies repeat count.
    - **source** Specifies source address.
    - **standard** Sets the standard to comply with.
    - **tos** Specifies ToS value.
    - **tos** Specifies MPLS traffic class field.
- **name** Specifies session by name.
  - ***word*** RSVP session name.
    - **lsp** Specifies LSP
    - **pad-reply** Indicates that the reply should copy the pad TLV.
    - **repeat** Specifies repeat count.
    - **source** Specifies source address.
    - **standard** Sets the standard to comply with.
    - **tos** Specifies ToS value.
    - **traffic-class** Specifies MPLS traffic class field.

Examples

- A user can invoke the ping utility for a specific LSP by using the session and LSP IDs displayed in the CLI show commands.

```
switch# ping mpls rsvp session id 1 lsp 1 repeat 3
LSP ping to RSVP session #1 LSP #1
   timeout is 5000ms, interval is 1000ms
Via 10.0.12.2, Ethernet1, label 100000
   Reply from 10.0.34.4: seq=1, time=53.294ms, success: egress ok
Via 10.0.12.2, Ethernet1, label 100000
   Reply from 10.0.34.4: seq=2, time=75.329ms, success: egress ok
Via 10.0.12.2, Ethernet1, label 100000
   Reply from 10.0.34.4: seq=3, time=85.574ms, success: egress ok

--- RSVP target fec 0.4.4.4 : lspping statistics ---
Via 10.0.12.2, Ethernet1, label 100000
   3 packets transmitted, 3 received, 0% packet loss, time 2272ms
   3 received from 10.0.34.4, rtt min/max/avg 53.294/85.574/71.399 ms
```

- For the **`ping mpls rsvp session`** command, the argument **lsp** is optional. If unspecified, the utility pings all LSPs within that session.

```
switch# ping mpls rsvp session id 1 repeat 2
LSP ping to RSVP session #1
   timeout is 5000ms, interval is 1000ms
LSP 1
Via 10.0.12.2, Ethernet1, label 100000
   Reply from 10.0.34.4: seq=1, time=60.28ms, success: egress ok
LSP 2
```

```
Via 10.0.12.2, Ethernet1, label 100002
   Reply from 10.0.34.4: seq=1, time=81.701ms, success: egress ok
LSP 1
Via 10.0.12.2, Ethernet1, label 100000
   Reply from 10.0.34.4: seq=2, time=52.807ms, success: egress ok
LSP 2
Via 10.0.12.2, Ethernet1, label 100002
   Reply from 10.0.34.4: seq=2, time=62.814ms, success: egress ok

--- RSVP target fec 0.4.4.4 : lspping statistics ---
LSP 1
Via 10.0.12.2, Ethernet1, label 100000
   2 packets transmitted, 2 received, 0% packet loss, time 1262ms
   2 received from 10.0.34.4, rtt min/max/avg 52.807/60.280/56.544 ms

LSP 2
Via 10.0.12.2, Ethernet1, label 100002
   2 packets transmitted, 2 received, 0% packet loss, time 1262ms
   2 received from 10.0.34.4, rtt min/max/avg 62.814/81.701/72.257 ms
```

- Similarly, the session can be specified by name and the utility pings all the LSPs within that session.

```
switch# ping mpls rsvp session name Session1to4 repeat 2
LSP ping to session Session1to4
   timeout is 5000ms, interval is 1000ms
LSP 1
Via 10.0.12.2, Ethernet1, label 100000
   Reply from 10.0.34.4: seq=1, time=63.314ms, success: egress ok
LSP 2
Via 10.0.12.2, Ethernet1, label 100002
   Reply from 10.0.34.4: seq=1, time=75.639ms, success: egress ok
LSP 1
Via 10.0.12.2, Ethernet1, label 100000
   Reply from 10.0.34.4: seq=2, time=60.875ms, success: egress ok
LSP 2
Via 10.0.12.2, Ethernet1, label 100002
   Reply from 10.0.34.4: seq=2, time=77.135ms, success: egress ok

--- RSVP target fec 0.4.4.4 : lspping statistics ---
LSP 1
Via 10.0.12.2, Ethernet1, label 100000
   2 packets transmitted, 2 received, 0% packet loss, time 1262ms
   2 received from 10.0.34.4, rtt min/max/avg 60.875/63.314/62.094 ms

LSP 2
Via 10.0.12.2, Ethernet1, label 100002
   2 packets transmitted, 2 received, 0% packet loss, time 1262ms
   2 received from 10.0.34.4, rtt min/max/avg 75.639/77.135/76.387 ms
```

19.8.14 preemption method

The **`preemption method`** command enables deferred failure of RSVP-TE LSPs on link oversubscription. Use a preemption timer value to configure a delay on a transit router to support LSPs signaled with soft preemption enabled by the headend (***RFC 5712***).

The default preemption method is soft preemption with a timer value of 30 seconds.

Command Mode

Configuration sub-mode for RSVP

Command Syntax

`preemption method` [hard | soft] timer ***t***

no preemption method [hard | soft] timer ***t***

default preemption method [hard | soft] timer ***t***

Parameters

- **hard** Hard preemption.
- **soft** Soft preemption.
  - **timer** Time limit for LSP teardown.
    - ***t 1-65535*** Timer value in units of seconds.

Examples

- In this example, the preemption method is soft preemption with a timer value of ***10*** seconds.

```
(config-mpls-rsvp)# preemption method soft timer 10
```

- Setting the preemption method to hard preemption results in a timer value of ***0*** seconds and disables the feature.

```
(config-mpls-rsvp)# preemption method hard
```

19.8.15 refresh method

Using the **refresh method** command with the **bundled** keyword enables the Refresh Overhead Reduction that supports the sending of message IDs and refreshing state with refresh messages.

Command Mode

Configuration sub-mode for RSVP

Command Syntax

refresh method [bundled | explicit]

Parameters

- **bundled** Refresh states using message identifiers lists. This is the default setting.
- **explicit** Send each message individually.

Examples

- Use the **bundled** keyword to enable the Refresh Overhead Reduction.

```
(config-mpls-rsvp)# refresh method bundled
```

- To turn off refresh overhead reduction, use the **explicit** keyword.

```
(config-mpls-rsvp)# refresh method explicit
```

19.8.16 show mpls route summary

The **show mpls route summary** command displays statistics about the configuration and implementation of MPLS rules.

Command Mode

EXEC

Command Syntax

show mpls route summary

Example

This command displays a summary of MPLS rule implementation.

```
switch> show mpls route summary
Number of Labels: 1 (1 unprogrammed)
Number of adjacencies in hardware: 0
Number of backup adjacencies: 2 switch>
```

19.8.17 show mpls route

The **show mpls config route** command displays the switch's MPLS static rule configuration for the specified routes and rules.

Command Mode

EXEC

Command Syntax

show mpls [**INFO_LEVEL**] **route** [***header_label***]

Parameters

- **INFO_LEVEL** Specifies the filters that are used to select the routes to display. Options include:
  - ***no parameter*** Displays routes published by the forwarding agent.
  - **config** Displays all configured routes.
  - **lfib** Displays routes stored to the Label Forwarding Information Base (LFIB).
- ***header_label*** Filters routes by MPLS top header label. Options include:
  - ***no parameter*** Displays routes for all header values.
- ***0*** to ***1048575*** Specifies header for which command displays information.

Examples

- This command displays the MPLS rule configuration.

```
switch# show mpls config route
Codes: S - Static MPLS Route, IA - IS-IS SR Adjacency Segment,
       IP - IS-IS SR Prefix Segment, L - LDP,
       I>L - IS-IS SR Segment to LDP, L>I - LDP to IS-IS SR Segment, R -
 RSVP

In-Label Out-Label Metric Payload NextHop   Egress-ACL Status Monitored
-------- --------- ------ ------- --------- ---------- ------ --------------
1000     pop       100    ipv4    100.0.0.1 apply      up
1001     pop       100    ipv4    20.0.0.2  apply      down
1002     pop       100    ipv4    100.0.0.2 apply      down   100.0.0.2(Bgp)
1003     pop       100    ipv4    20.0.0.2  apply      down   100.0.0.2(Bgp)
1004     pop       100    ipv4    20.0.0.2  apply      up     20.0.0.2(Bgp)
1005     pop       100    ipv4    30.0.0.3  apply      up     200.0.0.3(Bgp)
```

  The status could be down if either of the following is true

  1. Nexthop is not resolved (**show ip route** ***nexthop*** - shows no route).
  2. The monitored BGP session is down (**show ip bgp summary** - shows the peer is not in established state).
- The following example shows that the first route is down because the BGP peer session is down.

```
switch# show mpls config route
Codes: S - Static MPLS Route, IA - IS-IS SR Adjacency Segment,
       IP - IS-IS SR Prefix Segment, L - LDP,
       I>L - IS-IS SR Segment to LDP, L>I - LDP to IS-IS SR Segment, R -
 RSVP
```

```
In-Label Out-Label Metric Payload NextHop  Egress-ACL Status Monitored
-------- --------- ------ ------- -------- ---------- ------ -------------
88886    pop       100    ipv4    14.0.0.4 apply      down   14.0.0.4(BGP)
88887    pop       100    ipv4    15.0.0.5 apply      down
```

- The following example shows the status of label ***88886*** is down because the BGP peer session to ***14.0.0.4*** is down, but route exists for ***14.0.0.4***.

```
switch# show ip bgp summary
BGP summary information for VRF default
Router identifier 3.0.1.3, local AS number 3000
Neighbor Status Codes: m - Under maintenance
  Neighbor V AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State
 PfxRcd PfxAcc
  14.0.0.4 4 4000             16        18    0    0 01:04:42 Active

switch# show ip route 14.0.0.4

VRF: default
Codes: C - connected, S - static, K - kernel,
 O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - Other BGP Routes,
       B I - iBGP, B E - eBGP, R - RIP, I L1 - IS-IS level 1,
       I L2 - IS-IS level 2, O3 - OSPFv3, A B - BGP Aggregate,
       A O - OSPF Summary, NG - Nexthop Group Static Route,
       V - VXLAN Control Service, M - Martian,
       DH - DHCP client installed default route,
       DP - Dynamic Policy Route, L - VRF Leaked,
       G  - gRIBI, RC - Route Cache Route

 S        14.0.0.4/32 [1/0] via 2.0.1.4, Ethernet2
```

19.8.18 show mpls rsvp

Use the **show mpls rsvp** to display the overall state of the RSVP.

Command Mode

EXEC

Command Syntax

show mpls rsvp [bandwidth | counters | ip | ipv6 | neighbor | session]

Parameters

- **bandwidth** Displays RSVP bandwidth information.
- **counters** Displays RSVP message counters.
- **ip** Displays details related to IPv4.
- **ipv6** Displays details related to IPv6.
- **neighbor** Displays RSVP neighbors.
- **session** Displays RSVP session information.

Example

```
switch> show mpls rsvp
Administrative state: enabled
Operational state: up
Refresh interval: 30 seconds
Refresh reduction: enabled
Hello messages: enabled
```

```
   Hello interval: 10 seconds
   Hello multiplier: 4
Fast Re-Route: disabled
   Mode: none
   Hierarchical FECs: enabled
Cryptographic authentication: disabled
MTU signaling: disabled
Number of sessions: 1
   Ingress/Transit/Egress: 0/1/0
Number of LSPs: 1
   Operational: 1
   Ingress/Transit/Egress: 0/1/0
   Currently using bypass tunnels: 0
Number of bypass tunnels: 0
Number of neighbors: 2
Number of interfaces: 2
```

19.8.19 show mpls rsvp counters

Use the **show mpls rsvp counters** command to display RSVP message counters, per interface.

Command Mode

EXEC

Command Syntax

show mpls rsvp counters [[interface [Ethernet |Fabric | Loopback | Management | Port-Channel | Switch | Tunnel | Vlan | Vxlan]]| ipv4 ***interface*** | ipv6 ***interface***

Parameters

- **interface** Filter by interface.
  - **Ethernet** Hardware Ethernet interface.
  - **Fabric** Fabric interfaces.
  - **Loopback** Hardware interface used for looping packets.
  - **Management** Management interface.
  - **Port-Channel** Lag interface.
  - **Switch** Switch interface.
  - **Tunnel** Tunnel interface.
  - **Vlan** Logical interface into a VLAN.
  - **Vxlan** VXLAN tunnel interface.
- **ipv4** Filter by IPv4.
  - ***interface*** Filter by interface.
- **ipv6** Filter by IPv6.
  - ***interface*** Filter by interface.

Example

```
switch> show mpls rsvp counters
Received Messages:

Interface  Path  PathTear  PathErr  Resv  ResvTear  ResvErr  Srefresh  Other  Errors
---------  ----  --------  -------  ----  --------  -------  --------  -----  ------
Ethernet1  5     0         0        0     0         0        8         51     1
Ethernet2  0     0         0        14    0         0        0         0      0

Sent Messages:

Interface  Path  PathTear  PathErr  Resv  ResvTear  ResvErr  Srefresh  Other  Errors
---------  ----  --------  -------  ----  --------  -------  --------  -----  ------
Ethernet1  0     0         0        4     0         0        9         49     0
```

```
Ethernet2  13    0         0        0     0         0        0         4      0
```

19.8.20 show mpls rsvp neighbor

Use the **show mpls rsvp neighbor** command to display a summary of all RSVP neighbors, or the list of active LSPs for a specific neighbor.

Command Mode

EXEC

Command Syntax

show mpls rsvp neighbor [***A.B.C.D*** | ***A:B:C:D:E:F:G:H*** | summary]

Parameters

- ***A.B.C.D*** IP (v4 or v6) address of neighbor.
- ***A:B:C:D:E:F:G:H*** IP (v4 or v6) address of neighbor.
- **summary** Displays summarized information.

Examples

- Use the **show mpls rsvp neighbor** command to display a summary of all RSVP neighbors.

```
switch> show mpls rsvp neighbor
Neighbor 10.0.1.1
   Upstream for
      Session #1 LSP #1
   Downstream for
   Neighbor uptime: 00:01:24
   Authentication type: disabled
   Last hello received: 1 seconds ago
   Last hello sent: 1 seconds ago
   Bypass tunnel: not requested
Neighbor 10.0.2.2
   Upstream for
   Downstream for
   Session #1 LSP #1
   Neighbor uptime: 00:01:24
   Authentication type: disabled
   Last hello received: -
   Last hello sent: 31 seconds ago
   Bypass tunnel: not requested
```

- Use the **show mpls rsvp neighbor summary** command to display summarized information.

```
switch> show mpls rsvp neighbor summary
Neighbor             Role       Sessions LSPs
==================== ========== ======== ========
10.0.1.1             Upstream   1     1
10.0.2.2             Downstream 1     1
```

19.8.21 show mpls rsvp session detail

Use the **show mpls rsvp session detail** to display the detailed information of RSVP sessions.

Command Mode

EXEC

Command Syntax

show mpls rsvp session detail

Example

```
switch> show mpls rsvp session detail
Session #1
   Destination address: 0.4.4.4
   Tunnel ID: 0
   Extended Tunnel ID: 0.1.1.1
   Role: transit
   LSP #1
        State: up
        [...]
        MTU Signaling: enabled
           Received Path MTU: 1800 bytes
           Sent Path MTU: 1500 bytes
  [...]
```

19.8.22 show mpls rsvp session summary

Use the **show mpls rsvp session summary** command to filter sessions in the show command and in LSP ping and traceroute commands. Sessions can further be filtered by name, destination, router role (transit/ingress/egress), and state.

Command Mode

EXEC

Command Syntax

show mpls rsvp session summary

Example

```
switch> show mpls rsvp session summary
Session  Destination        LSP      Name         Role    Bypass State
======== ================== ======== ============ ======= ====== =========
1        0.4.4.4             1        Session1    transit n/req  up
```

19.8.23 show mpls rsvp session

Use the **show mpls rsvp session** command to list the current RSVP sessions filtered by IP address.

Command Mode

EXEC

Command Syntax

show mpls rsvp session

Example

```
switch> show mpls rsvp session
Session #1
   Destination address: 0.4.4.4
   Tunnel ID: 0
   Extended Tunnel ID: 0.1.1.1
   Role: transit
   LSP #1
        State: up
        Type: primary
        Source address: 0.1.1.1
        LSP ID: 1
        LSP uptime: 00:02:38
```

```
        Session name: Session1
        Local label: 100000
        Downstream label: 100000
        Upstream neighbor: 10.0.1.1
           Last refresh received: 17 seconds ago
           Last refresh sent: 10 seconds ago
        Downstream neighbor: 10.0.2.2
           Last refresh received: 7 seconds ago
           Last refresh sent: 9 seconds ago
        Bypass tunnel: not requested
```

19.8.24 show mpls tunnel termination qos maps

The **show mpls tunnel termination qos maps** command shows the DSCP to TC map associated with each VRF, or with a specified VRF.

Command Mode

Privileged EXEC mode

Command Syntax

show mpls tunnel termination qos maps [**vrf** ***vrf_name***]

Parameters

- ***vrf_name*** The VRF whose DSCP to TC map association is to be shown. If this parameter is omitted, all DSCP to TC maps associated with a VRF are shown.

Examples

This command shows the DSCP to TC maps associated with VRFs.

```
switch#show mpls tunnel termination qos maps
VRF newVRF1 DSCP to TC map: map1
VRF newVRF2 DSCP to TC map: map1 switch#
```

This command shows the DSCP to TC map associated with VRF ***newVRF2***.

```
switch#show mpls tunnel termination qos maps vrf newVRF2
VRF newVRF2 DSCP to TC map: map1 switch#
```

19.8.25 show traffic-engineering cspf path

Use the **show traffic-engineering cspf path** command to display all the paths conuted by CSPF.

Command Mode

EXEC

Command Syntax

show traffic-engineering cspf path [***destination-IP***]detail]

Parameters

- ***A.B.C.D*** CSPF path to this destination IP address.
- **detail** Show detailed path information.

Examples

- In this example, the CSPF destination path ***20.0.0.1*** is selected to display.

```
switch> show traffic-engineering cspf path 20.0.0.1
```

```
Destination     Constraint                              Path
20.0.0.1        exclude Ethernet1                       0.1.1.1
                exclude SRLG of Ethernet1               0.1.1.2
                                                        0.2.2.2
                                                        3.3.3.2
                exclude Ethernet2                       0.1.1.1
                                                        0.1.1.2
                                                        0.2.2.2
                                                        3.3.3.2
```

- In this example, the CSPF destination path ***20.0.0.1*** is selected to display detailed information.

```
switch> show traffic-engineering cspf path 20.0.0.1 detail

Destination: 20.0.0.1
   Path Constraint: exclude Ethernet1
                    exclude SRLG of Ethernet1: orange-link (500),
                       green-link (400), 100, red-link (200), 600
   Request Sequence number: 1
   Response Sequence number: 1
   Number of times path updated: 2
   Last updated: 00:01:58
   Reoptimize: Always
   Path:
   0.1.1.1
   0.1.1.2
   0.2.2.2
   3.3.3.2

   Path Constraint: exclude Ethernet2
   Request Sequence number: 2
   Response Sequence number: 2
   Number of times path updated: 3
   Last updated: 00:00:38
   Reoptimize: Always
   Path:
   0.1.1.1
   0.1.1.2
   0.2.2.2
   3.3.3.2
```

19.8.26 show traffic-engineering database

Use the **`show traffic-engineering database`** to display the topology used for CSPF computations. Starting from ***EOS Release 4.23.1F***, the SRLG group details of a neighbor are shown if it is advertised.

Command Mode

EXEC

Command Syntax

`show traffic-engineering database`

Special Considerations

- Beginning with ***EOS Release 4.23.1F***, the SRLG group details of a neighbor are shown if it is advertised.
- Beginning with ***EOS Release 4.24.2F***, information for the OSPFv2 topology is displayed if configured.

Example

```
switch# show traffic-engineering database
```

```
TE Router-ID: 1.0.0.2
  Source: IS-IS Level-1 IPv4 Topology Database
    IS-IS System-ID: 1111.1111.1001
      Number of Links: 2
        Network type: P2P
          Neighbor: 1111.1111.1002
   Administrative group (Color): 0x123a
            TE Metric: 30
            IPv4 Interface Addresses:
              20.20.20.1
              192.168.20.1
            IPv4 Neighbor Addresses:
              20.20.20.2
              192.168.20.2
            Maximum link BW: 25.00 Gbps
            Maximum reservable link BW: 10.00 Mbps
            Unreserved BW:
              TE class 0: 9.00 Mbps     TE class 1: 9.00 Mbps
              TE class 2: 8.5.00 Mbps   TE class 3: 8.00 Mbps
              TE class 4: 7.00 Mbps     TE class 5: 7.50 Mbps
              TE class 6: 6.00 Mbps     TE class 7: 6.00 Mbps


        Network Type: LAN
          Neighbor: 1111.1111.1003.02
            TE Metric: 30
         Administrative Group: 0x12
            IPv4 Local Addresses:
              30.30.30.1

            Maximum Link BW: 10.00 Gbps

            Maximum Reservable Link BW: 10.50 Gbps

            Unreserved BW:

              TE-Class 0: 8.50 Gbps       TE-Class 1: 8.70 Gbps

              TE-Class 2: 7.50 Gbps       TE-Class 3: 7.25 Gbps

              TE-Class 4: 6.50 Gbps       TE-Class 5: 7.30 Gbps

              TE-Class 6: 3.50 Gbps       TE-Class 7: 7.20 Gbps
  Source: IS-IS Level-2 IPv4 Topology Database
    IS-IS System-ID: 1111.1111.1003
      Number of Links: 1
        Network Type: LAN
         Neighbor: 1111.1111.1003.16
           IPv4 Local Addresses:
             40.40.40.1
            Maximum link BW: 10.00 Gbps
            Maximum reservable link BW: 5.00 Gbps
            Unreserved BW:
              TE class 0: 4.00 Gbps     TE class 1: 4.00 Gbps
              TE class 2: 4.00 Gbps     TE class 3: 4.00 Gbps
              TE class 4: 3.00 Gbps     TE class 5: 3.00 Gbps
              TE class 6: 3.00 Gbps     TE class 7: 3.00 Gbps
  Source: OSPFv2 Instance ID 33 Area-ID 0.0.0.0 Topology Database
    OSPFv2 Router-ID: 1.2.3.4
      Number of Links: 2
        Network type: P2P
          Neighbor: 3.4.5.6
            Administrative group (Color): 0x123a
            TE metric: 30
```

```
            IPv4 Interface Addresses:
              20.20.20.1
              192.168.20.1
            IPv4 Neighbor Addresses:
              20.20.20.2
              192.168.20.1
            Maximum link BW: 25.00 Gbps
            Maximum reservable link BW: 10.00 Mbps
            Unreserved BW:
              TE class 0: 9.00 Mbps    TE class 1: 9.00 Mbps
              TE class 2: 8.50 Mbps    TE class 3: 8.00 Mbps
              TE class 4: 7.00 Mbps    TE class 5: 7.50 Mbps
              TE class 6: 6.00 Mbps    TE class 7: 6.00 Mbps
        Network type: LAN
          Neighbor: 2.3.4.5
            Administrative group (Color): 0x12
            TE metric: 30
            IPv4 Interface Addresses:
              30.30.30.1
            IPv4 Neighbor Addresses:
              0.0.0.0
            Maximum link BW: 10.00 Gbps
            Maximum reservable link BW: 10.5 Gbps
            Unreserved BW:
              TE class 0: 8.50 Gbps    TE class 1: 8.70 Gbps
              TE class 2: 7.50 Gbps    TE class 3: 7.25 Gbps
              TE class 4: 6.50 Gbps    TE class 5: 7.30 Gbps
              TE class 6: 3.50 Gbps    TE class 7: 7.20 Gbps


TE Router-ID: 1.0.0.3
  Source: IS-IS Level-1 IPv4 Topology Database
    IS-IS System-ID: 1111.1111.1004
      Number of Links: 2
        Network type: P2P
          Neighbor: 1111.1111.1002
          IPv4 Interface Addresses:
              1.0.5.1
            IPv4 Neighbor Addresses:
              1.0.5.2
            Maximum link BW: 50.00 Gbps
            Maximum reservable link BW: 10.00 Gbps
            Unreserved BW:
              TE class 0: 8.00 Gbps     TE class 1: 8.00 Gbps
              TE class 2: 8.00 Gbps     TE class 3: 8.00 Gbps
              TE class 4: 7.00 Gbps     TE class 5: 7.00 Gbps
              TE class 6: 7.00 Gbps     TE class 7: 7.00 Gbps
            Shared Risk Link Group:
              Group: 100
              Group: green-link (150)
```

19.8.27 shutdown

The **shutdown** command disables the RSVP-TE protocol instance or the RSVP-related functions for the interface. The RSVP-TE configuration information associated with the interface is retained.

RSVP-TE is enabled globally by issuing **no shutdown** in the configuration sub-mode. This is the only mandatory setting for RSVP-TE to work. There is no per-interface knob to enable or disable RSVP. However, RSVP is only enabled on interfaces on which MPLS is enabled.

Command Mode

Configuration sub-mode for RSVP

Command Syntax

shutdown

no shutdown

Examples

- You can enable RSVP-TE globally by issuing the **no shutdown** command.

```
(config-mpls-rsvp)# no shutdown
```

- Similarly, it is disabled with the **shutdown** command, which is the default.

```
(config-mpls-rsvp)# shutdown
```

19.8.28 srlg

The **srlg** command specifies if link SRLGs of a primary LSP are to be considered as constraints while creating a fast-reroute bypass tunnel with either link or node protection. When the **srlg** command is specified with **strict** keyword, then when a path for a bypass tunnel excluding SRLGs of next-hop interface of primary LSP can not be found, RSVP does not setup the bypass tunnel. When the **srlg** command is specified without the **strict** keyword, then a bypass tunnel is setup with as many links as possible that exclude the SRLGs of next-hop interface of primary LSP and where such links are not available, links that have the least number of SRLGs which are to be excluded are used.

When this CLI is not configured, the behavior remains the same as before, which is to turn off SRLG processing. Therefore, the **no** and **default** versions of the command takes you back to the default of SRLG processing being turned off.

Command Mode

Configuration sub-mode for RSVP

Command Syntax

srlg strict

no srlg strict

default srlg strict

Parameters

strict Applies strict SRLG constraints.

Example

The SRLGs of an interface can be configured using the following traffic-engineering CLIs.

```
switch(config)# interface Et1
switch(config-if-Et1)# traffic-engineering srlg 100

switch(config)# interface Et2
switch(config-if-Et2)# traffic-engineering srlg 200
```

19.8.29 vrf (MPLS tunnel termination)

The **vrf (MPLS tunnel termination)** command places the switch in MPLS Tunnel Termination VRF Configuration mode. There is one command available in this mode: **qos map dscp (MPLS tunnel termination VRF)**.

Command Mode

MPLS Tunnel Termination Configuration mode

Command Syntax

`vrf` ***vrf_name***

Parameter

- ***vrf_name*** The name of the VRF to configure. This does not create a VRF. The VRF must be created with the **`vrf`** command in Configuration mode.

Example

These commands place the switch in MPLS Tunnel Termination VRF Configuration mode for VRF ***newVRF1***.

```
switch(config)#mpls tunnel configuration
switch(config-mpls-tunnel-configuration)#vrf newVRF1
switch(config-mpls-tunnel-configuration-vrf-newVRF1)#
```

Chapter 20

Visibility and Monitoring Services

The Visability and Monitors Services chapter contains the following sections:

- Test Access Point Aggregation
- Latency Analyzer (LANZ)
- Sampled Flow Tracking
- sFlow
- SNMP
- VM Tracer
- MapReduce Tracer
- Transceiver Performance Monitoring
- AVA Sensor

20.1 Test Access Point Aggregation

This section describes Test Access Point (TAP) aggregation and the data structures that it requires. Topics in this section include:

- TAP Aggregation Introduction
- TAP Aggregation Description
- TAP Aggregation Extra MPLS Pop (4 to 6 Labels)
- TAP Aggregation Configuration
- TAP Aggregation Traffic Steering
- TAP Aggregation GUI
- TAP Aggregation Keyframe and Timestamp Configuration
- TapAgg GRE Tunnel Termination
- Tap Aggregation Hardware Forwarding Profile
- TAP Aggregation MPLS Pop
- TAP Aggregation 802.1br EVN Tag Stripping
- TAP Aggregation Commands

20.1.1 TAP Aggregation Introduction

Ethernet-based switches are commonly deployed in dedicated networks to support Test Access Point (TAP) and mirror port traffic toward one or more analysis applications. Ports configured to mirror data can simultaneously switch traffic to its primary destination while directing a copy of that traffic to analysis or test devices. TAP ports are typically part of a dedicated environment that allows for the aggregation of data streams from multiple sources that can be directed to multiple destinations.

Arista switches support port mirroring and TAP aggregation and the data structures required by these functions.

20.1.2 TAP Aggregation Description

These sections describe TAP aggregation, timestamps, and keyframes:

- TAP Aggregation
- Timestamps and Keyframes

20.1.2.1 TAP Aggregation

Test Access Point (TAP) aggregation is the accumulation of data streams and the subsequent dispersal of these streams to devices and applications that analyze, test, verify, parse, detect, or store data. TAP aggregation requires an environment free from switching operations. Arista switches operate in one of two device modes:

- **Switching mode**: the switch performs normal switching and routing operations. Data mirroring is supported in switching mode. Tap aggregation is not available in switching mode.
- **TAP aggregation mode**: The switch is a data-monitoring device and does not provide normal switching and routing services. Data mirroring is not available in tap aggregation mode.

  Access control lists, port channels, LAGs, QoS, and VLANs function normally in both modes.

Ethernet and port channel interfaces are configured as **TAP** and **tool** ports to support tap aggregation.

- **TAP ports**: a tap port is an interface that receives a data stream that two network ports exchange.

  TAP ports prohibit egress traffic. MAC learning is disabled. All control plane interaction is prevented. Traps for inbound traffic are disabled. Tap ports are in STP forwarding mode.
- **Tool ports**: A tool port is an interface that replicates data streams received by one or more tap ports. Tool ports connect to devices that process the monitored data streams.

  Tool ports prohibit ingress traffic. MAC learning is disabled. All control plane interaction is prevented. Tool ports are in STP forwarding mode.

TAP and tool ports are configured with the **`switchport mode`** command. These ports are active when the switch is in tap aggregation mode and error-disabled when the switch is in switching mode.

TAP and tool ports are designated through switchport mode commands and act similar to trunk ports, in that they can allow access to VLANs specified through allowed-VLAN lists. Tap ports also specify a native VLAN for handling untagged frames.

Access, trunk, and dot1q-tunnel mode ports are active when the switch is in switching mode and error-disabled when the switch is in tap aggregation mode.

TAP and tool mode ports are active when the switch is in TAP aggregation mode and error-disabled when the switch is in switching mode.

TAP aggregation groups are data structures that map a set of TAP ports to a set of tool ports. Both TAP and tool ports may belong to multiple TAP aggregation groups, and a TAP aggregation group may contain multiple TAP and tool ports.

20.1.2.2 Timestamps and Keyframes

FM6000 platform switches support packet timestamping of packets sent from any port at line rate. Timestamps are used to correlate network events and in performance analysis. Keyframes provide information to assist in the interpretation of timestamps.

The switch contains two 64-bit counters to maintain ASIC time and UTC time. ASIC time is based on an internal 350 MHz counter. UTC is absolute time that is maintained by a precision oscillator and synchronized through PTP.

Timestamps are derived from the least significant 31 bits of ASIC time. Based on the 350 MHz counter period and 31-bit resolution, timestamp values repeat every 6.135 seconds.

Keyframes are periodically inserted into the data stream to provide context for interpreting timestamps. Keyframes contain the 64-bit value of the ASIC time counter, the corresponding 64-bit value of the UTC time counter, and the elapsed time since the last PTP synchronization of the UTC counter. Inserting one keyframe every second into the data stream assures that the timestamp value in each egress packet can be associated with values of the complete 64-bit ASIC time counter and the corresponding UTC counter.

20.1.2.2.1 Timestamps

Timestamps are based on a frame's ingress time and applied to frames sent on egress ports, ensuring that timestamps on monitored traffic reflect ingress timing of the original frames. Timestamping is configured on the egress port where the timestamp is applied to the frame.

A timestamp consists of the least significant 31 bits of the ASIC time counter. The most significant bit of the least significant byte is a 0 pad, resulting in a 32-bit timestamp with 31 bits of data. The keyframe mechanism provides recovery of the most significant 33 bits of the ASIC counters and a map to UTC time. Applications use this mechanism to determine the absolute time of the frame timestamp.

The switch supports three timestamp modes, which are configurable on individual Ethernet ports. The modes differ in the management of the egress frame's 32-bit frame check sequence (FCS):

- **Disabled**: timestamping is disabled.
- **FCS Replacement Mode**: the original FCS is discarded, and the ingress timestamp is appended to frame data, followed by a new FCS that is based on the appended timestamp. The result is a valid Ethernet frame, but the headers of all nested protocols are not updated to reflect the timestamp.
- **FCS Appending Mode**: the original FCS is discarded and replaced by the ingress timestamp. The size of the original frame is maintained without any latency impact, but the FCS is not valid.

20.1.2.2.2 Keyframes

Keyframes contain routable IP packets that provide information to relate timestamps with the complete ASIC counter and absolute UTC time. Keyframes have valid L2 and L3 headers. Keyframes contain these header fields:

- MAC fields (12 bytes):
  - Source MAC address is the address of the egress interface transmitting the keyframe.
  - Destination MAC address is configured through a CLI command.
- IP Header (20 bytes):
  - Source IP address is configured through CLI; default is management interface IP address.
  - Destination IP address is configured through a CLI command.
  - TTL is set to ***64***.
  - TOS is set to ***0***.
  - Protocol field is set to ***253***.
  - IP header's ID field is set to ***0***.

Keyframes contain these payload fields:

- **ASIC time**: (64 bits) ASIC time counter. (***2.857*** ns resolution).
- **UTC time**:(64 bits) Unix time that corresponds to ASIC time (ns).
- **Last sync time**: (64 bits) ASIC time of most recent PTP synchronization.
- **Keyframe time**: (64 bits) ASIC time of the keyframe's egress (ns).
- **Egress interface drops**: (64 bits) Number of dropped frames on keyframe's egress interface.
- **Device ID**: (16 bits) device ID (user defined).
- **Egress interface**: (16 bits) Keyframe's egress switchport.
- **FCS type (8 bits)**: Timestamping mode configured on keyframe's egress port.
  - ***0***: timestamping disabled.
  - ***1***: timestamp is appended to payload; new FCS is added to the frame.
  - ***2***: timestamp overwrites the existing FCS.
- **Reserved (8 bits)**: reserved for future use.
- **Skew numerator/skew denominator**: form a ratio indicating the ASIC clock skew. If the ratio is greater than ***1***, the clock is skewed fast; if the ratio is less than ***1***, the clock is skewed slow.

Last sync time equals ***0*** when there was no previous synchronization or the time since the last synchronization is greater than ***8*** hours.

The 31-bit frame timestamp provides high-resolution timing, rolling over about every ***6.135*** seconds (31 bits at 2.857ns per tick). To obtain the full ASIC time and to correlate the timestamp to an absolute UTC time, the switch sends keyframes. Each keyframe contains the current ASIC time and UTC time; hence an application can compute the high order bits of the ASIC time (for precise, relative timing) from the ASIC to UTC time mapping, and then determine absolute time.

ASIC to UTC time conversion is not quite immediate, so the UTC time in the frame will not be the current time. A keyframe timestamp is provided for this purpose. The frame also includes the timestamping mode (FCS type) so applications can dynamically determine the timestamp's byte offset. Each field is shown in the following table.

Table 72: Keyframe Payload

0 7	8 15	16 31
ASIC time		
UTC time		
Last sync time		
Skew numerator		
Skew denominator		
Keyframe timestamp		
Drop count		
Device ID		Egress interface
FCS type	Reserved	

20.1.3 TAP Aggregation Extra MPLS Pop (4 to 6 Labels)

Available starting with ***EOS Release 4.23.1F*** extra MPLS pop for TAP Aggregation allows you to remove four to six MPLS labels from a packet. Previously, only one to three labels were able to be popped.

When configured, popping four to six MPLS labels will work in all the cases where popping one to three MPLS labels works. MPLS pop for ***1*** to ***3*** labels can be found here: https://www.arista.com/en/support/toi/eos-4-15-0f/13620-tap-aggregation-mpls-pop.

20.1.3.1 TAP Aggregation Extra MPLS Pop Configuration

MPLS pop is configurable per tap port as follows to pop one to three labels:

```
(config-if-Et1/1)# switchport tap mpls pop all
```

However, MPLS pop for four to six labels requires a user defined TAP Aggregation profile based on off of tap-aggregation-extended. This profile must have one action and three fields configured under the tapagg port feature, such as the following:

```
(config)# hardware tcam
```

```
(config-hw-tcam)# profile foo copy tap-aggregation-extended
(config-profile-foo)# feature tapagg port
(config-profile-foo-feature-tapagg-port)# action set-fwd-header
(config-profile-foo-feature-tapagg-port)# key field mpls-label1-l
ower-24b mpls-label2-lower-24b mpls-label3-lower-24b
(config-profile-foo-feature-tapagg-port)# tap aggregation
Saving new profile 'foo'
(config-tap-agg)# mode exclusive profile foo
```

With the above profile changes, four to six MLPS label pop is additionally supported when the MPLS pop feature is configured on a tap port.

To check if a profile is installed correctly, use the **show hardware tcam profile** command.

```
(config-tap-agg)# show hardware tcam profile
```

Upon successful installation of a TAP Aggregation profile, you should see a display similar to the following:

```
Configuration            Status
FixedSystem          foo*                    foo
* configuration overridden by TapAgg
```

20.1.3.1.1 Configuring a Local LFIB Convergence Delay for Protected Node or Adjacency Segments

The Point of Local Repair (PLR) switches to the TI-LFA backup path on link failure or BFD neighbor failure but switches back to the post-convergence path once the PLR computes SPF and updates its LFIB. This sequence of events can lead to micro-loops in the topology if the PLR converges faster than other routers along the post-convergence path. So a configuration option is provided to apply a delay, after which the LFIB route being protected by the TI-LFA loop-free repair path will be replaced by the post-convergence LFIB route.

To configure a convergence delay only to LFIB routes that are being protected, the following command is used either in the router IS-IS mode or the router IS-IS address-family sub-mode. A default of 10 seconds is used when using the command without an explicitly specified delay.

```
switch(config-router-isis-af)# timers local-convergence-delay
 [<delay_in_milliseconds>] protected-prefixes
```

20.1.3.2 Limitations

20.1.3.2.1 IP Steering

IP steering does not work for MPLS packets with four or more labels. Currently, you must work around this limitation with a configuration similar to the **Configuration for IP Steering**. This workaround does not work when MPLS pop is not enabled.

20.1.3.2.2 Configuration for IP Steering

When MPLS pop is enabled, you can enable IP steering for for to six MPLS labels by adding an extra port into the configuration as a tap-tool port.

For example, you want to perform IP steering from interface ***Ethernet1*** to a set of interfaces and also want MPLS labels to be popped. Add a new interface ***Ethernet2***, which will not receive any external traffic, as a tap-tool port with traffic loopback. Then configure ***Ethernet1*** to forward to ***Ethernet2*** as its default interface. The commands would look similar to the following:

```
(config-if-Et2)# swi mode tap-tool
(config-if-Et2)# traffic-loopback source system device phy
(config-if-Et2)# int et1
(config-if-Et1)# swi tap default interface et2
```

All of ***Ethernet1***'s old tool ports should be disabled and become ***Ethernet2***'s tool ports. Then, all IP steering configuration that was originally meant to be between ***ET1*** and its old tool ports should be configured with ***Ethernet2*** in place of ***Ethernet1***. Finally, configure MPLS pop on ***Ethernet1***.

20.1.3.3 TAP Aggregation Extra MPLS POP (4 to 6 Labels) Commands

show hardware tcam profile

20.1.4 TAP Aggregation Configuration

These sections describe TAP aggregation configuration tasks:

- Enabling Tap Aggregation Mode
- Tap Aggregation Mixed Mode
- Tap Port Configuration
- Tool Port Configuration
- Per-linecard TCAM Profile Configuration
- Two-Way Ports for Tap Aggregation
- Tap Aggregation QoS Handling on Tap Ports
- Identity VLAN Tagging
- Tap Aggregation Group Configuration

20.1.4.1 Enabling Tap Aggregation Mode

The switch supports switching mode and TAP aggregation mode. In switching mode, normal switching and routing functions are supported while TAP aggregation functions are disabled. In TAP aggregation mode, TAP aggregation functions are enabled while normal switching and routing functions are disabled. By default, the switch is in switching mode.

A ports switchport status depends on its switchport mode and the switch's TAP aggregation mode.

- **Tap aggregation mode enabled**: TAP and tool ports are enabled. Switching ports are errdisabled.
- **Tap aggregation mode disabled**: TAP and tool ports are errdisabled. Switching ports are enabled.

To enable the switch to carry out TAP aggregation, first enter TAP aggregation configuration mode using the **`tap aggregation`** command, then set the mode to **exclusive**.

Note: The switch can also perform TAP aggregation in **mixed** mode. See Mixed Mode Configuration.

Example

These commands enter TAP aggregation configuration mode, then place the switch in TAP aggregation **exclusive** mode.

```
switch(config)# tap aggregation
switch(config-tap-agg)# mode exclusive
switch(config-tap-agg)# show active
tap aggregation
   mode exclusive
switch(config-tap-agg)#
```

To return the switch to switching mode, remove the **mode** command from ***running-config***.

Examples

- These commands enter TAP aggregation configuration mode, then place the switch in switching mode.

```
switch(config)# tap aggregation
switch(config-tap-agg)# no mode
switch(config-tap-agg)# show active
switch(config-tap-agg)#
```

- These commands enter switching mode and remove all TAP aggregation configuration mode statements.

```
switch(config)# no tap aggregation
switch(config)#
```

20.1.4.2 TAP Aggregation Mixed Mode

On a modular switch, the user can configure TAP Aggregation on some linecards and leave other linecards to operate normally. This is referred to as TAP aggregation mixed mode.

- Mixed Mode Platform Compatibility
- Mixed Mode Configuration

20.1.4.2.1 Mixed Mode Platform Compatibility

The following platforms support TAP Aggregation Mixed Mode.

- DCS-7500R
- DCS-7500R2

20.1.4.2.2 Mixed Mode Configuration

Complete the following steps to configure ***Linecard 3*** as a TAP aggregation linecard in mixed mode.

1. Enable the switch for configuration.

```
switch> configure terminal
```

2. Enable TAP aggregation.

```
switch(config)# tap aggregation
```

3. Enable TAP aggregation mixed mode, selecting the targeted linecard module using the TAP aggregation default.

```
switch(config-tap-agg)# mode mixed module linecard 3 tap-aggregation-default
```

Note: Changing modes may affect available functionality. Unsupported configuration elements will be ignored.

The profile selection in mixed mode is the same as in exclusive mode. The user can configure multiple linecards for TAP aggregation in mixed mode.

The user can check TAP Aggregation Mixed Mode status by executing the following show commands:

```
switch(config)# show running-config section tap
tap aggregation
   mode mixed module linecard 3 profile tap-aggregation-default
switch(config)#show hardware tcam profile
                    Configuration        Status
Linecard4           default              default
Linecard3           default              tap-aggregation-default
Linecard6           default              default
switch(config)#
```

20.1.4.3 TAP Port Configuration

TAP ports function when the switch is in TAP aggregation mode. TAP ports receive traffic for replication to one or more tool ports. In TAP aggregation mode, TAP ports are in STP forwarding state and prohibit egress traffic. MAC learning, control plane interaction and traps for inbound traffic are disabled.

TAP mode ports are configured through switchport mode commands. TAP mode command settings persist in ***running-config*** without taking effect when the switch is not in TAP aggregation mode or the interface is not in TAP aggregation mode.

This section describes the following tap port configuration steps.

- Configuring an Interface as a TAP Mode Port
- TAP Port Allowed VLAN List Configuration
- TAP Port Native VLAN
- TAP Port Packet Truncation

Configuring an Interface as a Tap Mode Port

Ethernet and port-channel interfaces are configured as TAP ports with the **`switchport mode`** command.

Example

These commands configure interface ethernet ***41*** through ***43*** as TAP mode ports.

```
switch(config)# interface ethernet 41-43
switch(config-if-Et41-43)# switchport mode tap
switch(config-if-Et41-43)# show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation
 Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------
------------
Et41      tap            tap            1        1    0          ---
Et42      tap            tap            1        1    0          ---
Et43      tap            tap            1        1    0          ---
```

```
switch(config-if-Et41-43)#
```

TAP Port Allowed VLAN List Configuration

By default, TAP mode interfaces handle tagged traffic for all VLANs. The **switchport tap allowed vlan** command creates or modifies the set of VLANs for which a TAP port handles tagged traffic.

Example

These commands create TAP-mode allowed VLAN lists for interface ethernet ***41*** through ***43***.

```
switch(config)# interface ethernet 41
switch(config-if-Et41)# switchport tap allowed vlan 401-410
switch(config-if-Et41)# interface ethernet 42
switch(config-if-Et42)# switchport tap allowed vlan 411-420
switch(config-if-Et41)# interface ethernet 41-42
switch(config-if-Et41-42)# show active
interface Ethernet41
   switchport mode tap
   switchport tap allowed vlan 401-410
interface Ethernet42
   switchport mode tap
   switchport tap allowed vlan 411-420
switch(config-if-Et41-42)#
```

TAP Port Native VLAN

Tap mode Interfaces associate untagged frames with the tap mode native VLAN. The **switchport tap native vlan** command specifies the TAP-mode native VLAN for the configuration-mode interface. The default TAP-mode native VLAN for all interfaces is ***vlan 41***.

Example

These commands assign ***vlan 400*** as the TAP-mode native VLAN for ***interface ethernet 41***.

```
switch(config)# interface ethernet 41
switch(config-if-Et41)# switchport tap native vlan 400
switch(config-if-Et41)# show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation
 Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------
------------
Et41      tap            tap            400      1    0          ---
Et42      tap            tap            1        1    0          ---
Et43      tap            tap            1        1    0          ---
switch(config-if-Et41)#
```

TAP Port Packet Truncation

TAP ports can be configured to truncate inbound packets. The **switchport tap truncation** command configures the configuration-mode interface, as a TAP port, to truncate inbound packets to the specified packet size. By default, TAP ports do not truncate packets.

Examples

- These commands configure ***interface ethernet 41*** to truncate packets to ***150*** bytes.

```
switch(config)# interface ethernet 41
switch(config-if-Et41)# switchport tap truncation 150
switch(config-if-Et41)# show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation
 Default
          Mode                          Vlan     Vlan
 Group
-------------------------------------------------------------
------------
Et41      tap            tap            400      1    150        ---
Et42      tap            tap            1        1    0          ---
Et43      tap            tap            1        1    0          ---
switch(config-if-Et41)#
```

- These commands configure ***interface ethernet 41*** to send complete packets for replication.

```
switch(config-if-Et41)# no switchport tap truncation
switch(config-if-Et41)# show interface ethernet 41 tap
Port      Configured     Status         Native   Id   Truncation
 Default
          Mode                          Vlan     Vlan
 Group
-------------------------------------------------------------
------------
Et41      tap            tap            400      1    0          ---
switch(config-if-Et41)#
```

20.1.4.4 Tool Port Configuration

Tool ports replicate traffic received by TAP ports. Tool ports are mapped to the TAP ports through TAP aggregation groups. A tool port may belong to multiple aggregation groups and an aggregation group may contain multiple tool ports.

Tool ports function when the switch is in TAP aggregation mode. In this switch mode, tool ports are in STP forwarding state and ingress traffic is prohibited. MAC learning, control plane interaction, and traps for inbound traffic are disabled. All control plane interaction is prevented and L2 agents do not send PDUs to tool-mode interfaces. When the switch is in switching mode, tool ports are error-disabled.

Tool-mode ports are configured through switchport commands. Tool-mode command settings persist in ***running-config*** without taking effect when the switch is not in TAP aggregation mode or the interface is not in TAP aggregation mode.

This section describes the following tool port configuration steps.

- Configuring an Interface as a Tool-mode Port
- Tool Port Allowed VLAN List Configuration
- Tool Port Packet Truncation

Configuring an Interface as a Tool-mode Port

Ethernet and port channel interfaces are configured as tool ports with the **`switchport mode`** command.

Example

These commands configure port-channel interfaces ***101*** through ***103*** as tool-mode ports and display the result.

```
switch(config)# interface port-channel 101-103
switch(config-if-Po101-103)# switchport mode tool
switch(config-if-Po101-103)# show interface port-channel 101-103 tool
Port      Configured     Status         Allowed             Id
 Timestamp
          Mode                          Vlans               Tag  Mode
------------------------------------------------------------
-----------
Po101     tool           tool           All                 Off  ---
Po102     tool           tool           All                 Off  ---
Po103     tool           tool           All                 Off  ---
switch(config-if-Po101-103)#
```

Tool Port Allowed VLAN List Configuration

By default, tool mode interfaces handle tagged traffic for all VLANs. The **switchport tool allowed vlan** command creates or modifies the set of VLANs for which a tool port handles tagged traffic.

Example

These commands create tool mode allowed VLAN lists for port-channel interfaces 101 through 103.

```
switch(config)# interface port-channel 101-103
switch(config-if-Po101-103)# switchport tool allowed vlan 1010-1020
switch(config-if-Po101-103)# interface port-channel 101
switch(config-if-Po101)# switchport tool allowed vlan add 1001-1009
switch(config-if-Po103)# interface port-channel 102
switch(config-if-Po102)# switchport tool allowed vlan remove 1016-1020
switch(config-if-Po102)# interface port-channel 103
switch(config-if-Po103)# switchport tool allowed vlan add 1021-1030
switch(config-if-Po103)# show interface port-channel 101-103 tool
Port      Configured     Status         Allowed             Id
 Timestamp
          Mode                          Vlans               Tag  Mode
------------------------------------------------------------
-----------
Po101     tool           tool           1001-1020           Off  ---
Po102     tool           tool           1010-1015           Off  ---
Po103     tool           tool           1010-1030           Off  ---
switch(config-if-Po103)#
```

Tool Port Packet Truncation

Tool ports can be configured to truncate outbound packets. The **switchport tool truncation** command configures the configuration-mode interface, as a tool port, to truncate outbound packets to ***160*** bytes. By default, tool ports do not truncate packets.

Tool port packet truncation is supported only on the 7150 series platform.

Examples

- These commands configure ***interface ethernet 41***, as a tool port, to truncate packets on egress to ***160*** bytes.

```
switch(config)# interface ethernet 41
switch(config-if-Et41)# switchport mode tool
switch(config-if-Et41)# switchport tool truncation 160
switch(config-if-Et41)#
```

- These commands configure ***interface ethernet 41*** to send complete packets.

```
switch(config-if-Et41)# no switchport tool truncation
switch(config-if-Et41)#
```

20.1.4.5 Per-linecard TCAM Profile Configuration

This feature gives the ability to specify different profiles for different linecards in mixed mode.

To enable the TAP aggregation mode and configure a TCAM profile for a linecard set, complete the following steps:

1. Enable the switch for configuration.

```
switch> configure terminal
```

2. Enable TAP aggregation mode.

```
switch(config)# tap aggregation
```

3. Configure the TCAM profile for a linecard set.

```
switch(config-tap-agg)# mode mixed module linecard 3,4 profile tap-aggregati
on-default
switch(config-tap-agg)# mode mixed module linecard 5,6 profile tap-aggregati
on-extended
switch(config-tap-agg)#
```

To disable TAP aggregation on a linecard set, complete the following steps:

1. Enable the switch for configuration.

```
switch> configure terminal
```

2. Enable TAP aggregation mode.

```
switch(config)# tap aggregation
```

3. Disable TAP aggregation for a linecard set.

```
switch(config-tap-agg)# no mode mixed module linecard 3,4
switch(config-tap-agg)#
```

Note: If a TAP is a port-channel, its members must all come from linecards using the same profile.

20.1.4.6 Two-Way Ports for TAP Aggregation

While in TAP aggregation mode, there is support for traffic only in one direction through either TAP ports that receive packets from mirroring, or through optical TAP or tool ports that send out packets to customer devices. Two-way ports for TAP aggregation allow bidirectional transmit and receive capability on a single

port in TAP aggregation mode. Using the TAP-tool switchport mode enables both TAP and tool configurations simultaneously on an interface.

- Two-Way Ports Platform Compatibility
- Two-Way Ports Configuration

20.1.4.6.1 Two-Way Ports Platform Compatibility

The following platforms support two-way ports for TAP aggregation.

- DCS-7280R
- DCS-7280R2
- DCS-7500R
- DCS-7500R2

20.1.4.6.2 Two-Way Ports Configuration

To enable a two-way port, use the **tap-tool** option of the **`switchport mode`** command.

Example

The following commands configure ***interface ethernet 4/1*** as a two-way port, allowing it to function as both a TAP and a tool port.

```
switch(config)# interface ethernet 4/1
switch(config-if-Et4/1)# switchport mode tap-tool
switch(config-if-Et4/1)#
```

Additional configurations for TAP and tool functionality on the interface remain the same. Once the user enables the TAP-tool switchport mode on the interface, they can use the existing TAP and tool mode commands to enable their respective configurations.

Arista recommends using this feature with unidirectional send-receive enabled on the interface, which allows the receiver and transmitter for the interface to operate independently. If one goes down, the other remains active. To enable unidirectional send-receive on an interface, use the **`unidirectional send-receive`** command.

Example

These commands enable unidirectional send-receive on ***interface ethernet 4/1***.

```
switch(config)# interface ethernet 4/1
switch(config-if-Et4/1)# unidirectional send-receive
switch(config-if-Et4/1)#
```

20.1.4.7 TAP Aggregation QoS Handling on TAP Ports

Before EOS 4.20.5F, QoS behavior was not enforced for TAP aggregation ports, meaning that QoS behavior for packets passing through the device was not changed.

- QoS Handling Platform Compatibility
- QoS Handling Configuration
- Displaying QoS Handling Status

20.1.4.7.1 QoS Handling Platform Compatibility

The following platforms support QoS handling on TAP ports.

- DCS-7280E
- DCS-7280R
- DCS-7500E
- DCS-7500R
- DCS-7280R2

Note: QoS is not available on TAP aggregation ports on the DCS-7150.

20.1.4.7.2 QoS Handling Configuration

Trust Mode of TAP Ports

TAP ports are in QoS **untrusted** mode by default. This means that the QoS marking of an incoming packet is not trusted when determining the QoS attributes of the packet. Therefore, the default QoS handling takes place. Consider the default CoS to traffic class mapping in the following example.

```
switch(config)# show qos maps
[...]
 Cos-tc map:
   cos:  0  1  2  3  4  5  6  7
   ---------------------------
   tc:   1  0  2  3  4  5  6  7
[...]
```

The Class of Service (CoS) field of incoming packets is ignored and is assumed to be zero. In this example, all packets are assigned to traffic class 1 when using the above mapping.

To override the default trust mode behavior on a TAP port, use the `qos trust` command.

Example

The following commands override the default trust mode behavior on ***Ethernet port 1***, configuring it to use Class of Service (CoS) trust mode instead so that incoming packets will be placed in their CoS-marked classes.

```
switch(config-if-Et1)# qos trust cos
switch(config-if-Et1)#
```

Class of Service Rewrite of TAP Ports

By default, TAP ports do not override the existing Class of Service (CoS) field of incoming packets. In other words, the CoS marking of steered packets is not changed in any way.

However, the CoS field of added tags may change according to the traffic class to CoS mapping. For example, the identity tag added by TAP ports may have the CoS value from the global traffic class to CoS mapping. Consider the following mapping:

```
switch(config)# show qos maps
[...]
   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ---------------------------
```

```
     cos:  1  7  2  3  4  5  6  0
[...]
```

Using this mapping, the added tag CoS field of packets assigned to ***traffic class 1*** may be set to ***7***.

20.1.4.7.3 Displaying QoS Handling Status

Use the **show qos maps** command to see the active QoS mappings.

Example

This command displays the QoS maps that are configured on the switch.

```
switch# show qos maps
Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
   Cos Rewrite:  Disabled
   Dscp Rewrite:  Disabled

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     --------------------------
     tc:   1  0  2  3  4  5  6  7

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :     1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  3  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  5  5  5  5  5  5  5  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     --------------------------
     cos:  1  0  2  3  4  5  6  7

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ------------------------------
     dscp:  8  0 16 24 32 40 48 56

   Tc - tx-queue map:
     tc:         0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:  0  1  2  3  4  5  6  7 switch#
```

20.1.4.8 Identity VLAN Tagging

By default, tool port output packets are identical to the replicated packets they receive from the tap ports to which they are associated. Identity tagging modifies packets sent by tool ports by adding a dot1q VLAN tag that identifies the originating TAP port. Each TAP port is associated with an identity number. Tool ports that are configured to add an identity tag append the originating TAP port's identity number in the outer layer (or s-VLAN) tag.

The following sections describe identity VLAN tagging on TAP and tool ports.

- Tap Port Identity Value Configuration
- Tool Port Identity Tag Configuration

Tap Port Identity Value Configuration

The **`switchport tap identity`** command configures the TAP port identity value for the configuration-mode interface. The default identity value for all TAP ports is ***1***.

Example

These commands configure ***1042*** as the identity value for ***interface ethernet 42*** and display the result.

```
switch(config)# interface ethernet 42
switch(config-if-Et42)# switchport tap identity 1042
switch(config-if-Et42)# show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation
 Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------
------------
Et41      tap            tap            400      1    0          ---
Et42      tap            tap            1        1042 0          ---
Et43      tap            tap            1        1    0          ---
switch(config-if-Et42)#
```

Tool Port Identity Tag Configuration

The **`switchport tool identity`** command configures the configuration-mode interface to include a tier-1 VLAN tag (dot1q) in packets it transmits. The VLAN number on the dot1q tag is the identity value configured for the TAP port that supplies the packets. By default, tool ports do not encapsulate packets with the tier-1 VLAN tag.

Example

These commands configure ***port channel 102*** to include the identity tag in packets it transmits.

```
switch(config)# interface port-channel 102
switch(config-if-Po102)# switchport tool identity dot1q
switch(config-if-Po102)# show interface port-channel 101-103 tool
Port      Configured     Status         Allowed             Id
 Timestamp
          Mode                          Vlans               Tag  Mode
-------------------------------------------------------------
-----------
Po101     tool           tool           1001-1020           Off  ---
Po102     tool           tool           1010-1015           On   ---
Po103     tool           tool           1010-1030           Off  ---
switch(config-if-Po102)#
```

20.1.4.9 TAP Aggregation Group Configuration

TAP aggregation groups associate a set of TAP ports with a set of tool ports. A tool port replicates packets it receives from TAP ports that are in the aggregation groups to which it belongs. A TAP port can be configured

to send data to multiple TAP aggregation groups. Tool ports may belong to multiple TAP aggregation groups. TAP aggregation groups may contain multiple TAP ports and multiple tool ports.

The following sections describe the configuration of TAP aggregation groups:

- Assigning a Tool Port to a TAP Aggregation Group
- Assigning TAP Ports to a TAP Aggregation Group
- Viewing TAP Aggregation Group Assignments
- LAGs in Tool Groups

Assigning a Tool Port to a TAP Aggregation Group

Tool ports are assigned to a TAP aggregation group through the **`switchport tool group`** command. Each command either creates a list or alters the existing list of groups to which a tool port belongs.

Examples

- These commands assign ***port-channel interface 101*** to TAP aggregation groups **analyze1**, ***analyze2***, and ***analyze3***.

```
switch(config)# interface port-channel 101
switch(config-if-Po101)# switchport tool group set analyze1 analyze2
 analyze3
switch(config-if-Po101)# show active
interface Port-Channel101
   switchport mode tool
   switchport tap identity 2101
   switchport tool allowed vlan 1001-1020
   switchport tap default group tag-9
   switchport tool group set analyze3 analyze1 analyze2
switch(config-if-Po101)#
```

- These commands remove **analyze-1** from ***port channel 101***'s TAP aggregation group list.

```
switch(config-if-Po101)# switchport tool group remove analyze1
switch(config-if-Po101)# show active
interface Port-Channel101
   switchport mode tool
   switchport tap identity 2101
   switchport tool allowed vlan 1001-1020
   switchport tap default group tag-9
   switchport tool group set analyze3 analyze2
switch(config-if-Po101)#
```

Assigning TAP Ports to a TAP Aggregation Group

TAP ports are assigned to a TAP aggregation group using the **`switchport tap default group`** command.

Note: A TAP port has multiple default groups.

Multiple ports can be added to a group simultaneously by entering the command in group interface configuration mode for all of the ports to be included.

Example

These commands assign ***interface ethernet 41-42*** to TAP aggregation group ***analyze2*** and assign ***interface ethernet 43*** to TAP aggregation group ***analyze3***.

```
switch(config)# interface ethernet 41-42
switch(config-if-Et41-42)# switchport tap default group analyze2
switch(config-if-Et41-42)# interface ethernet 43
switch(config-if-Et43)# switchport tap default group analyze2
switch(config-if-Et43)# show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation
 Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------
------------
Et41      tap            tap            400      1    0
 analyze2
Et42      tap            tap            1        1042 0
 analyze2
Et43      tap            tap            1        1    0
 analyze3
switch(config-if-Et43)#
```

Default TAP Aggregation Groups

A TAP port has multiple default groups. When traffic entering a TAP port does not match any filtering or traffic-steering rules for TAP aggregation groups configured on that port, it is sent to all default groups.

Example

These commands assign ***interface ethernet 43*** to TAP aggregation groups **analyze2** and **analyze3**. Because it is listed first, **analyze2** is configured as the default group for the interface.

```
switch(config)# interface ethernet 43
switch(config-if-Et43)# switchport tap default group analyze2 group
 analyze3
switch(config-if-Et43)#
```

Viewing TAP Aggregation Group Assignments

TAP aggregation group membership is displayed by **`show tap aggregation groups`**. Options allow the display of individual groups or of all configured groups. The command displays active tool and TAP ports by default, and provides an option to display configured ports that are not active.

Example

This command displays the contents of all configured TAP aggregation groups.

```
switch# show tap aggregation groups
Group Name                          Tool Members
-----------------------------------------------------
analyze2                            Po101, Po102
analyze3                            Po101, Po103

Group Name                          Tap Members
-----------------------------------------------------
analyze2                            Et41, Et42
```

```
analyze3                                  Et43 switch#
```

LAGs in Tool Groups

Link Aggregation Groups (LAGs) can be included in tool groups for load balancing. A tool group can contain both LAGs and regular ports. Each member of a tool group receives one copy of the traffic destined to the group. Traffic is replicated to tool group members using multicast replication. The traffic replicated to LAGs is then load balanced to their members as per load-balance policies configured on the system.

If a tool group has no more than ***60*** members with at least one hardware LAG, then the replication mode of the tool group is set to **ingress-only**. Otherwise, the replication mode of the tool group is set to the configured system default multicast replication mode. See **`platform sand multicast replication default`** for more information on configuration of the system default replication mode.

Example

The following command changes the system-wide default multicast replication mode to **ingress**.

```
switch(config)# platform sand multicast replication default ingress
switch(config)#
```

20.1.5 TAP Aggregation Traffic Steering

Traffic steering is a TAP aggregation process that uses class maps and policy maps to direct data streams at tool ports that are not otherwise associated to the ingress TAP port. A policy map is a data structure that filters data streams upon which identity VLAN tagging or TAP aggregation group assignment is implemented.

TAP-aggregation class maps and policy maps are similar to QoS and control-plane maps. However, policy maps and their components are not interchangeable among function types.

20.1.5.1 TAP Aggregation Policies

A policy map filters data packets by using classes and match rules. Each class contains an eponymous class map and a traffic resolution command. Each match rule contains packet content descriptors and a traffic resolution parameter.

- A class map uses ACLs that identify packets that comprise a specified data stream.
- Packet content descriptors specify packet field values that are compared to inbound packets.
- A traffic resolution command or parameter specifies data handling methods for filtered traffic.

Each data packet entering an entity to which a policy map is assigned is managed as defined by the traffic resolution command of the highest priority class or rule that matches the packet.

Class maps are user-created and can be edited or deleted. They filter traffic with IPv4 ACLs and are listed in ***running-config***. TAP aggregation traffic resolution commands do one the following:

- specify a TAP aggregation group to direct the packet.
- specify a VLAN number for identity tagging the packet.

TAP aggregation policy maps do not define an implicit **deny** statement. Packets that do not match a policy map class or rule are replicated and sent out tool ports specified by the default aggregation group assigned to the ingress TAP port. If no default group is selected, these packets are dropped.

20.1.5.2 Configuring TAP Aggregation Traffic Policies

TAP aggregation traffic policies are implemented by creating class maps and policy maps, then applying the policy maps to Ethernet and port-channel interfaces.

Creating Class Maps

A class map is an ordered list of IPv4 Access Control Lists (ACLs). Each ACL is assigned a sequence number that specifies its priority in the class map. TAP aggregation class maps utilize ACL permit rules to pass packets and deny rules to drop packets.

Class maps are created and modified in class-map configuration mode, which is entered using the **`class-map type tapagg`**. The **match (class-map (tapagg))** command inserts a specified ACL into the class map, assigning it a sequence number that denotes its placement.

Class-map configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **`show active`** command displays the saved version of class map. The **`exit`** command returns the switch to ***global*** configuration mode and saves pending class-map changes. The **`abort`** command returns the switch to global configuration mode and discards pending changes.

Examples

- This command creates a TAP aggregation class map named ***t-class_1*** and places the switch in the ***class-map*** configuration mode.

```
switch(config)# class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)#
```

- These commands add two IPv4 ACLs (***tacl-1*** and ***tacl-2***) to the ***t-class_1*** class map. The commands use the default method of assigning sequence numbers to the ACLs.

```
switch(config-cmap-t-class_1)# match ip access-group tacl-1
switch(config-cmap-t-class_1)#match ip access-group tacl-2
switch(config-cmap-t-class_1)#
```

- These commands exit class-map configuration mode, store pending changes to ***running-config***, then display the class map.

```
switch(config-cmap-t-class_1)# exit
switch(config)# class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)# show active
class-map type tapagg match-any t-class_1
10 match ip access-group tacl-1
20 match ip access-group tacl-2
switch(config-cmap-t-class_1)#
```

Creating Policy Maps

Policy maps are created and modified in ***policy-map*** configuration mode. A policy map is an ordered list of classes and match rules. Policy maps are edited by adding or removing map elements. Data packets are managed by commands of the highest priority class or rule that matches the packet.

Classes

Each class contains a class map, a set command, and a sequence number:

- The **class map** identifies a data stream by using an ordered list of ACLs. Class maps are configured in class-map (tapagg) configuration mode.
- The **`set`** command specifies the replication method for filtered data packets, either through an associated aggregation group or identity VLAN tagging.

- The **sequence number** specifies the class's priority within the policy map. Lower sequence numbers denote higher priority.

Matching Rules

Each rule contains a filter list, an action, and a sequence number:

- The **filter list** identifies a data stream by using a set of packet field values.
- The action, (**SET_VALUE** parameter) specifies the replication method of filtered data packets, either through an associated aggregation group or identity VLAN tagging.
- The **sequence number** specifies the rule's priority within the policy map. Lower sequence numbers denote higher priority.

Policy-map and ***policy-map-class*** configuration modes are group-change modes. Changes are saved with the **`exit`** command or discarded with the **`abort`** command. The **`show active`** and **`show pending`** commands display the saved and modified policy map versions respectively.

The **`class (policy-map (tapagg))`** command enters policy-map configuration mode.

Example

This command creates the TAP aggregation policy map named ***t-policy_1*** and places the switch in ***policy-map*** configuration mode.

```
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#
```

The **`class (policy-map (tapagg))`** command adds a class to the configuration mode policy map and places the switch in ***policy-map-class*** configuration mode for adding a traffic resolution command to the class. The **`set (policy-map-class (tapagg))`** command specifies the data replication method for traffic filtered by the associated class map in the *configuration-mode* policy map. The **`set`** command performs one of the following replication actions for filtered data packets.

- specifies an aggregation group.
- specifies a VLAN identity tag for replicated packets.
- specifies an aggregation group and a VLAN identity tag.

Examples

- These commands add the ***t-class_1*** class map to the ***t-policy_1*** policy map, associate a set statement with the class, then save the policy map by exiting the modes. Packets filtered by the class map are identity tagged with ***VLAN 444*** and replicated as specified by the ***t-grp*** aggregation group.

  ```
  switch(config-pmap-t-policy_1)# class t-class_1
  switch(config-pmap-c-t-policy_1-t-class_1)# set aggregation-group
  t-grp id-tag 444
  switch(config-pmap-c-t-policy_1-t-class_1)# exit
  switch(config-pmap-t-policy_1)# exit
  switch(config)# policy-map type tapagg t-policy_1
  switch(config-pmap-t-policy_1)# show active
  policy-map type tapagg t-policy_1
  10 class t-class_1
  set aggregation-group t-group id-tag 444
  switch(config-pmap-t-policy_1)#
  ```

  The **match (policy-map (tapagg))** command adds a match rule to the configuration-mode TAP aggregation policy map.
- This command enters policy-map configuration mode for ***t-policy_1***, then creates a match rule for the policy map that filters OSPF packets and replicates them as specified by ***t-grp*** TAP aggregation group.

  ```
  switch(config-pmap-t-policy_1)# match ip ospf any any set
  ```

```
aggregation-group t-grp
switch(config-pmap-t-policy_1)#
```

Applying Policy Maps to an Interface

The **service-policy type tapagg (Interface mode)** command applies a specified policy map to the configuration-mode interface.

Example

These commands apply the ***t-policy_1*** policy map to ***interface ethernet 17***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# service-policy type tapagg input tpolicy_1
switch(config-if-Et17)#
```

Stripping VLAN Tags

The traffic-steering policies in tap aggregation mode allows steering traffic from tap ports to tool ports using **set (policy-map-class (tapagg))** command, while the 'set id-tag' tags the traffic with the specified VLAN ID in the dot1q format. The **class (policy-map (tapagg))** command allows removing VLAN tags from the steered traffic. It supports all traffic types that the traffic steering policies support such as IPv4, IPv6, and MAC.

A tap port is an interface that receives a data stream where two network ports exchange.

A tool port is an interface that replicates data streams received by one or more tap ports. Tool ports connect to the devices that process monitored data streams.

Example

These commands place the switch in policy-map-class to add the ***t-class_1*** class map to the ***t-policy_1*** policy map. The first, second, or both of the two outer-most VLAN tags are stripped.

```
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# class t-class_1
switch(config-pmap-t-policy_1t-class_1)# set aggregation-group t-group remove
 dot1q outer 1-2
switch(config-pmap-c-t-policy_1-t-class_1)# set aggregation-group t-group id-
tag 10
switch(config-pmap-c-t-policy_1-t-class_1)# set id-tag 10 remove dot1q outer 1
switch(config-pmap-c-t-policy_1-t-class_1)# set aggregation-group t-group
switch(config-pmap-c-t-policy_1-t-class_1)# set id-tag 10
switch(config-pmap-c-t-policy_1-t-class_1)# set aggregation-group t-group id-
tag 10 remove dot1q outer 1-2
```

20.1.5.3 Traffic Steering to Match Inner Header Fields

This feature allows matching the inner header fields (either IPv4 or IPv6 inner fields) of encapsulated traffic.

The following convention is adopted when describing an encapsulated traffic: ***<inner-protocol>-over-<outer-protocol>***. For example, the IPv4-over-IPv6 packet indicates that the inner fields belong to IPv4 protocol and the outer fields belong to IPv6 protocol.

Supported Packet Types and Inner Header Fields

The types of traffic for which inner header field matching is supported are:

1. IP-over-IP traffic
2. IP-over-GRE traffic, with the following packet format supported:
   - [Eth | IPv4 or IPv6 | GRE | Inner IPv4 or IPv6]
   - [Eth | IPv4 | GRE | Inner Eth | Inner IPv4], which will be referred to as L2-GRE packet

3. IP-over-GTP traffic (only in DCS-7280R3 and DCS-7500R3 series), with the following packet format supported
   - [Eth | IPv4 | GTP | Inner IPv4 or IPv6]
   - [Eth | IPv6 | GTP | Inner IPv4 or IPv6]

The following inner header fields are supported:

- inner source IP address
- inner destination IP address
- inner TCP/UDP protocol
- inner source port number
- inner destination port number.

20.1.5.3.1 Configuring Traffic Steering to Match Inner Header Fields

To enable traffic steering with an ACL rule that matches the inner IP addresses, a user-defined TCAM profile must be configured for each type of IP protocol to which the inner IP addresses belong. For example, to match the inner IPv4 addresses of an IPv4 packet (i.e. IPv4-over-IPv4 traffic) or the inner IPv4 addresses of an IPv6 packet (i.e. IPv4-over-IPv6 traffic), one user-defined TCAM profile must be configured. A different user-defined TCAM profile is needed if matching the inner IPv6 addresses of an IPv4 packet or the inner IPv6 addresses of an IPv6 packet is desired.

The following commands show how to configure different TCAM profiles needed to enable support for matching inner IP headers, depending on the desired inner IP version, and inner IP header fields to be matched. These TCAM profiles are referred to as user-defined TCAM profiles and created based on the existing TCAM profile tap-aggregation-extended.

TCAM Profile to Match Inner IPv4 Header Fields

Match inner IPv4 addresses only.

The following commands configure the TCAM profile for matching only the inner source and destination IP addresses of IPv4 traffic based on the existing TCAM profile **tap-aggregation-extended**.

Example

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile tap-aggregation-user-inner-ip4 copy tap-
aggregation-extended
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# feature tapagg
 ip
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# no key field src-ip dst-ip
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# key field inner-src-ip-high inner-src-ip-low inner-dst-ip-high inner-dst-
ip-low
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# feature tapagg ipv6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
pv6)# no key field src-ipv6 src-ipv6-high dst-ipv6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
pv6)# key field inner-src-ip-high inner-src-ip-low inner-dst-ip-high inner-dst-
ip-low
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
pv6)# exit
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# exit
```

Match inner IPv4 addresses, inner TCP/UDP protocol and inner port numbers for IP-over-IP and IP-over-GRE traffic.

In addition to matching inner IPv4 addresses, the previous user-defined TCAM profile can also be configured to match the inner TCP/UDP protocol and their corresponding port numbers. The commands to configure these inner fields are shown below.

Note that, due to hardware limitation, the feature “tapagg mac” is disabled in this TCAM profile to make room for matching additional inner header fields.

Example

```
switch(config-hw-tcam)# profile tap-aggregation-user-inner-ip4
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# no feature
 tapagg mac
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# feature tapagg
 ip
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# no key field outer-vlan-id inner-vlan-id l4-src-port l4-dst-port udf-16b-1
 udf-16b-2 udf-16b-3 udf-16b-4 udf-32b-1 udf-32b-2
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# key field inner-ip-protocol inner-l4-src-port inner-l4-dst-port
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# feature tapagg ipv6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
pv6)# no key field l4-src-port l4-dst-port
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
pv6)# key field inner-ip-protocol inner-l4-src-port inner-l4-dst-port
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
pv6)# exit
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# exit
```

Match inner IPv4 addresses, inner TCP/UDP protocol and inner port numbers for L2-GRE packets.

The same user-defined TCAM profile can be updated to support traffic steering that matches on inner header fields for L2-GRE packets, simply by adding a new packet type.

Example

```
switch(config-hw-tcam)# profile tap-aggregation-user-inner-ip4
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# feature tapagg
 ip
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# packet ipv4 eth ipv4 forwarding bridged
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# feature tapagg
 port
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-p
ort)# packet ipv4 eth ipv4 forwarding bridged
```

Match inner IPv4 addresses, inner TCP/UDP protocol and inner port numbers for IP-over-GTP packets.

We can re-use the same user-defined TCAM profile and update it to support traffic steering that matches on inner header fields for IP-over-GTP packets, simply by adding a new packet type.

Example

```
switch(config-hw-tcam)# profile tap-aggregation-user-inner-ip4
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# feature tapagg
 ip
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# no packet ipv4 eth ipv4 forwarding bridged
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-i
p)# packet ipv4 gtpv1 ipv4 forwarding bridged
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# feature tapagg
 port
```

```
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-p
ort)# no packet ipv4 eth ipv4 forwarding bridged
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-p
ort)# packet ipv4 gtpv1 ipv4 forwarding bridged
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4)# feature tapagg
 ipv6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip4-feature-tapagg-p
ort)# packet ipv6 gtpv1 ipv4 forwarding bridged
```

Match inner and outer IPv4 addresses.

If matching a combination of inner and outer IP header fields is desired, the user-defined TCAM profile must be updated to include the desired fields. For example, instead of matching both inner source and inner destination IPv4 addresses as illustrated by this TCAM profile, if users would like to match outer source IPv4 address and inner destination IPv4 address only (and ignoring outer destination and inner source IPv4 addresses). The following commands show how to configure a TCAM profile for such filtering.

Example

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile tap-aggregation-user-outer-inner-ip4 copy tap-
aggregation-extended
switch(config-hw-tcam-profile-tap-aggregation-user-outer-inner-ip4)# feature
 tapagg ip
switch(config-hw-tcam-profile-tap-aggregation-user-outer-inner-ip4-feature-ta
pagg-ip)# no key field dst-ip
switch(config-hw-tcam-profile-tap-aggregation-user-outer-inner-ip4-feature-ta
pagg-ip)# key field inner-src-ip-high inner-src-ip-low
switch(config-hw-tcam-profile-tap-aggregation-user-outer-inner-ip4-feature-ta
pagg-ip)# exit
switch(config-hw-tcam-profile-tap-aggregation-user-outer-inner-ip4)# exit
```

TCAM Profile to Match Inner IPv6 Header Fields

Match inner IPv6 addresses only.

These commands are to configure a TCAM profile that can match inner IP addresses of IPv6 traffic are as follows:

Example

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile tap-aggregation-user-inner-ip6 copy tap-
aggregation-extended
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# feature tapagg
 ip
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-
ip)# no key field dst-ip src-ip udf-16b-1  udf-16b-2 udf-16b-3 udf-16b-4
 udf-32b-1 udf-32b-2 outer-vlan-id inner-vlan-id l4-src-port l4-dst-port l4-
ops
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
p)# key field inner-src-ipv6-high-high-32b inner-src-ipv6-high-low-32b inner-
src-ipv6-low-high-32b inner-src-ipv6-low-low-32b
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
p)# key field inner-dst-ipv6-high-high-32b inner-dst-ipv6-high-low-32b inner-
dst-ipv6-low-high-32b inner-dst-ipv6-low-low-32b
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
p)# exit
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# feature tapagg
 ipv6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
pv6)# no key field src-ipv6 src-ipv6-high dst-ipv6 l4-dst-port l4-src-port
```

```
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
pv6)# key field inner-src-ipv6-high-high-32b inner-src-ipv6-high-low-32b inner-
src-ipv6-low-high-32b inner-src-ipv6-low-low-32b
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
pv6)# key field inner-dst-ipv6-high-high-32b inner-dst-ipv6-high-low-32b inner-
dst-ipv6-low-high-32b inner-dst-ipv6-low-low-32b
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
pv6)# exit
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# exit
```

Match inner IPv6 addresses, inner TCP/UDP protocol and inner port numbers.

In addition to matching inner IPv6 addresses, the previous user-defined TCAM profile can also be configured to match the inner TCP/UDP protocol and their corresponding port numbers. The commands to configure these inner fields are shown below.

Note that, due to hardware limitation, the feature **tapagg mac** is disabled in this TCAM profile to make room for matching additional inner header fields.

Example

```
switch(config-hw-tcam)# profile tap-aggregation-user-inner-ip6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# no feature
 tapagg mac
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# feature tapagg
 ip
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
p)# key field inner-ipv6-next-header inner-l4-src-port inner-l4-dst-port
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
p)# feature tapagg ipv6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
pv6)# key field inner-ipv6-next-header inner-l4-src-port inner-l4-dst-port
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
pv6)# exit
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# exit
```

Match inner IPv6 addresses, inner TCP/UDP protocol and inner port numbers for IP-over-GTP packets.

We can re-use the same user-defined TCAM profile and update it to support traffic steering that matches on inner header fields for IP-over-GTP packets, simply by adding a new packet type.

```
switch(config-hw-tcam)# profile tap-aggregation-user-inner-ip6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# feature tapagg
 ip
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-i
p)# packet ipv4 gtpv1 ipv6 forwarding bridged
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# feature tapagg
 port
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-p
ort)# packet ipv4 gtpv1 ipv6 forwarding bridged
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6)# feature tapagg
 ipv6
switch(config-hw-tcam-profile-tap-aggregation-user-inner-ip6-feature-tapagg-p
ort)# packet ipv6 gtpv1 ipv6 forwarding bridged
```

Apply TCAM Profile

To apply the newly-defined TCAM profile use **tap-aggregation-user-inner-ip4** or **tap-aggregation-user-inner-ip6** command.

```
switch(config)# tap aggregation
```

```
switch(config-tap-agg)# mode exclusive profile tap-aggregation-user-inner-ip4
```

For modular systems

```
switch(config)# tap aggregation
switch(config-tap-agg)# mode mixed module Linecard <linecard number> profile
 tap-aggregation-user-inner-ip4
```

Set-Up ACL Rules for Matching Inner Header Fields

In order to apply traffic steering for particular packets based on their inner header fields, ACL rules that can filter such traffic must be created, as shown in the example.

1. ACL rule for filtering an IPv4-over-IPv4 traffic based on the inner addresses.

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# permit ip any any inner ip host 1.2.3.4 host
 5.6.7.8
```

2. ACL rule for filtering an IPv4-over-IPv4 traffic based on the outer source and inner destination addresses.

```
switch(config)# ip access-list acl1
switch(config-acl-acl1)# permit ip host 1.2.3.4 any inner ip host any host
 5.6.7.8
```

3. ACL rule for filtering an IPv6-over-IPv4 traffic based on the inner addresses.

```
switch(config)# ip access-list acl2
switch(config-acl-acl2)# permit ip any any inner ipv6 host 1001::abcd host
 2002::cafe
```

4. ACL rule for filtering an IPv4-over-IPv6 traffic based on the inner addresses, inner TCP/UDP protocol, and inner TCP/UDP ports.

```
switch(config)# ipv6 access-list acl3
switch(config-ipv6-acl-acl3)# permit ipv6 any any inner ip tcp host
 10.11.12.13 eq 9999 host 20.22.24.26 eq 8888
```

5. ACL rule for filtering an IPv6-over-GRE(IPv4) traffic based on the inner addresses, inner TCP/UDP protocol, and inner TCP/UDP ports.

```
switch(config)# ip access-list acl4
switch(config-acl-acl4)# permit gre any any inner ipv6 udp host 2019::baba
 eq 9999 host 1986::deca eq 8888
```

6. ACL rule for filtering an IPv4-over-GTP traffic based on the inner addresses.

```
switch(config)# ip access-list acl5
switch(config-acl-acl5)# permit gtp any any version 1 protocol gtp-u inner
 ip host 1.2.3.4 host 5.6.7.8
```

20.1.5.3.2 Limitations

- Only IP (IPv4 and IPv6), GRE and GTP outer protocols are supported.
  - Other outer protocols are not supported.
- Only IP (IPv4 and IPv6), TCP and UDP inner protocols are supported.
  - Other inner protocols are not supported.

- To configure support for matching inner IPv4 addresses (for either IPv4 or IPv6 outer packets), a custom user-defined TCAM profile is required. A different user-defined TCAM profile is needed if matching inner IPv6 addresses is desired.
- The total size of fields to match must not exceed the TCAM key size limitation (which is ***320*** bit if tap-aggregation-extended system profile is used as the base profile, or ***160*** bit for tap-aggregation-default system profile). For example, it is not possible to create a user-defined TCAM profile that matches both outer and inner source/destination address pairs for an IPv6-over-IPv6 packet. This is because each IPv6 address is ***128*** bit, requiring at least ***512*** bit to match all ***4*** addresses, and the TCAM key size is limited to ***320*** bit.

20.1.6 TAP Aggregation GUI

The switch provides a Graphical User Interface (GUI) for creating and viewing a TAP aggregation configuration and displaying LANZ traffic statistics.

All commands available on the GUI are accessible through the CLI. The TAP aggregation configuration created through either the CLI or the GUI can be viewed and modified through either medium.

This section provides a brief description of the TAP aggregation GUI.

20.1.6.1 Accessing the TAP Aggregation GUI

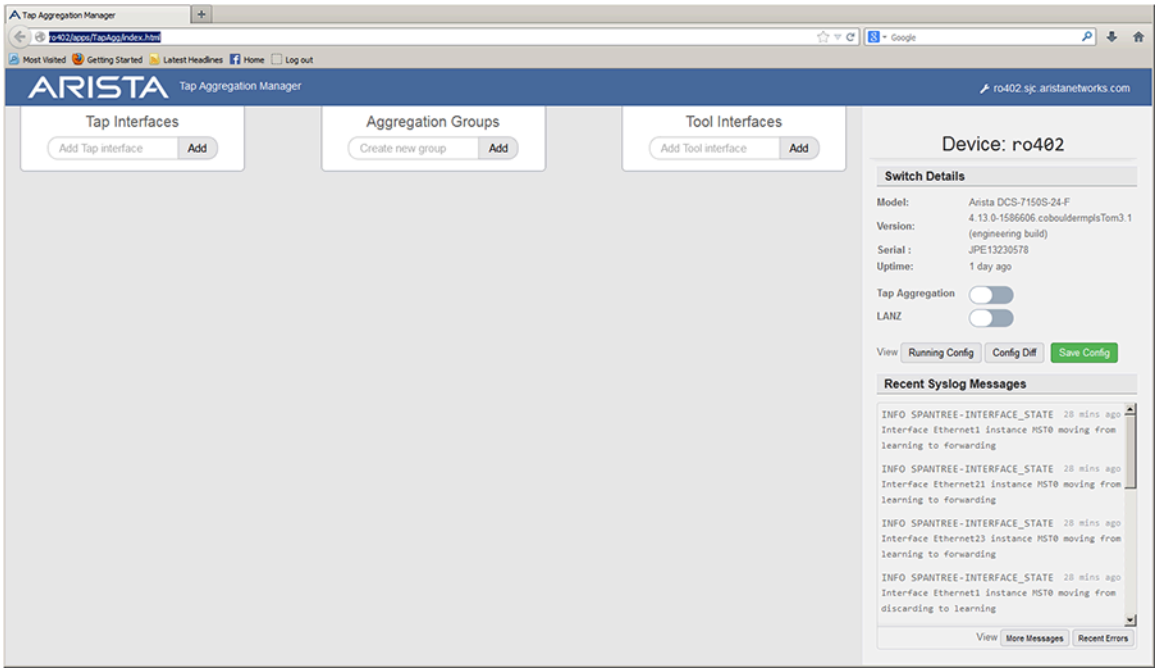
The URL for the TAP aggregation GUI is: **`//hostname/apps/TapAgg/index.html`** where the hostname is the switch's configured hostname. The **`TAP Aggregation GUI Initial Panel`** displays the initial TAP aggregation GUI panel for the switch with the hostname ***ro402***.

The TAP aggregation panel contains two sections:

- The **configuration section** displays the TAP aggregation configuration, including the TAP interfaces, tool interfaces, and aggregation groups. Links are displayed to indicate interface group membership.
- The **component section** displays information and control buttons for the active configuration entity. When an entity is not selected, the section displays information for the switch (device).

The configuration section displays TAP aggregation components only when the switch is in TAP aggregation mode. To enter TAP aggregation mode, click the TAP Aggregation icon in the component section for the device. The icon is a toggle mechanism; clicking it again disables TAP aggregation mode.

Figure 20-1: TAP Aggregation GUI Initial Panel



20.1.6.2 Viewing TAP Aggregation Component Details

TAP Aggregation GUI Panel with TAP Aggregation Mode Enabled displays the TAP aggregation panel when the switch is in TAP aggregation mode. The configuration section indicates that the TAP aggregation configuration consists of three tool interfaces, one TAP interface, and four aggregation groups. ***Ethernet port 10*** is the active component; configuration control and traffic information for this interface is available in the component section.

The active component is changed by clicking on the desired component in the configuration section. To display device (switch) information, click on any configuration section outside of any component.

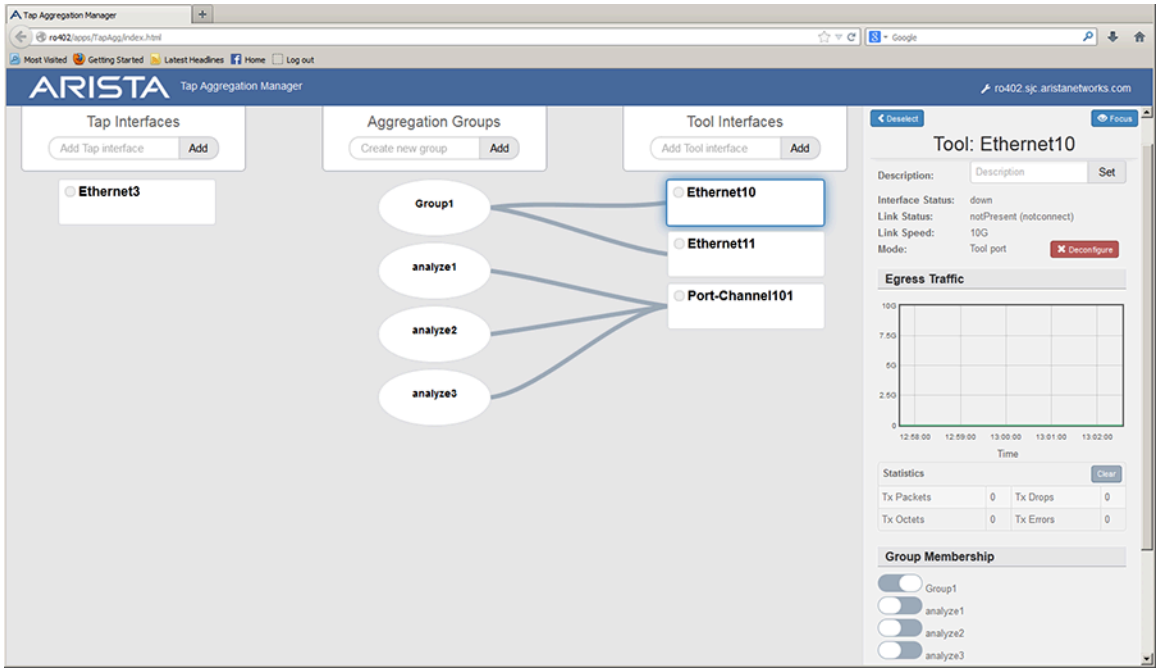
20.1.6.3 Modifying a TAP Aggregation Configuration

The TAP aggregation configuration can be modified only when the switch is in TAP aggregation mode, (see Accessing the TAP Aggregation GUI). The following is a partial list of configuration tasks that are available from the GUI:

- **adding a TAP or tool interface**: begin typing the interface name in the desired add-interface data entry area to access a drop-down list of available interfaces. Select the desired interface and press the **Add** button.
- **removing an interface from the configuration**: select the desired interface in the configuration section and click the deconfigure button in that interface's component section.
- **adding an aggregation group**: type the desired name of the new group in the data entry area and press the **Add** button. The TAP aggregation group name can consist of alphanumeric characters and specific special characters (**- _ [ ] { } :**) only.
- **adding an interface to an aggregation group**: select the desired interface in the configuration section, then press the icon of the group in the group membership area of the interface's component section.

Group icons are toggle buttons; clicking the icon of a group to which the interface belongs removes that interface from the group.

Figure 20-2: TAP Aggregation GUI Panel with TAP Aggregation Mode Enabled



20.1.7 TAP Aggregation Keyframe and Timestamp Configuration

This section contains the following topics:

- TAP Aggregation Keyframe Generation
- Enabling Timestamp Insertion on an Interface

20.1.7.1 TAP Aggregation Keyframe Generation

Keyframes contain routable IP packets that provide information to relate timestamps with the complete ASIC counter and absolute UTC time. The switch supports a maximum of ten keyframes, which are distinguished by their name label. Each keyframe can egress from every Ethernet port.

Keyframe generation is enabled by the **`platform fm6000 keyframe`** command. Command options specify ports that transmit keyframes along with the destination MAC address and IP address in the keyframe's header. Other keyframe commands specify the transmission rate and the frame's source:

- the **`platform fm6000 keyframe rate`** command configures the keyframe's transmission rate.
- the **`platform fm6000 keyframe source`** command configures the source IP address that is placed in each keyframe's header. The management interface IP address is the default source address.

  The source MAC address is the MAC address of the egress interface transmitting the keyframe
- the **`platform fm6000 keyframe device`** command configures the 16-bit number that keyframes list as the device ID in their payload.
- the **`platform fm6000 keyframe fields skew`** command enables the inclusion of clock skew fields in the keyframe.
- the **`show platform fm6000 keyframe`** command displays keyframe configuration information.

Examples

- This command enables the generation of a keyframe named ***key-1*** and configures it to egress from interfaces ethernet ***11*** through ***15*** with a source IP address of ***10.21.1.4*** and a MAC address of ***10.4E21.9F11***.

```
switch(config)# platform fm6000 keyframe key-1 interface ethernet
 11-15 10.21.1.4 10.4E21.9F11
switch(config)#
```

- This command configures the generation rate for the keyframe of ***10*** frames per second on each of the five interfaces that it is configured to egress.

```
switch(config)# platform fm6000 keyframe key-1 rate 10
switch(config)#
```

- This command enables the generation of a keyframe named ***key-1*** and configures ***100*** as the value that is placed in the keyframe's device ID field.

```
switch(config)# platform fm6000 keyframe key-1 device 100
switch(config)#
```

- This command enables the inclusion of clock skew fields in the keyframe named ***key-1***.

```
switch(config)# platform fm6000 keyframe key-1 fields skew
switch(config)#
```

- This command displays configuration information for keyframe ***key-1***.

```
switch(config)# show platform fm6000 keyframe

Keyframe key-1
------------------------
Egress Interface(s): Ethernet11, Ethernet12, Ethernet13, Ethernet14,
 Ethernet15
Source IP: 172.22.30.142
Destination IP: 10.21.1.4
Destination MAC: 00:10:4e:21:9f:11
Device ID: 100
Rate: 10 packet(s) per second

switch(config)#
```

20.1.7.2 Enabling Timestamp Insertion on an Interface

Timestamps are based on a frame's ingress time and applied to frames sent on egress ports, ensuring that timestamps on monitored traffic reflect ingress timing of the original frames. Time-stamping is configured on the egress port where the timestamp is applied to the frame.

When timestamping is enabled on an egress interface, packets leave the interface with timestamps that were applied in hardware when the packet arrived at the switch. This is facilitated by applying a hardware timestamp to all frames arriving on all interfaces when timestamping is enabled on any interface, then removing timestamps on packets egressing interfaces where timestamping is not enabled.

The **`mac timestamp`** command enables time-stamping on the configuration-mode interface. The switch supports two timestamp modes, which differ in managing the egress frame's 32-bit Frame Check Sequence (FCS):

- **before-fcs**: the switch discards the original FCS, appends the ingress timestamp at the end of the frame data, recalculates a new FCS based on the appended timestamp, then appends the new FCS to the end of the frame. This creates a valid Ethernet frame but does not update headers of any nested protocols.

- **replace-fcs**: the switch replaces the original FCS with the timestamp. This mode maintains the size of the original frame without any latency impact, but the FCS is not valid.

Examples

- These commands enable timestamping in **before-fcs** mode on ***interface ethernet 44***.

```
switch(config)# interface ethernet 44
switch(config-if-Et44)# mac timestamp before-fcs
switch(config-if-Et44)# show active
interface Ethernet44
   mac timestamp before-fcs
switch(config-if-Et44)#
```

- These commands disable timestamping on ***interface ethernet 44***.

```
switch(config-if-Et44)# no mac timestamp
switch(config-if-Et44)# show active
interface Ethernet44
switch(config-if-Et44)#
```

20.1.8 TapAgg GRE Tunnel Termination

The TapAgg GRE Tunnel Termination feature terminates the GRE packets on a TapAgg switch by stripping the GRE header and then letting the decapped packets go through the normal TapAgg path. With this feature, we can use an L3 GRE tunnel to transit tapped traffic to the TapAgg switch over an L3 network. That would widely extend the available use cases for TapAgg.

- Support IPv4 GRE tunnel interfaces only.
- Support the following GRE types: IPoGRE, L2GRE, GREenSPAN, ERSPAN Type I, ERSPAN Type II, and GREenTAP.
- Do packet forwarding/steering on the decapped packets. Not routing.
- Packets can be forwarded to a set of tool ports.
- Have a command to configure for not stripping the GRE header, so that the TapAgg side may use the information present in the GRE metadata.

20.1.8.1 Configuring TapAgg GRE Tunnel Termination

The TapAgg GRE Tunnel Termination is allowed to be configured in the following two modes:

- TapAgg Exclusive Mode
- TapAgg Mixed Mode

20.1.8.1.1 TapAgg Exclusive Mode

In Tapagg Exclusive mode, GRE tunnel termination is enabled on a selected tap ports through CLI. When traffic comes into those tap ports, the matched GRE packets is decapped and forwarded to the tap port's default forwarding destinations or the steering destinations if it matches the policy ACLs applied.

Adding the TCAM Feature

Tap tunnel termination on tapAgg exclusive mode is supported by built-in profiles **tap-aggregation-default** and **tap-aggregation-extended**. Custom profiles can support tap tunnel termination with the configuration of the tapagg tunnel termination feature through the following steps:

```
switch# configure
```

```
switch(config)# hardware tcam
switch(config-tcam)# profile tap-aggregation-gre copy tap-aggregation-profile
switch(config-tcam-profile-tap-aggregation-gre)# feature tapagg tunnel
 termination copy system-feature-source-profile
switch(config-tcam-feature-tapagg-tunnel-termination)# exit
switch(config-tcam-profile-tap-aggregation-gre)# exit
Saving new profile 'tap-aggregation-gre'
switch(config-tcam)# exit
switch(config)#
```

To check if a TCAM profile has the feature, use the following command:

```
switch(config)# show hardware tcam profile tap-aggregation-gre
Features enabled in TCAM profile tap-aggregation-gre:
 mpls
 acl port ipv6
 tapagg port
 tapagg mac
 tapagg ip
 tapagg ipv6
 tapagg tunnel termination
 acl port ip
 tunnel vxlan
 acl port mac
 forwarding-destination mpls
```

Change Hardware Forwarding Profile

On DCS-7280R3, DCS-7500R3 and DCS-7800R3 systems, the hardware forwarding profile needs to be set to **system-profile-tap-aggregation** on all linecards.

Enable GRE Termination on a Tap Port

Use **`switchport tap encapsulation gre [tunnel destination <dst IP address> [source <src IP address> ]] strip`** to enable GRE termination on a tap port. The command allows specification of tunnel endpoints, to support termination on a specific tunnel.

```
switch(config-if-Et7/1)# switchport tap encapsulation gre destination <dst-ip>
 source <src-ip> strip
```

Without the specification of tunnel endpoints, it will terminate GRE packets for all tunnels. If only the destination endpoint is specified, then any GRE packet that matches this destination is terminated.

Example

In this example the following configuration tap port ***et7/1*** will strip GRE packets for all GRE tunnels and forward the decapped packets to tool group ***tool1***.

```
switch(config-if-Et7/1)# switchport mode tap
switch(config-if-Et7/1)# switchport tap default group tool1
switch(config-if-Et7/1)# switchport tap encapsulation gre strip
```

GRE Termination into Traffic Steering

If the matched GRE packet also hits the policy map that is applied on the tap port, after stripping the GRE header, the packets are forwarded to the steering destination configured for the policy map. In the example below, a policy map named ‘pm’ is applied on the tap port which has gre strip enabled on it. Then the GRE packets with an outer destination IP ***dst-ip*** and source IP ***src-ip***, matching the permit rule in the policy map, is forwarded to the tool group ***tool2*** aftering stripping the GRE header.

Note: The ACLs are applied on the original GRE packet, and not the one after the GRE header is stripped.

```
switch(config)# show policy-map type tapagg
Service policy pm
  Configured on:
  Applied on:
  10: Class-map: cm (match-any)
    Match: 10 IP Access List testAcl
        10 permit ip host 1.1.1.1 host 2.2.2.2

    Configured actions: set group tool2

switch(config-if-Et7/1)#show active
interface Ethernet7/1
   service-policy type tapagg input pm
   switchport mode tap
   switchport tap encapsulation gre strip
   switchport tap default group tool1
```

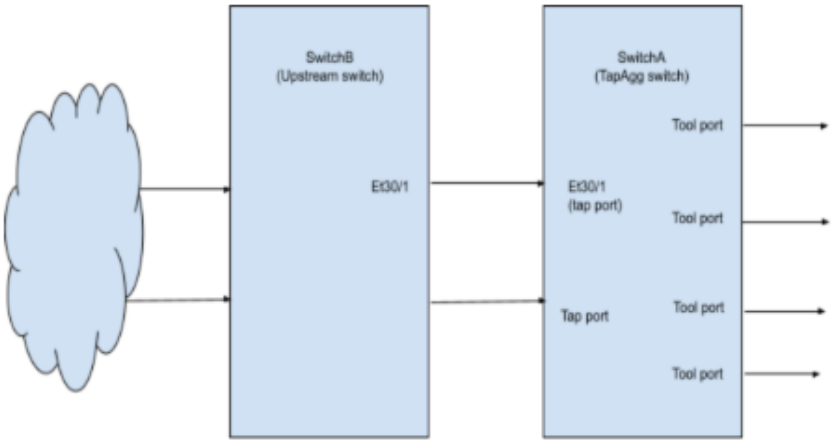
GRE Termination with Ingress/Egress Truncation

When there is truncation enabled, either on the tap port that has GRE termination configured or on the tool port that is configured to egress out the decapped packets, the GRE header is stripped from the truncated packets. In other words, the GRE header stripping occurs after truncation. That means the size of the egress packet will be less than the original expectation. For example, if we have configured to truncate packets to a size of ***178*** bytes, after enabling GRE termination, the egress out packet for a GREenSpan will have a size of ***136***. It is ***178*** minus ***42*** (the total bytes stripped for GREenSpan).

Configuring the Upstream Switch

In TapAgg exclusive mode, the routing protocols are not running so TapAgg Switch will rely on its upstream switch which has routing protocol enabled to forward the GRE packets to it. Tap ports which are selected to receive the GRE packets should be connected to the upstream switch. The upstream switch needs to configure a static route to forward the GRE packets to the link.

Example



If TapAgg switch A is connected to the upstream ***Switch B*** through the link between ***A-et30/1*** and ***B-et30/1*** then the static route for forwarding GRE packets with Destination IP address ***40.1.1.2*** terminated on TapAgg ***Switch A*** can be configured with the following commands:

```
switchB(config-if-Et30/1)# show active
interface Ethernet30/1
   speed forced 100gfull
   no switchport
   ip address 10.10.10.1/24
SwitchB(config-if-Et30/1)#ip route 40.1.1.2/32 ethernet 30/1
```

The static route should be present in the output of the **show ip route** command.

```
switchB# show ip route

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked,
       RC - Route Cache Route

Gateway of last resort is not set

 C        10.10.10.0/24 is directly connected, Ethernet30/1
 S        10.80.0.0/13 [1/0] via 10.240.25.1, Management1
 S        10.95.0.0/16 [1/0] via 10.240.25.1, Management1
 C        10.240.25.0/25 is directly connected, Management1
 S        10.240.0.0/15 [1/0] via 10.240.25.1, Management1
 S        10.242.0.0/15 [1/0] via 10.240.25.1, Management1
 S        40.1.1.2/32 is directly connected, Ethernet30/1
 S        172.16.0.0/12 [1/0] via 10.240.25.1, Management1
```

Setup static ARP entry for the route. ***44:4c:a8:be:35:e9*** is the MAC address of ***A-et30/1***. Note that any unicast MAC address can be used for static ARP entry.

```
switchB(config)# arp 40.1.1.2 44:4c:a8:be:35:e9 arpa
```

Use the following command to check ARP entries.

```
switchB# show arp
Address                  HWtype  HWaddress           Flags Mask
 Iface
10.240.25.1              ether   28:99:3a:67:7f:93   C                     ma1
40.1.1.2                 ether   44:4c:a8:be:35:e9   CM
 et30_1
```

With the above configurations, the upstream ***Switch B*** can forward the GRE packets to TapAgg ***Switch A***.

20.1.8.1.2 TapAgg Mixed Mode

For a modular system, such as the DCS-7500R or DCS-7500R2, routed ports on a non-TapAgg linecard is used to receive GRE packets. In order to use the tap tunnel termination feature, **dbTapTunnelTermination** must be added to the TCAM profile for the non-TapAgg linecards.

Adding the TCAM Feature

For example, if the current system profile is profile **default**, we can use the following commands to create a new profile **default-with-tap-gre**, which is copied from the current system profile, and add the new feature that we need.

```
switch(config)# hardware tcam
switch(config-tcam)# profile default-with-tap-gre copy default
switch(config-tcam-profile-default-with-tap-gre)# feature tapagg tunnel
 termination copy system-feature-source-profile
switch(config-tcam-feature-tapagg-tunnel-termination)# exit
switch(config-tcam-profile-default-with-tap-gre)# exit
Saving new profile 'default-with-tap-gre'
switch(config-tcam)# exit
```

Use the following command to apply the new TCAM profile.

```
switch(config)# hardware tcam
switch(config-tcam)# system profile default-with-tap-gre
```

Configure TapAgg mixed mode. In the below example, we configure TapAgg mixed mode and leave the linecard ***6*** in normal mode.

```
switch(config)# tap agg
switch(config-tap-agg)# mode mixed module linecard 3,4,5
! Changing modes may affect available functionality. Unsupported configuration
 elements will be ignored.
switch(config)#show hardware tcam profile
                    Configuration            Status
Linecard3           tap-aggregation-default* tap-aggregation-default
Linecard4           tap-aggregation-default* tap-aggregation-default
Linecard5           tap-aggregation-default* tap-aggregation-default
Linecard6           default-with-tap-gre     default-with-tap-gre
* configuration overridden by TapAgg
```

There could be some cases where we need to remove some features from the TCAM profile in order to have room to fit the feature **tapagg tunnel termination** in. This kind of ‘tradeoff’ is needed when we want this feature to be with a specific user-defined tcam profile.

Below is an example of solving that problem.

```
switch(config)# tap agg
switch(config-tap-agg)# mode mixed module linecard 3,4,5
! Changing modes may affect available functionality. Unsupported configuration
 elements will be ignored.
switch(config)#show hardware tcam profile
                    Configuration            Status
Linecard3           tap-aggregation-default* tap-aggregation-default
Linecard4           tap-aggregation-default* tap-aggregation-default
Linecard5           tap-aggregation-default* tap-aggregation-default
Linecard6           default-with-tap-gre     ERROR
* configuration overridden by TapAgg

switch(config)# hardware tcam
switch(config-tcam)# profile default-with-tap-gre
switch(config-tcam-profile-default-with-tap-gre)# no feature mirror ip
switch(config-tcam-profile-default-with-tap-gre)# no feature pbr ip
switch(config-tcam-profile-default-with-tap-gre)# exit
switch(config-tcam)# show hardware tcam profile
                    Configuration            Status
Linecard3           tap-aggregation-default* tap-aggregation-default
```

```
Linecard4            tap-aggregation-default* tap-aggregation-default
Linecard5            tap-aggregation-default* tap-aggregation-default
Linecard6            default-with-tap-gre     default-with-tap-gre
* configuration overridden by TapAgg
```

Change Hardware Forwarding Profile

On DCS-7280R3, DCS-7500R3 and DCS-7800R3 systems, the hardware forwarding profile needs to be set to system-profile-tap-aggregation on all tap agg linecards. Detailed instructions on how to set it can be found in the Resources section.

Configure GRE Tunnel Interface

Configure the GRE tunnel interface that we want the packets it received to be terminated and redirected.

In the below example, the GRE packets for this tunnel interface should have destination IP address ***40.1.1.1*** and source IP address ***40.1.1.2***.

```
switch(config)# interface tunnel 1
switch(config-if-Tu1)# tunnel mode gre
switch(config-if-Tu1)# tunnel source 40.1.1.1
switch(config-if-Tu1)# tunnel destination 40.1.1.2
```

For a tunnel interface, if **tunnel source** is configured but **tunnel destination** is not, then any GRE packet with destination IP address matching the tunnel source will be stripped and redirected. In the example below, the GRE packet with destination IP ***40.1.1.1*** is what we are interested in.

```
switch(config-if-Tu1)# show active
   tunnel mode gre
   tunnel source 40.1.1.1
   tap default group gr1
   tap default interface Ethernet3/1
```

If neither **tunnel source** nor **tunnel destination** is configured, then the GRE packets for any endpoints will be stripped and redirected.

```
switch(config-if-Tu1)# show active
   tunnel mode gre
   tap default group gr1
   tap default interface Ethernet3/1
```

Use the **`tap default group`** and **`tap default interface`** commands to configure the redirecting destination. In the example below, the GRE packets sent to ***interface tunnel 1*** will be redirected to tool group gr1 and tool ***interface et3/1*** after stripping the GRE header. (In current version, the target interfaces must be tool ports on TapAgg linecards. A non tool port putting in this command will not take effect.)

```
switch(config)# interface tunnel 1
switch(config-if-Tu1)# tap default group gr1
switch(config-if-Tu1)# tap default interface ethernet 3/1
switch(config-if-Tu1)#
```

If we want to redirect the packets but do not want to strip the GRE header, command **`tap encapsulation gre preserve`** can be configured for that purpose, not stripping the GRE header.

```
switch(config)# interface tunnel 1
switch(config-if-Tu1)# tap encapsulation gre preserve
switch(config-if-Tu1)#
```

The show command below is used to check the tunnel interface status. In TapAgg mixed mode, routing protocols can run on non-TapAgg linecards, and any routed port in non-TapAgg linecards can be used to receive the GRE packets. For Tap tunnel termination feature, the status of the tunnel interface doesn't have to be **connected**, but there must be some routes to help the GRE packets find the way to the routed ports on this switch.

```
switch(config)# show interfaces tunnel 1
Tunnel1 is up, line protocol is up (connected)
  Hardware is Tunnel, address is 2801.0102.0800
  Tunnel source 40.1.1.2, destination 40.1.1.1
  Tunnel protocol/transport GRE/IP
  Hardware forwarding enabled
  Tunnel transport MTU 1476 bytes (default)
  Up 35 minutes, 57 seconds
```

Enable Tap Tunnel Termination on TapAgg Linecards in Mixed Mode

If you do want to enable GRE Termination on a specific tap port in TapAgg mixed mode, follow the **TapAgg exclusive mode** section and refer to configuring TCAM profile and GRE termination for the TapAgg linecard.

20.1.9 Tap Aggregation Hardware Forwarding Profile

In order to enable the MPLS Pop and 802.1br-E/VN Tag Stripping features, the hardware forwarding profile needs to be set. Setting the hardware forwarding profile will not affect the functionality of any other features that are supported while in Tap Aggregation mode. However, changing the forwarding profile does interrupt forwarding for a short period of time while the new configuration is applied.

20.1.9.1 Configuring Tap Aggregation Hardware Forwarding Profile

- The default hardware forwarding profile can be set globally with the **`hardware forwarding system profile`** command:

  ```
  switch(config)# hardware forwarding system profile system-profile-tap-aggregation
  ```

- When not in Tap Aggregation mode the default hardware forwarding profile should be returned to the default system profile with the same command:

  ```
  switch(config)# hardware forwarding system profile system-profile-default
  ```

- For a modular system the system default profile can also be overridden on a per line card basis with the command:

  ```
  switch(config)# hardware forwarding module Linecard3 profile system-profile-tap-aggregation
  ```

  Note: If this command is used to set the hardware forwarding profile of a specific line card then changing the system default profile with the **`hardware forwarding system profile`** command will not affect that line card until the line card specific configuration is removed.

20.1.9.2 Show Commands

The following show command is available to see the hardware forwarding profile configuration:

```
(config)# show hardware forwarding profile
Linecard          Configured Profile
----------------- ------------------------------
FixedSystem       system-profile-tap-aggregation
```

```
System default profile: system-profile-tap-aggregation
```

For a modular system, this example shows the system default profile, configured by **hardware forwarding system profile**, is set to **system-profile-default**. While the configuration for line cards ***4*** and ***5*** has been overridden and is set to **system-profile-tap-aggregation**.

```
(config)# show hardware forwarding profile
Linecard        Configured Profile
--------------- ------------------------------
Linecard3       system-profile-default
Linecard4       system-profile-tap-aggregation
Linecard5       system-profile-tap-aggregation

System default profile: system-profile-default
```

20.1.9.3 Limitations

Changing the hardware forwarding profile on the system or on a specific line card will interrupt forwarding on that system or line card for a short period of time.

20.1.10 TAP Aggregation MPLS Pop

The MPLS pop supports tools that do not parse MPLS labels and therefore need the switch to remove (pop) the MPLS header. The MPLS pop supports both IPv4 and IPv6 over MPLS.

20.1.10.1 Configuring MPLS Pop

- On DCS-7280R/R2, DCS-7500R/R2, and DCS-7020R MPLS pop is configurable at the tap port using the switchport command:

  ```
  switch(config-if-Et1)# [no] switchport tap mpls pop all
  ```

- On DCS-7280R3, DCS-7500R3, and DCS7800R3 MPLS pop is configurable at the tool port using the switchport command. As the commands suggest, all MPLS labels will be popped.

  ```
  switch(config-if-Et1)# [no] switchport tool mpls pop all
  ```

- On DCS-7280R3, DCS-7500R3, and DCS7800R3 the hardware forwarding profile must also be set before MPLS pop can be used. This is done globally by the command:

  ```
  switch(config)# hardware forwarding system profile system-profile-tap-
  aggregation
  ```

- The system profile can also be overridden on a per line card basis with the command:

  ```
  switch(config)# hardware forwarding module Linecard3 profile system-profile-
  tap-aggregation
  ```

20.1.10.2 Limitations

R/R2 series MPLS pop has the following limitations:

- New L2 Header on Egress
  - Always inserts a tap port Identity VLAN ID, even if not configured. Default Identity VID is *1*.
  - Destination / source MAC addresses are not configurable.
- Tool Port / Egress VLAN Filtering

- Because the incoming L2 header is discarded, the tool port does not have access to the original VLAN ID.
- Tap ports can still filter on VLAN ID.

- ACL-Based Tap Aggregation
  - Matching IP headers contained inside an MPLS header is only supported since ***EOS-4.20.5F***.

On both R/R2 series and R3 series some Tap Aggregation features do not work when forwarding MPLS-encapsulated traffic:

- Non-IP over MPLS
  - If the incoming L3 header is not recognized as IPv4 or IPv6 the MPLS pop feature is bypassed and the packet is not stripped.

20.1.11 TAP Aggregation 802.1br EVN Tag Stripping

802.1br-E/VN Tag Stripping feature for Tap Aggregation mode strips IEEE 802.1BR E-Tag and Cisco VN-Tag headers from all tagged packets received on tap interface before delivering them out of tool interfaces. Untagged packets are unaffected. The 802.1br EVN Tag Stripping may be useful for third-party tools and/or packet analyzers which cannot parse these headers.

20.1.11.1 Configuring 802.1br-E/VN Tag Stripping

By default, Arista switches do not strip BR-E/VN tags from ingress packets.

On DCS-7280R/R2, DCS-7500R/R2, and DCS-7020R the BR-E/VN tag stripping is globally configured for Tap Aggregation. This means that packet ingressing any tap port will have their BR-E/VN tags stripped if the BR-E/VN tag stripping is enabled. The BR-E/VN tag stripping allows a choice of stripping both or either of the tags.

- To enable BR E-Tag stripping add the following configuration:

```
switch(config)# tap aggregation
switch(config-tap-agg)# [no] encapsulation dot1br strip
```

- To enable VN-TAG stripping add the following configuration:

```
switch(config)# tap aggregation
switch(config-tap-agg)# [no] encapsulation vn-tag strip
```

On DCS-7280R3, DCS-7500R3and DCS7800R3, BR-E/VN tag stripping is configured at the tool port level. When enabled on a tool port, only packets egressing this tool port will have their BR-E/VN tags stripped. The BR-E/VN tag stripping allows a choice of stripping both or either of the tags.

- To activate BR E-Tag stripping on a tool port, add the following configuration:

```
switch(config-if-Et1)# [no] switchport tool encapsulation dot1br strip
```

- To activate VN-TAG stripping on a tool port, add the following configuration:

```
switch(config-if-Et1)# [no] switchport tool encapsulation vn-tag strip
```

On DCS-7280R3, DCS-7500R3, and DCS7800R3 the hardware forwarding profile must also be set before BR-E/VN tag stripping is used. This is done globally by the command:

```
switch(config)# hardware forwarding system profile system-profile-tap-aggrega
tion
```

The system profile can also be overridden on a per line card basis with the command:

```
switch(config)# hardware forwarding module Linecard3 profile system-profile-
tap-aggregation
```

20.1.11.2 Show Commands

The tap or tool ports with BR-E/VN tag stripping configured is verified with the following show commands.

- On DCS-7280R/R2, DCS-7500R/R2, and DCS-7020R:

```
switch(config)# show interfaces tap tunnel
  Port        Configured         Status       Port            Allowed Vlans       Native     Truncation
              Mode                            Identity                            Vlan       Size
---------- ------------------ ------------ --------------- ------------------- ------------ -----------
  Et1/1       tap                tap          0               All                 1          0



   Port        VN Tag Action       BR Tag Action       VxLAN Action    MPLS Action
----------- ------------------- ------------------- ------------------ -----------
   Et1/1       strip               none                none            none
```

- On DCS-7280R3, DCS-7500R3, and DCS7800R3:

```
switch(config)# show interface tool tunnel
   Port         Configured         Status       Id Tag       Allowed Vlans    Timestamp
                Mode                                                         Mode
----------- ------------------ ------------ ------------ ------------------- -----------
   Et1/1       tool               tool         Off          100              None



   Port         VN Tag Action       BR Tag Action    MPLS Action
----------- ------------------- ------------------- -----------
   Et1/1       strip               none             none
```

20.1.11.3 Feature Interactions

On DCS-7280R/R2, DCS-7500R/R2, and DCS-7020R the following Tap aggregation features are supported with BR-E/VN tagged packets when tag stripping is enabled. Some or all of these features may not work for BR-E/VN tagged packets when tag stripping is not configured.

- MAC, IPv4, IPv6 Traffic Steering
- Tap identity tagging
- MPLS Termination
- Ingress VLAN Membership Filtering
- Egress VLAN Membership Filtering

On DCS-7280R3, DCS-7500R3, and DCS7800R3 the following Tap aggregation features are supported with BR-E/VN tagged packets when tag stripping is enabled.

- MAC, IPv4, IPv6 Traffic Steering
- Tap identity tagging

20.1.11.4 Limitations

The following lists feature limitations on DCS-7280R/R2, DCS-7500R/R2, and DCS-7020R when BR-E/VN tag stripping is enabled with tagged packets:

- IPv4/IPv6 traffic steering on VLAN IDs is not supported.
- IPv6 traffic steering is not supported on packets with two or more accompanying 802.1Q tags
- Ingress VLAN membership filtering is scale limited. Only ***4096*** entries, used for whitelisting or allowing specific port-VLAN pairs, are supported. The default case of allowing all VLANs does not consume

any of these entries. For example, a tap interface configured to allow ***4093*** VLANs will consume all but three entries.

- Ingress VLAN membership filtering on Tap interface LAGs for BR-E/VN tagged packets is not supported. To use Ingress VLAN membership for BR-E/VN tagged packets on a LAG Tap interface the Ingress VLAN membership configuration must be configured on each physical interface of the LAG.
- Time stamping and BR-E/VN tag stripping cannot be configured on the same tool interfaces. Time stamping, if configured, will take precedence over BR-E/VN tag stripping.
- 802.1Q Tag Stripping does not support BR-E/VN tagged packets.

The following lists feature limitations on DCS-7280R3, DCS-7500R3, and DCS7800R3 when BR-E/VN tag stripping is enabled with tagged packets:

- IPv4/IPv6/MAC traffic steering on VLAN IDs is not supported.
- MPLS pop and BR-E/VN tag stripping can not be used simultaneously on a tool port. If MPLS pop and BR-E/VN tag stripping are both configured on the same tool port, MPLS pop will take precedence and BR-E/VN tags will not be stripped.
- Ingress and Egress VLAN Membership filtering of BR-E/VN tagged packets is not supported in ***EOS-4.25.2F***.
- 802.1Q Tag Stripping does not support BR-E/VN tagged packets.

20.1.12 TAP Aggregation Commands

Global Configuration Commands

- hardware forwarding system-profile-tap-aggregation
- platform fm6000 keyframe
- platform fm6000 keyframe device
- platform fm6000 keyframe fields skew
- platform fm6000 keyframe rate
- platform fm6000 keyframe source
- platform sand multicast replication default
- platform sand multicast replication ingress maximum
- tap aggregation

Interface Configuration Commands

- mac timestamp
- switchport mpls pop all
- switchport tap allowed vlan
- switchport tap default group
- switchport tap identity
- switchport tap native vlan
- switchport tap truncation
- switchport tool allowed vlan
- switchport tool encapsulation
- switchport tool group
- switchport tool identity
- switchport tool truncation

Tap Aggregation Configuration Mode

- encapsulation
- mode (tap-agg configuration mode)
- mode exclusive no-errdisable (tap-agg configuration mode)

Tap Aggregation Traffic Steering

- class (policy-map (tapagg))
- class-map type tapagg
- match (class-map (tapagg))
- match (policy-map (tapagg))
- policy-map type tapagg
- resequence (class-map (tapagg))
- resequence (policy-map (tapagg))
- service-policy type tapagg (Interface mode)
- set (policy-map-class (tapagg))

Display Commands EXEC Mode

- show hardware tcam profile
- show interfaces tap
- show interfaces tool
- show platform fm6000 keyframe
- show platform sand mcast capacity
- show tap aggregation groups

20.1.12.1 class (policy-map (tapagg))

The **`class (policy-map (tapagg))`** command places the switch in the ***policy-map-class*** (TAPagg) configuration mode, which is a group-change mode that defines a TAP aggregation class by associating the class's eponymous class-map to a **`set`** statement. Upon exiting the policy-map-class mode, the class is placed in the policy-map as specified by an assigned sequence number.

A policy map is an ordered list of classes and match rules. Each class contains a class map, a **`set`** command, and a sequence number:

- The class map identifies a data stream by using an ordered list of ACLs. Class maps are configured in class-map (tapagg) configuration mode. Data packets are managed by commands of the highest priority class or rule that matches the packet.
- **`set`** commands specify the replication method of filtered data packets, either through an associated aggregation group or identity VLAN tagging.
- Sequence numbers specify the class's priority within the policy map. Lower sequence numbers denote higher priority.

The **`exit`** command returns the switch to policy-map configuration mode. However, saving policymap-class changes also requires an exit from policy-map mode. This saves all pending policy map and policy-map-class changes to ***running-config*** and returns the switch to global configuration mode. The **`abort`** command discards pending changes and returns the switch to global configuration mode.

The **`no class`** and **`default class`** commands remove the class assignment from the configuration mode policy map by deleting the corresponding class configuration from ***running-config***.

Command Mode

Policy-Map (tapagg) Configuration accessed through **`class (policy-map (tapagg))`**

Command Syntax

[**SEQ_NUM**] **`class`** ***class_name***

`default` [**SEQ_NUM**] **`class`** ***class_name***

`no` [**SEQ_NUM**] **`class`** ***class_name***

Parameters

- **SEQ_NUM** priority of the class within the policy map. Lower numbers denote higher priority.
  - ***no parameter*** number is derived by adding 10 to number of the map's last class or rule.
  - ***1 to 4294967295*** number assigned to class.
- ***class_name*** name of the class.

Guidelines

When a class is not associated with a **set (policy-map-class (tapagg))** command, the filtered traffic is managed as specified by the TAP port's default aggregation group.

Commands Available in Policy-map-class (tapagg) Configuration Mode

- **set (policy-map-class (tapagg))** assigns VLAN identity tag or tap aggregation group to class.
- **`exit`** returns the switch to parent policy map configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

- **class (policy-map (tapagg))** places the switch in the ***policy-map*** (tapagg) configuration mode.
- **match (policy-map (tapagg))** assigns a match rule to a TAP aggregation policy map.

Examples

- These commands place the switch in ***policy-map-class*** and add the ***t-class_1*** class map to the ***tpolicy_1*** policy map. Packets filtered by the class map are identity tagged with ***VLAN 444***.

```
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# class t-class_1
switch(config-pmap-c-t-policy_1-t-class_1)# set id-tag 444
switch(config-pmap-c-t-policy_1-t-class_1)# exit
switch(config-pmap-t-policy_1)# exit
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# show active
policy-map type tapagg t-policy_1
10 class t-class_1
set id-tag 444
switch(config-pmap-t-policy_1)#
```

- These commands place the switch in policy-map-class to add the ***t-class_1*** class map to the ***t-policy_1*** policy map. The first, second, or both of the two outer-most VLAN tags are stripped.

```
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# class t-class_1
switch(config-pmap-t-policy_1t-class_1)# set aggregation-group t-group
 remove dot1q outer 1-2
switch(config-pmap-c-t-policy_1-t-class_1)# set aggregation-group t-group
 id-tag 10
switch(config-pmap-c-t-policy_1-t-class_1)# set id-tag 10 remove dot1q outer
 1
switch(config-pmap-c-t-policy_1-t-class_1)# set aggregation-group t-group
switch(config-pmap-c-t-policy_1-t-class_1)# set id-tag 10
switch(config-pmap-c-t-policy_1-t-class_1)# set aggregation-group t-group
 id-tag 10 remove dot1q outer 1-2
```

20.1.12.2 class-map type tapagg

The **`class-map type tapagg`** command places the switch in class-map (tapagg) configuration mode, which is a group change mode that modifies a tapagg class map. A tapagg class map is a data structure that uses Access Control Lists (ACLs) to define a data stream by specifying characteristics of data packets that comprise the stream. Tapagg policy maps use class maps to specify traffic that is managed by policy map criteria.

The **exit** command saves pending class map changes to ***running-config***, then returns the switch to the ***global*** configuration mode. Class map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes and returns the switch to global configuration mode.

The **no class-map type tapagg** and **default class-map type tapagg** commands delete the specified class map by removing the corresponding class-map type qos command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

class-map type tapagg match-any ***class_name***

no class-map type tapagg match-any ***class_name***

default class-map type tapagg match-any ***class_name***

Parameters

class_name name of class map.

Commands Available in Class-Map (tapagg) Configuration Mode

- **match (class-map (tapagg))**
- **resequence (class-map (tapagg))**

Related Commands

class (policy-map (tapagg))

Example

This command creates a TAP aggregation class map named ***t-class_1*** and places the switch in the ***class-map*** configuration mode.

```
switch(config)# class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)#
```

20.1.12.3 encapsulation

The **encapsulation** command is configured under Tap Aggregation mode to enable the 802.1br-E/VN Tag Stripping the packets ingressing any tap port on DCS-7280R/R2, DCS-7500R/R2, and DCS-7020R.

The **no encapsulation** or **default encapsulation** commands disable the 802.1br-E/VN Tag Stripping configuration from the **running-config**.

Command Mode

Tap Aggregation Configuration Mode

Command Syntax

encapsulation {dot1br | vn-tag} strip

no encapsulation {dot1br | vn-tag} strip

default encapsulation {dot1br | vn-tag} strip

Parameters

- **dot1br** strips the 802.1br-E headers.
- **vn-tag** strips the VN Tag headers.

Examples

- To enable BR E-Tag stripping add the following configuration:

```
switch(config)# tap aggregation
switch(config-tap-agg)# encapsulation dot1br strip
```

- To enable VN-TAG stripping add the following configuration:

```
switch(config)# tap aggregation
switch(config-tap-agg)# encapsulation vn-tag strip
```

20.1.12.4 hardware forwarding system-profile-tap-aggregation

The **hardware forwarding system-profile-tap-aggregation** command sets the global hardware forwarding system profile on a switch.

The **hardware forwarding system-profile-default** command returns the switch to the default system profile configurations when not in Tap Aggregation mode.

Command Mode

Global Configuration Mode

Command Syntax

hardware forwarding system profile [system-profile-default | system-profile-tap-aggregation | module]

default hardware forwarding system profile [system-profile-default | system-profile-tap-aggregation | module]

Parameters

- **system-profile-default** normal forwarding profile.
- **system-profile-tap-aggregation** tap aggregation features profile.
- **module** used to set the tap aggregation features profile on a modular system.

Examples

- This command configures the tap aggregation features profile.

```
switch(config)# hardware forwarding system profile system-profile-tap-
aggregation
```

- This command returns the to the default system profile configurations when not in Tap Aggregation mode.

```
switch(config)# hardware forwarding system profile system-profile-default
```

- This command overrides the system default profile on a per line card basis for a modular system.

```
switch(config)# hardware forwarding module Linecard3 profile system-profile-
tap-aggregation
```

20.1.12.5 mac timestamp

The **mac timestamp** command enables timestamping on the configuration mode interface.

When timestamping is enabled on an egress interface, packets leave the interface with timestamps that were applied in hardware upon arriving at the switch. This is facilitated by applying a hardware timestamp to all frames arriving on all interfaces when timestamping is enabled on any interface, then removing timestamps on packets egressing interfaces where timestamping is not enabled.

The switch supports two timestamp modes, which differ in managing the egress frame's 32-bit Frame Check Sequence (FCS):

- **before-fcs**: the switch discards the original FCS, appends the ingress timestamp at the end of the frame data, recalculates a new FCS based on the appended timestamp, then appends the new FCS to the end of the frame. This creates a valid Ethernet frame but does not update headers of any nested protocols.
- **replace-fcs**: the switch replaces the original FCS with the timestamp. This mode maintains the size of the original frame without any latency impact, but the FCS is not valid.

The **`no mac timestamp`** and **`default mac timestamp`** commands restore the default behavior of disabling timestamping on the configuration mode interface by removing the corresponding **`mac timestamp`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

`mac timestamp` **TS_PROPERTY**

Parameters

- **TS_PROPERTY** specifies the timestamp insertion mode. Options include:
  - **before-fcs** the ingress timestamp is appended to the frame and the FCS is recalculated.
  - **replace-fcs** the ingress timestamp replaces the original FCS.

Examples

- These commands enable timestamping in **before-fcs** mode on ***interface ethernet 44***.

```
switch(config)# interface ethernet 44
switch(config-if-Et44)# mac timestamp before-fcs
switch(config-if-Et44)# show active
interface Ethernet44
   mac timestamp before-fcs
switch(config-if-Et44)#
```

- These commands disable timestamping on ***interface ethernet 44***.

```
switch(config-if-Et44)# no mac timestamp
switch(config-if-Et44)# show active
interface Ethernet44
switch(config-if-Et44)#
```

20.1.12.6 match (class-map (tapagg))

The **`match`** command adds an ACL to the configuration-mode class map and associates a sequence number to the ACL. A class map is an ordered list of ACLs that define a data stream; the sequence number specifies an ACL's priority within the list. A class map is used by policy maps to filter data packets. Tapagg class maps utilize ACL permit rules to pass packets and deny rules to drop packets.

Class map (tapagg) configuration mode is a group change mode. **`Match`** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **`no match`** and **`default match`** commands remove the specified **`match`** statement from the configuration-mode class map by deleting the corresponding **`match`** command from ***running-config***.

Command Mode

Class-map (tagagg) Configuration accessed through **class-map type tapagg** command.

Command Syntax

[**SEQ_NUM**] **`match ip access-group`** ***list_name***

`no` **SEQ_NUM**] **`match ip access-group`** ***list_name***

`default` SEQ_NUM] **`match ip access-group`** ***list_name***

Parameters

- **SEQ_NUM** sequence number assigned to the ACL. Options include:
  - ***no parameter*** number is derived by adding 10 to the number of the map's last ACL.
  - **`1`** **to** ***4294967295*** number assigned to ACL.
- ***list_name*** name of ACL assigned to class map.

Guidelines

`match` statements accept IPv4 ACLs.

Related Commands

- **class-map type tapagg** places the switch in Class-Map configuration mode.
- **`exit`** saves pending class map changes, then returns the switch to global configuration mode.
- **`abort`** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (tapagg))** assigns a class map to a policy map.

Example

These commands add two IPv4 ACLs (***tacl-1*** and ***tacl-2***) to the ***t-class_1*** class map, save the command by exiting class-map mode, and re-enter the mode to display the added ACLs.

```
switch(config)# class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)# match ip access-group tacl-1
switch(config-cmap-t-class_1)# match ip access-group tacl-2
switch(config-cmap-t-class_1)# exit
switch(config)# class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)# show active
 class-map type tapagg match-any t-class_1
      10 match ip access-group tacl-1
      20 match ip access-group tacl-2
switch(config-cmap-t-class_1)#
```

20.1.12.7 match (policy-map (tapagg))

The **`match`** command adds a rule to the configuration-mode TAP aggregation policy map. A policy map is an ordered list of classes and rules. Each rule contains a filter list, an action, and a sequence number:

- The filter list identifies a data stream through a set of packet field values.
- The action, (**SET_VALUE** parameter) specifies the replication method of filtered data packets, either through an associated aggregation group or identity VLAN tagging.
- The sequence number specifies the rule's priority within the policy map.

The **`no match`** and **`default match`** commands remove the **`match`** rule from the configuration-mode policy by deleting the corresponding statement from ***running-config***.

Command Mode

Policy-Map (tapagg) Configuration accessed through **class (policy-map (tapagg))**.

Command Syntax

[**SEQ_NUM**] **`match`** [**VLAN_TAG**] **SOURCE_ADDR** [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**] [**PROTOCOL**] [**FLAGS**] [**MESSAGE**] [**fragments**] [**tracked**] [**DSCP_FILTER**] [**TTL_FILTER**] [**log**] **SET_VALUE**

`no match` [**VLAN_TAG**] **SOURCE_ADDR** [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**] [**PROTOCOL**] [**FLAGS**][**MESSAGE**] [fragments][tracked] [**DSCP_FILTER**][**TTL_FILTER**] [log] **SET_VALUE**

`default match` [**VLAN_TAG**] **SOURCE_ADDR** [**SOURCE_PORT**] **DEST_ADDR** [**DEST_PORT**] [**PROTOCOL**][**FLAGS**] [**MESSAGE**][fragments] [tracked][**DSCP_FILTER**] [**TTL_FILTER**][log] **SET_VALUE**

Note: Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a permit rule.

Parameters

- **SEQ_NUM** priority of the rule within the policy map. Lower numbers denote higher priority.
  - ***no parameter*** number derived by adding 10 to number of the map's last class or rule.
  - ***1*** to ***4294967295*** number assigned to class.
- **VLAN_TAG** VLAN field filter. Options include:
  - ***no parameter*** packets are not filtered by VLAN field.
  - **vlan *1*** to ***4094 0*** to ***4095*** VLAN ID and mask.
  - **vlan inner *1*** to ***4094 0*** to ***4095*** VLAN ID and mask.
  - **vlan *1*** to ***4094 0*** to ***4095*** **inner *1*** to ***4094 0*** to ***4095*** VLAN ID and mask.
- **PROTOCOL** protocol field filter. Values include:
  - ***no parameter*** packets are not filtered by host name.
  - **ahp** authentication header protocol (51).
  - **icmp** internet control message protocol (1).
  - **igmp** internet group management protocol (2).
  - **ip** internet protocol IPv4 (4).
  - **ospf** open shortest path first (89).
  - **pim** protocol independent multicast (103).
  - **tcp** transmission control protocol (6).
  - **udp** user datagram protocol (17).
  - **vrrp** virtual router redundancy protocol (112).
  - ***protocol_num*** integer corresponding to an IP protocol. Values range from ***0*** to ***255***.
- **SOURCE_ADDR** and **DEST_ADDR** source and destination address filters. Options include:
  - ***network_addr*** subnet address (CIDR or address-mask).
  - **any** packets from all addresses are filtered.
  - **host *ip_addr*** IP address (dotted decimal notation).

  Source and destination subnet addresses support discontiguous masks.
- **SOURCE_PORT** and **DEST_PORT** source and destination port filters. Options include:
  - **any** all ports.
  - **eq *port-1 port-2*** ... ***port-n*** a list of ports. Maximum list size is ***10*** ports.
  - **neq *port-1 port-2*** ... ***port-n*** the set of all ports not listed. Maximum list size is ***10*** ports.
  - **gt *port*** the set of ports with larger numbers than the listed port.
  - **lt *port*** the set of ports with smaller numbers than the listed port.
  - **range *port_1 port_2*** the set of ports whose numbers are between the range.
- **fragments** filters packets with FO bit set (indicates a non-initial fragment packet).
- **FLAGS** flag bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- **MESSAGE** message type filters (ICMP packets). Use CLI syntax assistance (?) to display options.
- **tracked** rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- **DSCP_FILTER** rule filters packet by its DSCP value. Values include:
  - ***no parameter*** rule does not use DSCP to filter packets.
  - **dscp *dscp_value*** packets match if DSCP field in packet is equal to ***dscp_value***.

- **TTL_FILTER** rule filters packet by its TTL (time-to-live) value. Values include:
  - ***no parameter*** rule does not use TTL field to filter packets.
  - **ttl eq *ttl_value*** packets match if ttle in packet is equal to ***ttl_value***.
  - **ttl gt *ttl_value*** packets match if ttl in packet is greater than ***ttl_value***.
  - **ttl lt *ttl_value*** packets match if ttl in packet is less than ***ttl_value***.
  - **ttl neq *ttl_value*** packets match if ttl in packet is not equal to ***ttl_value***.
- **log** triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- **SET_VALUE** specifies the replication method for filtered packets.
  - **set aggregation group *agg_group*** peplication specified by aggregation group.
  - **set id-tag *1*** to ***4094*** packet is identity tagged with specified VLAN number.
  - **set aggregation group *agg_group*** id-tag ***1*** to ***4094*** assigns agg group and identity tag.

Related Commands

- policy-map type tapagg places the switch in the ***policy-map*** (tapagg) configuration mode.
- class (policy-map (tapagg)) assigns a class to the configuration-mode policy map.

Example

This command creates a match rule for the ***t-policy_1*** policy map that filters OSPF packets and replicates them as specified by the ***t-group*** tap aggregation group.

```
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# match ip ospf any any set aggregation-group t-
group
switch(config-pmap-t-policy_1)# exit
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# show active
 policy-map type tapagg t-policy_1
   10 match ip ospf any any set aggregation-group t-group
switch(config-pmap-t-policy_1)#
```

20.1.12.8 mode (tap-agg configuration mode)

The **mode** command configures the switch's TAP aggregation mode. The **mode exclusive** command enables TAP aggregation. When TAP aggregation is enabled, TAP and tool ports are enabled, switching mode is disabled, and switching ports are errdisabled. TAP aggregation is disabled by default.

The **no mode** and **default mode** commands disable TAP aggregation mode and enable switching mode by removing the **mode** command from ***running-config***.

Command Mode

TAP Aggregation Configuration

Command Syntax

mode exclusive

no mode exclusive

default mode exclusive

Parameters

exclusive TAP aggregation is enabled.

Related Command

tap aggregation places the switch in the ***TAP-aggregation*** configuration mode.

Examples

- These commands place the switch in ***tap aggregation*** configuration mode, enable TAP aggregation mode, and display the results.

```
switch(config)# tap aggregation
switch(config-tap-agg)# mode exclusive
switch(config-tap-agg)# show active
tap aggregation
   mode exclusive
switch(config-tap-agg)#
```

- These commands disable ***tap aggregation*** mode by removing the **mode** command from ***running-config***, then display the results.

```
switch(config)# tap aggregation
switch(config-tap-agg)# no mode
switch(config-tap-agg)# show active
switch(config-tap-agg)#
```

20.1.12.9 mode exclusive no-errdisable (tap-agg configuration mode)

The **mode exclusive no-errdisable** command configures the specified interface to remain enabled, regardless of its switchport mode, when TAP aggregation is enabled. This command is used primarily to configure a port to support PTP functions while the switch operates as a TAP aggregator.

Each command configures one Ethernet or port-channel interface. Subsequent **mode exclusive no-errdisable** commands add to the list of ports that remain enabled when TAP aggregation is enabled.

The **no mode exclusive no-errdisable** and **default mode exclusive no-errdisable** commands configure the specified interface to be error-disabled when programmed in access, trunk, or dot1q-tunnel switching mode (when TAP aggregation is enabled) by removing the corresponding **mode exclusive no-errdisable** command from ***running-config***.

Command Mode

TAP Aggregation Configuration

Command Syntax

mode exclusive no-errdisable **INT_NAME**

Parameters

INT_NAME interface type and number. Options include:

- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **port-channel *p_num*** port-channel interface specified by ***p_num***.

Related Commands

- **tap aggregation** places the switch in ***TAP-aggregation*** configuration mode.
- **mode (tap-agg configuration mode)** configures the switch's TAP-aggregation mode.

Guidelines

In order for a TAP-aggregation switch to receive PTP traffic, the upstream device to which it is connected should be set to statically send PTP multicast traffic to the connected port on the switch.

Since IGMP snooping is disabled on TAP-aggregation switches and with no configuration to support sending upstream join messages in such a state, the messages are transmitted statically from the upstream device. Once the upstream messages are received, the port will move to the slave state and follow the standard PTP mechanism.

Example

These commands place the switch in TAP-aggregation configuration mode and place ***interface ethernet 21/3*** in no-errdisable mode.

```
switch(config)# tap aggregation
switch(config-tap-agg)# mode exclusive
switch(config-tap-agg)# mode exclusive no-errdisable ethernet 21/4
switch(config-tap-agg)#
```

20.1.12.10 platform fm6000 keyframe

The **`platform fm6000 keyframe`** command enables keyframe generation for data streams transmitted from specified ethernet interfaces. Keyframes are routable IP packets that the switch inserts into a data stream to provide contextual information that correlate timestamps inserted into data packets with absolute UTC time and the switch's complete ASIC time counter.

The switch supports a maximum of ten keyframes. The keyframe name is the label that distinguishes different keyframes. Each keyframe can egress from every ethernet port. Command options specify the destination MAC address and IP address in the keyframe's header. Other keyframe commands specify the transmission rate and the frame's source.

The **`no platform fm6000 keyframe and default platform fm6000 keyframe`** commands disable generation of the specified keyframe by deleting the corresponding platform fm6000 keyframe command from ***running-config***. These command also remove all supporting **`platform fm6000 keyframe`** commands for the specified keyframe.

Command Mode

Global Configuration

Command Syntax

`platform fm6000 keyframe` ***kf_name*** interface ethernet ***e_range ipv4_addr mac_addr***

`no platform fm6000 keyframe` ***kf_name***

`default platform fm6000 keyframe` ***kf_name***

Parameters

- ***kf_name*** the keyframe's name.
- ***e_range*** Ethernet interface range over which the keyframe egresses. Valid formats include number, range, or comma-delimited list of numbers and ranges.
- ***ipv4_addr*** destination IPv4 address inserted into keyframes (dotted decimal notation).
- ***mac_addr*** destination MAC address inserted into keyframes (48-bit dotted hex notation).

Guidelines

Subsequent issuance of this command for a specified keyframe replaces the existing command in ***running-config***. Ethernet interfaces are inserted into an existing keyframe only by issuing the complete command that identifies all interfaces through which the keyframe is transmitted.

Example

This command enables the generation of a keyframe named ***key-1***. This keyframe egresses from Ethernet interfaces ***11*** through ***15*** and specifies a source IP address of ***10.21.1.4*** and a MAC address of ***10.4E21.9F11***.

```
switch(config)# platform fm6000 keyframe key-1 interface ethernet 11-15
 10.21.1.4 10.4E21.9F11
switch(config)#
```

20.1.12.11 platform fm6000 keyframe device

The **platform fm6000 keyframe device** command configures the 16-bit number that the specified keyframe lists as the device ID in its payload. By default, the device value placed in the specified keyframes is ***0***.

The **no platform fm6000 keyframe device** and **default platform fm6000 keyframe device** commands restore the default device ID insertion value of ***0*** for the specified keyframe by removing the corresponding **platform fm6000 keyframe device** command from ***running-config***. The **no platform fm6000 keyframe and default platform fm6000 keyframe** command also removes the corresponding **platform fm6000 keyframe device** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

platform fm6000 keyframe ***kf_name*** **device** ***device_id***

no platform fm6000 keyframe ***kf_name*** **device**

default platform fm6000 keyframe ***kf_name*** **device**

Parameters

- ***kf_name*** keyframe name.
- ***device_id*** value inserted in keyframe's device ID field. Values range from ***0*** to ***65535***. Default is ***0***.

Example

These commands enable the generation of a keyframe named ***key-1***, then configure ***100*** as the value that is placed in the keyframe's device ID field.

```
switch(config)# platform fm6000 keyframe key-1 interface ethernet 11-15
 10.21.1.4 10.4E21.9F11
switch(config)# platform fm6000 keyframe key-1 device 100
switch(config)#
```

20.1.12.12 platform fm6000 keyframe fields skew

Keyframes may optionally include skew numerator and skew denominator fields. These skew fields form a ratio indicating the ASIC clock skew. If the ratio is greater than ***1***, the clock is skewed fast; if the ratio is less than ***1***, the clock is skewed slow. Clock skew fields are omitted by default.

The **platform fm6000 keyframe fields skew** command enables the inclusion of clock skew fields in the keyframe.

The **no platform fm6000 keyframe fields skew** and **default platform fm6000 keyframe fields skew** commands remove the clock skew fields from the keyframe.

Command Mode

Global Configuration

Command Syntax

platform fm6000 keyframe ***kf_name*** **fields skew**

Parameter

kf_name keyframe name.

Example

This command enables the inclusion of clock skew fields in the keyframe named ***key-1***.

```
switch(config)# platform fm6000 keyframe key-1 fields skew
switch(config)#
```

20.1.12.13 platform fm6000 keyframe rate

The **platform fm6000 keyframe rate** command specifies the transmission rate for the specified keyframe from each interface from which it is configured to egress. By default, one keyframe is sent per second.

The **no platform fm6000 keyframe rate** and **default platform fm6000 keyframe rate** commands restore the default transmission rate for the specified keyframe of one per second by removing the corresponding **platform fm6000 keyframe rate** command from ***running-config***. The **no platform fm6000 keyframe and default platform fm6000 keyframe** command also removes the corresponding **platform fm6000 keyframe rate** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

platform fm6000 keyframe ***kf_name*** **rate** ***tx_rate***

Parameters

- ***kf_name*** the keyframe's name.
- ***tx_rate*** keyframe transmission rate (frames per second). Values range from ***1*** to ***100***. Default value is ***1***.

Example

These commands enable the generation of a keyframe named ***key-1***, then configure the generation rate for the keyframe of ***10*** frames per second on each of the five interfaces that it is configured to egress.

```
switch(config)# platform fm6000 keyframe key-1 interface ethernet 11-15
 10.21.1.4 10.4E21.9F11
switch(config)# platform fm6000 keyframe key-1 rate 10
switch(config)#
```

20.1.12.14 platform fm6000 keyframe source

The **platform fm6000 keyframe source** command configures the source IP address that the specified keyframe lists in its IP header. By default, keyframes use the IP address of the management interface as their source address.

The **no platform fm6000 keyframe source** and **default platform fm6000 keyframe source** commands restore the management interface IP address as the specified keyframe's source IP address by removing the corresponding **platform fm6000 keyframe source** command from ***running-config***. The **no platform fm6000 keyframe and default platform fm6000 keyframe** command also removes the corresponding **platform fm6000 keyframe source** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

platform fm6000 keyframe ***kf_name*** **source ip** ***ipv4_addr***

no platform fm6000 keyframe*kf_name* source ip

default platform fm6000 keyframe ***kf_name*** **source ip**

Parameters

- ***kf_name*** keyframe's name.
- ***ipv4_addr*** keyframe's source IPv4 address (dotted decimal notation).

Example

These commands enable the generation of a keyframe named ***key-1***, then sets the keyframe source IP address to ***10.1.1.101***.

```
switch(config)# platform fm6000 keyframe key-1 interface ethernet 11-15
 10.21.1.4 10.4E21.9F11
switch(config)# platform fm6000 keyframe key-1 source 10.1.1.101
switch(config)#
```

20.1.12.15 platform sand multicast replication default

The **platform sand multicast replication default** command configures the default replication mode on Sand platform switches. The factory default replication mode differs in various scenarios as follows:

- The default replication mode on switches with fabric is **fabric-egress** mode.
- The default replication mode on switches with single Fabric Access Processor (FAP) systems is **ingress** mode.
- The default replication mode on switches without fabric barring single FAP systems is **ingress-egress** mode.
- If a tool group with less than ***60*** LAGs has at least one hardware LAG, then the default replication mode of the tool group is **ingress-only** mode. Else the default replication mode of the tool group is the one configured across all LAGs in the tool group.

The **default platform sand multicast replication default** and **no platform sand multicast replication default** commands revert the current state to the factory default behavior.

Command Mode

Global Configuration

Command Syntax

platform sand multicast replication default {fabric-egress | ingress}

no platform sand multicast replication default

default platform sand multicast replication default

Parameters

- **fabric-egress** configures the replication mode to use fabric-egress VoQ buffers.
- **ingress** configures the replication mode to use ingress VoQ buffers.

Guidelines

This command is supported on Sand platforms only.

Related Commands

- **platform sand multicast replication ingress maximum**
- **show platform sand mcast capacity**

Example

This command configures the default replication mode to ingress.

```
switch(config)# platform sand multicast replication default ingress
switch(config)#
```

20.1.12.16 platform sand multicast replication ingress maximum

The **platform sand multicast replication ingress maximum** command configures maximum members for ingress-only replication.

The **default platform sand multicast replication ingress maximum** command reverts the maximum members for ingress-only replication to the default value.

The **no platform sand multicast replication ingress maximum** command deletes the maximum member value for ingress-only replication.

Command Mode

Global Configuration

Command Syntax

platform sand multicast replication ingress maximum *max_value*

no platform sand multicast replication ingress maximum

default platform sand multicast replication ingress maximum

Parameters

max_value specifies the maximum number of members for ingress-only replication. Values range from ***1*** to ***64***. The default value is ***64***.



Note: ***max_value*** for a single FAP Jericho system ranges from ***1*** to ***4096***.

Guidelines

This command is supported on Sand platforms only.

Related Commands

- **platform sand multicast replication default**
- **show platform sand mcast capacity**

Example

This command specifies a maximum of ***63*** members for ingress-only replication.

```
switch(config)# platform sand multicast replication ingress maximum 63
switch(config)#
```

20.1.12.17 policy-map type tapagg

The **policy-map type tapagg** command places the switch in policy-map (tapagg) configuration mode, which is a group-change mode that modifies a TAP-aggregation policy map. A TAP-aggregation policy map is a data structure that consists of class maps and match statements that filter a specific data stream. Packets in that data stream are either managed as specified by a TAP aggregation group or modified to add a VLAN identity tag. Policy maps manage traffic when applied to an Ethernet or port-channel interface.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to global configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no policy-map type tapagg** and **default policy-map type tapagg** commands delete the specified policy map by removing the corresponding **policy-map type tapagg** command and the associated policy map statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`policy-map type tapagg` ***map_name***

`no policy-map type tapagg` ***map_name***

`default policy-map type tapagg` ***map_name***

Parameters

map_name name of policy map.

Commands Available in Policy-Map Configuration Mode

- class (policy-map (tapagg))
- match (policy-map (tapagg))

Related Commands

- class-map type tapagg
- service-policy type tapagg (Interface mode)

Example

This command creates the TAP-aggregation policy map named ***t-policy_1*** and places the switch in the ***policy-map*** configuration mode.

```
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#
```

20.1.12.18 resequence (class-map (tapagg))

The **`resequence`** command assigns sequence numbers to access control lists (ACLs) in the configuration mode TAP-aggregation class map. Sequence numbers denote an ACL's priority within the class map. Command parameters specify the number of the first ACL and the numeric interval between consecutive ACLs.

Maximum rule sequence number is ***4294967295***.

Command Mode

Class-map (tagagg) Configuration

accessed with the **class-map type tapagg** command

Command Syntax

`resequence` [***start_num*** [***inc_num***]]

Parameters

- ***start_num*** sequence number assigned to the first rule. Default is ***1***.
- ***inc_num*** numeric interval between consecutive rules. Default is ***1***.

Example

These commands display a policy map whose entities were entered with default sequence numbers, then renumber the contents.

```
switch(config-pmap-t-policy_1)# show active
 policy-map type tapagg t-policy_1
   10 match ip ospf any any set aggregation-group t-group
   20 class fred
      set aggregation-group t-group id-tag 444
   30 class t-class_2
      set id-tag 500
   40 class t-class_3
```

```
      set id-tag 600
   50 class t-class_4
      set id-tag 700
switch(config-pmap-t-policy_1)# resequence 100 20
switch(config-pmap-t-policy_1)# exit
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# show active
 policy-map type tapagg t-policy_1
   100 match ip ospf any any set aggregation-group t-group
   120 class fred
      set aggregation-group t-group id-tag 444
   140 class t-class_2
      set id-tag 500
   160 class t-class_3
      set id-tag 600
   180 class t-class_4
      set id-tag 700
switch(config-pmap-t-policy_1)#
```

20.1.12.19 resequence (policy-map (tapagg))

The **resequence** command assigns sequence numbers to classes and rules in the configuration mode TAP-aggregation policy map. Sequence numbers denote the priority of a class or rule within the policy map. Command parameters specify the number of the first policy map entity and the numeric interval between consecutive entities.

Maximum rule sequence number is ***4294967295***.

Command Mode

Policy-Map (tapagg) Configuration

accessed with the **class (policy-map (tapagg))** command

Command Syntax

resequence [***start_num*** [***inc_num***]]

Parameters

- ***start_num*** sequence number assigned to the first rule. Default is ***1***.
- ***inc_num*** numeric interval between consecutive rules. Default is ***1***.

Example

These commands display a policy map whose entities were entered with default sequence numbers, then use the **resequence** command to renumber the contents.

```
switch(config-pmap-t-policy_1)# show active
 policy-map type tapagg t-policy_1
   10 match ip ospf any any set aggregation-group t-group
   20 class fred
      set aggregation-group t-group id-tag 444
   30 class t-class_2
      set id-tag 500
   40 class t-class_3
      set id-tag 600
   50 class t-class_4
      set id-tag 700
switch(config-pmap-t-policy_1)# resequence 100 20
switch(config-pmap-t-policy_1)# exit
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# show active
 policy-map type tapagg t-policy_1
```

```
   100 match ip ospf any any set aggregation-group t-group
   120 class fred
      set aggregation-group t-group id-tag 444
   140 class t-class_2
      set id-tag 500
   160 class t-class_3
      set id-tag 600
   180 class t-class_4
      set id-tag 700
switch(config-pmap-t-policy_1)#
```

20.1.12.20 service-policy type tapagg (Interface mode)

The **service-policy type tapagg** command applies a specified TAP-aggregation policy map to the configuration-mode interface. A policy map is a data structure that identifies data traffic through class maps and match rules, then specifies the method of replicating the traffic. This command is active only when TAP aggregation mode is enabled on the switch.

The **no service-policy type tapagg** and **default service-policy type tapagg** commands remove the policy map assignment from the configuration mode interface by deleting the corresponding **service-policy tapagg** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

service-policy type tapagg input ***policymap_name***

Parameters

- **input** policy map applies to inbound packet streams. This is the only option.
- ***map_name*** name of policy map.

Guidelines

A policy map that is attached to a port-channel interface takes precedence for member interfaces of the port channel over their individual Ethernet interface configuration. Members that are removed from a port channel revert to the policy-map implementation specified by its Ethernet interface configuration.

Related Commands

class (policy-map (tapagg)) places the switch into the ***policy-map*** configuration mode to create a policy map.

Example

These commands apply the ***t-policy_1*** policy map to ***interface ethernet 17***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# service-policy type tapagg input t-policy_1
switch(config-if-Et17)#
```

20.1.12.21 set (policy-map-class (tapagg))

The **set** command specifies the data replication method for traffic filtered by the associated class map in the configuration-mode policy map. The **set** command specifies one of these replication actions for filtered data packets:

- specifies an aggregation group.
- specifies a VLAN identity tag for replicated packets.

- specifies an aggregation group and a VLAN identity tag.

The **no set** and **default set** commands remove the specified set command data action from the configuration-mode class by deleting the associated **set** command from ***running-config***.

Command Mode

Policy-map-class (tapagg) Configuration

accessed using the **class (policy-map (tapagg))** command

Command Syntax

set SET_VALUE

no set SET_VALUE

default set SET_VALUE

Parameters

SET_VALUE specifies the replication method for filtered packets. Options include:

- **aggregation group *agg_group*** replication specified by aggregation group.
- **id-tag *VLAN_number*** packet is identity tagged with specified VLAN number. VLAN numbers range from ***1*** to ***4094***.
- **aggregation group *agg_group* id-tag *VLAN_number*** assigns aggregation group and identity tag (VLAN number). VLAN numbers range from ***1*** to ***4094***.

Related Commands

- **policy-map type tapagg** places the switch in the ***policy-map*** (tapagg) configuration mode.
- **class (policy-map (tapagg))** assigns a class to the ***policy-map*** configuration mode.
- **match (policy-map (tapagg))** assigns a rule to the ***policy-map*** configuration mode.

Guidelines

When a class is not associated with a **set** command, the filtered traffic is managed as specified by the TAP port's default aggregation group.

Example

These commands place the switch in policy-map-class to add the ***t-class_1*** class map to the ***t-policy_1*** policy map. Packets filtered by the class map are identity tagged with ***vlan 444*** and replicated as specified through the ***t-group*** aggregation group.

```
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# class t-class_1
switch(config-pmap-c-t-policy_1-t-class_1)# set aggregation-group t-group id-
tag 444
switch(config-pmap-c-t-policy_1-t-class_1)# exit
switch(config-pmap-t-policy_1)# exit
switch(config)# policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)# show active
 policy-map type tapagg t-policy_1
   10 class t-class_1
      set aggregation-group t-group id-tag 444
switch(config-pmap-t-policy_1)#
```

20.1.12.22 show hardware tcam profile

Use the **show hardware tcam profile** command to display The TCAM prifile by configuration and status.

Command Mode

EXEC

Command Syntax

show hardware tcam profile detail

Parameter

detail Displays TCAM profile details.

Example

```
(config-tap-agg)# show hardware tcam profile
                     Configuration            Status
FixedSystem          foo*                     foo
* configuration overridden by TapAgg
```

20.1.12.23 show interfaces tap

The **show interfaces tap** command displays TAP-port configuration information for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [**INTERFACE**] **tap** [**INFO_LEVEL**]

Parameters

- **INTERFACE** interface type and numbers. Options include:
  - ***no parameter*** all interfaces.
  - **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
  - **management *m_range*** management interface range specified by ***m_range***.
  - **port-channel *p_range*** port-channel interface range specified by ***p_range***.
  - Valid ***e_range***, ***m_range***, and ***p_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **INFO_LEVEL** amount of information that is displayed. Options include:
  - ***no parameter*** command displays table that summarizes TAP data.
  - **detail** command displays TAP data summary table and a list of ACLS applied to TAP ports.

Examples

- This command displays TAP-port configuration information for interface ethernet ***36*** through ***40***.

```
switch# show interface ethernet 31-35 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et31      tap            tap            301      31   0          tag_1
Et32      tap            tap            1        132  0          tag_1
Et33      tap            tap            303      233  0          tag_1
Et34      tap            tap            1        334  0          tag_3
Et35      tap            tap            1        345  0          tag_3
 switch#
```

- This command displays detailed TAP-port configuration information for ***interface ethernet 17***.

```
switch# show interface ethernet 17 tap detail
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
```

```
-------------------------------------------------------------------------
Et31      tap            tap            301      31   0          tag_1


Port      ACLs Applied
-------------------------------------------------------------------
switch#
```

20.1.12.24 show interfaces tool

The **`show interfaces tool`** command displays tool port configuration information for the specified interfaces.

Command Mode

EXEC

Command Syntax

`show interfaces` [**INTERFACE**] **`tool`**

Parameters

INTERFACE interface type and numbers. Options include:

- ***no parameter*** all interfaces.
- **ethernet *e_range*** Ethernet interface range specified by ***e_range***.
- **management *m_range*** management interface range specified by ***m_range***.
- **port-channel *p_range*** port-channel interface range specified by ***p_range***.

  Valid ***e_range***, ***m_range***, and ***p_range*** formats include number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays tool port configuration information for interface ethernet ***36*** through ***40***.

```
switch# show interface ethernet 36-40 tool
Port      Configured     Status         Allowed             Id   Timestamp
          Mode                          Vlans               Tag  Mode
-------------------------------------------------------------------------
Et36      tool           tool           201-205             Off  None
Et37      tool           tool           201-205             Off  None
Et38      tool           tool           201-205             Off  None
Et39      access         errdisabled    All                 Off  None
Et40      tool           tool           All                 On   None switch#
```

20.1.12.25 show platform fm6000 keyframe

The **`show platform fm6000 keyframe`** command displays configured information for the specified keyframes. Keyframes are routable IP packets that the switch inserts into a data stream to provide contextual information that correlate timestamps inserted into data packets with the absolute UTC time and the switch's complete ASIC time counter.

Command Mode

Privileged EXEC

Command Syntax

`show platform fm6000 keyframe` [**KEYFRAME_ID**]

Parameters

KEYFRAME_ID specifies keyframes that the command displays. Options include:

- ***no parameter*** command displays all configured keyframes.
- ***kf_name*** specifies a single keyframe to display information for.

Example

This command displays information concerning the three keyframes that the switch sends.

```
switch# show platform fm6000 keyframe
Keyframe key-2
------------------------
Egress Interface(s): Ethernet17, Ethernet18, Ethernet19, Ethernet20, Ethernet21
Source IP: 10.22.30.144
Destination IP: 10.21.1.14
Destination MAC: 00:09:00:09:00:09
Device ID: 0
Rate: 5 packet(s) per second

Keyframe key-1
------------------------
Egress Interface(s): Ethernet11, Ethernet12, Ethernet13, Ethernet14, Ethernet15
Source IP: 10.22.30.146
Destination IP: 10.21.1.4
Destination MAC: 00:10:4e:21:9f:11
Device ID: 0
Rate: 2 packet(s) per second switch#
```

20.1.12.26 show platform sand mcast capacity

The **show platform sand mcast capacity** command displays the usage details of hardware resources on Sand platform switches.

Command Mode

EXEC

Command Syntax

show platform sand mcast capacity [threshold ***threshold_value***]

Parameters

threshold *threshold_value* displays the list of resources whose usage percentage is greater than or equal to the specified threshold value. Values range from ***0*** to ***100***. The default value is ***100***.

Guidelines

This command is supported on Sand platforms only.

Examples

This command displays the usage details of hardware resources on a Sand platform switch.

```
switch# show platform sand mcast capacity

Multicast Resources
-------------------
'*' - Applies to all Modules
'-' - Not applicable
                          TCAM Resources
--------------------------------------------------------------------------
Resource                            Module        Total     Used     Used%

   v4 MC TCAM       Linecard3-Jericho3/0         4096        2       0.0
```

```
   v4 MC TCAM       Linecard5-Jericho5/0         4096      506      12.4

                          Replication Table Resources
------------------------------------------------------------------------------
Resource                              Module        Total     Used     Used%

Multicast Table Row
                  Linecard3-Jericho3/0.0        262143    10586       4.0
                  Linecard3-Jericho3/1.0        262143    10576       4.0
                  Linecard3-Jericho3/0.1        262143    10586       4.0
                  Linecard3-Jericho3/1.1        262143    10576       4.0
                  Linecard6-Jericho6/2.0        262143    10576       4.0
 switch#
```

20.1.12.27 show tap aggregation groups

The **`show tap aggregation groups`** command displays the TAP and tool port members of the specified TAP aggregation groups.

Command Mode

EXEC

Command Syntax

`show tap aggregation groups` [**INFO_LEVEL**] [**GROUP_NAMES**]

Parameters

- **INFO_LEVEL** port information to display. Options include:
  - ***no parameter*** displays active TAP and tool ports.
  - **detail** displays all configured TAP and tool ports, including inactive ports.
- **GROUP_NAMES** TAP aggregation groups. Options include:
  - ***no parameter*** displays information for all TAP aggregation groups.
  - ***group_list*** displays information for the specified TAP aggregation group list.

    Valid ***group_list*** format is a space-delimited list of one or more TAP aggregation group names.

Example

This command displays the contents of all configured TAP aggregation groups.

```
switch# show tap aggregation groups
Group Name                          Tool Members
------------------------------------------------------
analyze2                            Po101, Po102
analyze3                            Po101, Po103

Group Name                          Tap Members
------------------------------------------------------
analyze2                            Et41, Et42
analyze3                            Et43 switch#
```

20.1.12.28 switchport mpls pop all

The **`switchport mpls pop all`** command configures the MPLS Pop on the switch.

- On DCS-7280R/R2, DCS-7500R/R2, and DCS-7020R MPLS pop is configurable at the tap port using the switchport command.
- On DCS-7280R3, DCS-7500R3, and DCS7800R3 MPLS pop is configurable at the tool port using the switchport command.

- On DCS-7280R3, DCS-7500R3, and DCS7800R3 the hardware forwarding profile must also be set before MPLS pop can be used.

The **no switchport mpls pop all** command removes the switchport configuration by modifying the corresponding statements in ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

switchport [tool | tap] **mpls pop all**

no switchport [tool | tap] **mpls pop all**

Parameters

- **tool** Tool port configuration.
- **tap** Tap port configuration.

Example

This command configures the MPLS Pop at the Tap port on the switch.

```
switch(config-if-Et1)# [no] switchport tap mpls pop all
```

20.1.12.29 switchport tap allowed vlan

The **switchport tap allowed vlan** command creates or modifies the list of VLANs for which the configuration mode interface, in TAP mode, handles tagged traffic. By default, interfaces handle tagged traffic for all VLANs. Command settings persist in ***running-config*** without taking effect when the switch is not in TAP aggregation mode or the interface is not in TAP aggregation mode.

The **no switchport tap allowed vlan** and **default switchport tap allowed vlan** commands restore the TAP mode default allowed VLAN setting of **all** by removing the corresponding **switchport tap allowed vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport tap allowed vlan **EDIT_ACTION**

Parameters

EDIT_ACTION modifications to the VLAN list. Options include:

- ***v_range*** creates VLAN list from range of VLANs specified by ***v_range***.
- **add** ***v_range*** adds specified VLANs to current list.
- **all** VLAN list contains all VLANs.
- **except** ***v_range*** VLAN list contains all VLANs except those specified by ***v_range***.
- **none** VLAN list is empty (no VLANs).
- **remove** ***v_range*** removes VLANs specified by ***v_range*** from current list.

Example

These commands create the TAP mode allowed VLAN list of ***26-30*** for ***interface ethernet 20***.

```
switch(config)# interface ethernet 20
switch(config-if-Et20)# switchport tap allowed vlan 26-30
eswitch(config-if-Et20)# show active
```

```
interface Ethernet20
   switchport mode tap
   switchport tap allowed vlan 26-30
switch(config-if-Et20)#
```

20.1.12.30 switchport tool encapsulation

The **switchport tool encapsulation** command is configured under Ethernet Interface configuration mode. When enabled on a tool port, only packets egressing this tool port will have their BR-E/VN tags stripped on DCS-7280R3, DCS-7500R3and DCS7800R3.

The **no switchport tool encapsulation** or **default switchport tool encapsulation** commands disables the 802.1br-E/VN Tag Stripping from the **running-config**.

Command Mode

Interface-Ethernet Configuration Mode

Command Syntax

switchport tool encapsulation {dot1br | vn-tag} **strip**

no switchport tool encapsulation dot1br | vn-tag} **strip**

defualt switchport tool encapsulation dot1br | vn-tag} **strip**

Parameters

- **dot1br** strips the 802.1br-E headers.
- **vn-tag** strips the VN Tag headers.

Examples

- To activate BR E-Tag stripping on a tool port, add the following configuration:

```
switch(config-if-Et1)# [no] switchport tool encapsulation dot1br strip
```

- To activate VN-TAG stripping on a tool port, add the following configuration:

```
switch(config-if-Et1)# [no] switchport tool encapsulation vn-tag strip
```

20.1.12.31 switchport tap default group

The **switchport tap default group** command assigns the configuration-mode interface to the specified tool group as a TAP port member. TAP aggregation groups associate a set of TAP ports with a set of tool ports. Both TAP ports and tool ports may belong to multiple TAP aggregation groups.

The **no switchport tap default group** and **default switchport tap default group** commands remove the configuration-mode interface from the TAP aggregation group to which it is assigned by deleting the corresponding **switchport tap default group** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-port Channel Configuration

Command Syntax

switchport tap default group ***group_name***

no switchport tap default group

default switchport tap default group

Parameters

group_name tool group name.

Restriction

This command is available on FM6000 platform switches only.

Example

These commands assign ***port channel 101*** to TAPs aggregation group ***tag-1***.

```
switch(config)# interface port-channel 101
switch(config-if-Po101)# switchport tap default group tag-1
switch(config-if-Po101)# show interfaces port-channel 101 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Po101     access         notconnect     1        1    0          tag-1
switch(config)#
```

20.1.12.32 switchport tap identity

The **`switchport tap identity`** command associates a VLAN number to the configuration mode TAP interface. Tool ports that are configured to encapsulate packets with an dot1q-style tag enter the number specified by this command as the s-VLAN (tier 1) for packets received from this TAPs port. The default identity value is 1.

The **`no switchport tap identity`** and **`default switchport tap identity`** commands restore ***vlan 1*** as the configuration-mode ports's identity VLAN by removing the corresponding **`switchport tap identity`** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

`switchport tap identity` ***port_id***

`no switchport tap identity`

`default switchport tap identity`

Parameter

port_id port's identity VLAN. Values range from ***1*** to ***4094***. Default is ***1***.

Related Commands

`switchport tool identity` configures a tool port to encapsulate packets received from TAP ports.

Restriction

This command is available only on FM6000 platform switches.

Example

These commands ***171*** as the identity value for ***interface ethernet 17***.

```
switch(config)# interface ethernet 17
switch(config-if-Et17)# switchport tap identity 171
switch(config-if-Et17)# show active
interface Ethernet17
   switchport tap identity 171
switch(config-if-Et17)# show interfaces ethernet 17 tap
```

```
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan             Group
-------------------------------------------------------------------------
Et17      access         connected      1        171   0
switch(config-if-Et17)#
```

20.1.12.33 switchport tap native vlan

The **switchport tap native vlan** command specifies the TAP-mode native VLAN for the configuration-mode interface. Interfaces in TAP mode associate untagged frames with the native VLAN. The default native VLAN for all interfaces is ***vlan 1***. Command settings persist in ***running-config*** without taking effect when the switch is not in TAP aggregation mode or the interface is not in TAP mode.

The **no switchport tap native vlan** and **default switchport tap native vlan** commands restore ***vlan 1*** as the TAP-mode native VLAN to the configuration-mode interface by removing the corresponding **switchport tap native vlan** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport tap native vlan ***v_num***

no switchport tap native vlan

default switchport tap native vlan

Parameter

v_num TAP-mode native VLAN ID. Values range from 1 to 4094. Default is 1.

Restriction

This command is available only on FM6000 platform switches.

Example

These commands assign ***vlan 25*** as the TAP-mode native VLAN for ***interface ethernet 7***.

```
switch(config)# interface ethernet 7
switch(config-if-Et7)# switchport tap native vlan 25
switch(config-if-Et7)# show interface ethernet 7 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan             Group
-------------------------------------------------------------------------
Et7       tool           connected      25       1     0          ---
switch(config-if-Et7)#
```

20.1.12.34 switchport tap truncation

The **switchport tap truncation** command configures the configuration-mode interface, as a TAP port, to truncate inbound packets to the specified packet size. This command is in effect when the port is in TAP mode and the switch is in TAP aggregation mode. Command settings persist in ***running-config*** without taking effect when the switch is not in TAP aggregation mode or the interface is not in TAP mode. By default, TAP ports do not truncate inbound packets.

The **no switchport tap truncation** and **default switchport tap truncation** commands restore the default behavior of not truncating packets received by the configuration-mode interface by removing the corresponding **switchport tap truncation** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport tap truncation ***packet_size***

no switchport tap truncation

default switchport tap truncation

Parameter

packet_size size of truncated packets (bytes). Values range from ***100*** to ***9236***. Default value of ***0*** corresponds to not truncating packets.

Restriction

This command is available only on FM6000 platform switches.

Examples

- These commands configure ***interface ethernet 38*** to truncate packets to ***150*** bytes.

```
switch(config)# interface ethernet 38
switch(config-if-Et38)# switchport tap truncation 150
switch(config-if-Et38)# show interface ethernet 38 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et38      access         notconnect     1        1     150        ---
switch(config-if-Et38)#
```

- These commands configure ***interface ethernet 38*** to send complete packets to tool ports in its TAP aggregation group.

```
switch(config-if-Et38)# no switchport tap truncation
switch(config-if-Et38)# show interface ethernet 38 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et38      access         notconnect     1        1     0          ---
switch(config-if-Et38)#
```

20.1.12.35 switchport tool allowed vlan

The **switchport tool allowed vlan** command creates or modifies the list of VLANs for which the configuration-mode interface, in tool mode, handles tagged traffic. By default, interfaces handle tagged traffic for all VLANs. Command settings persist in ***running-config*** without taking effect when the switch is not in TAP aggregation mode or the interface is not in TAP aggregation mode.

The **no switchport tool allowed vlan** and **default switchport tool allowed vlan** commands restore the tool mode default allowed VLAN setting of **all** by removing the corresponding **switchport tool allowed vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport tool allowed vlan **EDIT_ACTION**

Parameters

EDIT_ACTION modifications to the VLAN list. Options include:

- ***v_range*** creates VLAN list from ***v_range***.
- **add *v_range*** adds specified VLANs to current list.
- **all** VLAN list contains all VLANs.
- **except *v_range*** VLAN list contains all VLANs except those specified.
- **none** VLAN list is empty (no VLANs).
- **remove *v_range*** removes specified VLANs from current list.

  Valid ***v_range*** formats include number, range, or comma-delimited list of numbers and ranges.

Example

These commands create the tool mode allowed VLAN list of ***16-20*** for ***interface ethernet 38***.

```
switch(config)# interface ethernet 38
switch(config-if-Et38)# switchport tool allowed vlan 16-20
switch(config-if-Et38)# show interfaces ethernet 38 tool
Port      Configured     Status         Allowed             Id   Timestamp
          Mode                          Vlans               Tag  Mode
-------------------------------------------------------------------------
Et38      access         notconnect     16-20               Off  None
switch(config-if-Et38)#
```

20.1.12.36 switchport tool group

The **`switchport tool group`** command modifies the configuration-mode interface's tool port membership in the specified TAP aggregation groups. Tool ports may belong to multiple TAP aggregation groups. Command options for configuring a port's TAP aggregation group membership include:

- specifying the groups to which the port belongs (supersedes the port's previous group memberships).
- adding to the list of groups to which the port belongs.
- deleting from the list of groups to which the port belongs.

TAP aggregation groups associate a set of TAP ports with a set of tool ports. A TAP port can belong to a maximum of one default TAP aggregation group.

The **`no switchport tool default group`** and **`default switchport tool default group`** commands remove the configuration-mode interface from all TAP aggregation groups to which it is assigned as a tool port by modifying the corresponding statements in ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

`switchport tool group` **EDIT_ACTION**

Parameters

EDIT_ACTION specifies changes to the list of groups to which the port belongs.

- **add *group_list*** specifies additional groups to which the port belongs.
- **remove *group_list*** removes interface as a tool port member from specified groups.
- **set *group_list*** specifies groups to which interface belongs as a tool port.

  Valid ***group_list*** format is a space-delimited list of one or more TAP aggregation group names.

Restriction

This command is available only on FM6000 platform switches.

Examples

- These commands associate ***interface ethernet 40*** with three TAP aggregation groups.

```
switch(config)# interface ethernet 40
switch(config-if-Et40)# switchport tool group set tag-1 tag-2 tag-3
switch(config-if-Et40)# show active
interface Ethernet40
   switchport tool group set tag-3 tag-2 tag-1
switch(config-if-Et40)#
```

- These commands add ***tag-7*** to the tap aggregation groups to which ***interface ethernet 40*** belongs.

```
switch(config-if-Et40)# switchport tool group add tag-7
switch(config-if-Et40)# show active
interface Ethernet40
   switchport tool group set tag-3 tag-7 tag-2 tag-1
switch(config-if-Et40)#
```

- These commands specify ***tag-9*** as the only group to which ***interface ethernet 40***belongs.

```
switch(config-if-Et40)# switchport tool group set tag-9
switch(config-if-Et40)# show active
interface Ethernet40
   switchport tool group set tag-9
switch(config-if-Et40)#
```

20.1.12.37 switchport tool identity

The **`switchport tool identity`** command configures the configuration-mode interface to add a tier-1 VLAN tag (dot1q) to packets it receives from TAP ports. The VLAN number on the dot1q tag is specified by the **`switchport tap identity`** command configured for the TAP port that supplies the packets. By default, tool ports do not encapsulate packets with the tier-1 VLAN tag.

The **`no switchport tool identity`** and **`default switchport tool identity`** commands restore the default VLAN handling method for the configuration-mode interface by removing the corresponding **`switchport tool identity`** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

`switchport tool identity dot1q`

`no switchport tool identity dot1q`

`default switchport tool identity dot1q`

Restriction

This command is available only on FM6000 platform switches.

Example

These commands configure ***interface ethernet 40*** to include a dot1q tag on egress packets.

```
switch(config)# interface ethernet 40
switch(config-if-Et40)# switchport tool identity dot1q
switch(config-if-Et40)# show active
```

```
interface Ethernet40
   switchport mode tool
   switchport tool identity dot1q
   switchport tool group set tag-9
switch(config-if-Et40)#
```

20.1.12.38 switchport tool truncation

The **switchport tool truncation** command configures the configuration-mode interface, as a tool port, to truncate outbound packets to ***160*** bytes. This command is in effect when the port is in tool mode and the switch is in TAP aggregation mode. Command settings persist in ***running-config*** without taking effect when the switch is not in TAP aggregation mode or the interface is not in tool mode. By default, tool ports do not truncate outbound packets.

The **no switchport tool truncation** and **default switchport tool truncation** commands restore the default behavior (not truncating packets that exit the configuration mode interface) by removing the corresponding **switchport tool truncation** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Interface-Port Channel Configuration

Command Syntax

switchport tool truncation ***packet_size***

no switchport tool truncation

default switchport tool truncation

Parameters

packet_size size of truncated packets in bytes. The only permitted value is ***160***.

Examples

- These commands configure ***interface ethernet 38***, as a tool port, to truncate packets on egress to ***160*** bytes.

```
switch(config)# interface ethernet 38
switch(config-if-Et38)# switchport mode tool
switch(config-if-Et38)# switchport tool truncation 160
switch(config-if-Et38)#
```

- These commands configure ***interface ethernet 38*** to send complete packets.

```
switch(config)# interface ethernet 38
switch(config-if-Et38)# no switchport tool truncation
switch(config-if-Et38)#
```

20.1.12.39 tap aggregation

The **tap aggregation** command places the switch in TAP-aggregation configuration mode. The switch's TAP aggregation mode is enabled or disabled by the **mode** command in TAP-aggregation configuration mode.

When TAP aggregation mode is enabled, normal switching and routing operations are disabled. A port's switchport status depends on the switch's TAP aggregation mode and the port's switchport mode:

- **TAP aggregation mode enabled**: TAP and tool ports are enabled. Switching ports are errdisabled.
- **TAP aggregation mode disabled**: TAP and tool ports are errdisabled. Switching ports are enabled.

The **no tap aggregation** and **default tap aggregation** commands disable tap aggregation mode on the switch by removing all TAP-aggregation configuration mode commands from ***running-config***.

TAP-aggregation configuration mode is not a group-change mode; ***running-config*** is changed immediately upon entering commands. Exiting TAP-aggregation configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

tap aggregation

no tap aggregation

default tap aggregation

Commands Available in TAP-aggregation Configuration Mode

mode (tap-agg configuration mode)

Related Commands

switchport mode

Examples

- These commands place the switch in TAP-aggregation configuration mode and enable TAP aggregation.

```
switch(config)# tap aggregation
switch(config-tap-agg)# mode exclusive
switch(config-tap-agg)# show active
tap aggregation
 mode exclusive
switch(config-tap-agg)#
```

- This command disables TAP aggregation and removes all TAP-aggregation configuration mode commands from ***running-config***.

```
switch(config)# no tap aggregation
switch(config)#
```

20.2 Latency Analyzer (LANZ)

Arista Networks' Latency Analyzer (LANZ) is a family of EOS features that provide enhanced visibility into network dynamics, particularly in areas related to the delay packets experience through the network. The LANZ feature is available on the FM6000, Arad, Trident II, Trident 3, Jericho, Tomahawk and XP80 switch platforms.

This section describes the purpose, behavior, and configuration of LANZ features. Topics covered by this section include:

- Introduction to LANZ
- LANZ Overview
- Configuring LANZ
- LANZ Commands

20.2.1 Introduction to LANZ

LANZ tracks interface congestion and queuing latency with real-time reporting. With LANZ application layer event export, external applications can predict impending congestion and latency. This enables the application layer to make traffic routing decisions with visibility into the network layer.

With LANZ, network operations teams and administrators have near real-time visibility into the network, enabling early detection of microbursts. LANZ continually monitors congestion, allowing for rapid detection of congestion and sending of application layer messages.

20.2.2 LANZ Overview

LANZ monitors output queue lengths to provide congestion information for individual interfaces. This allows for more detailed analysis of congestion events, and allows identification of potential latency problems before they arise. On some platforms, LANZ also monitors global buffer usage.

Output queues for each port are monitored, and information about queue congestion events can be accessed in the form of system log messages, reports, or streaming.

20.2.2.1 LANZ Monitoring Mechanism

LANZ provides congestion data by continuously monitoring each port's output queue lengths. When the length of an output queue exceeds the upper threshold for that port, LANZ generates an over-threshold event.

20.2.2.1.1 Notifying and Polling Modes

Notifying Mode

In Notifying Mode (available on all platforms), LANZ monitors congestions on all queues and generates Start, Update, and End events every five seconds or at user-configured intervals.

- **Start** event is generated when any queue on an interface exceeds the upper threshold.
- **Update** events are generated periodically while the congested queue remains above the lower threshold. The interval at which Update events are generated is configured using the **queue-monitor length update-interval** command.
- **End** event is generated when the congested queue drops below the lower threshold.

Polling Mode

Polling Mode is available only on Arad and Jericho switches. On these switches, LANZ can be configured to use Notifying Mode, but it operates in Polling Mode by default. In Polling Mode, LANZ polls the most congested queue in each ASIC and continues to report an over-threshold state every ***800*** microseconds until all queue lengths for the port pass below the lower threshold.

20.2.2.2 LANZ Logging

Over-threshold events generated by LANZ can be logged as system log messages. Log messages are generated for events on all ports, at a maximum rate of one message per second per interface. The interval between messages can be configured globally.

Log messages indicate the time of the event, the interface affected, the threshold set for that interface, and the actual number of entries in the port's queue.

20.2.2.3 LANZ Reporting

Detailed LANZ data can be viewed through the CLI or exported as a CSV-formatted report.

A circular FIFO event buffer is dynamically shared by all interfaces. When an interface begins generating LANZ over-threshold events it can fill all available buffer space. However, each interface is guaranteed sufficient resources for a minimum of ***500*** entries.

20.2.2.4 LANZ Streaming

On some platforms, external client applications can also receive congestion event information as a data stream. The switch can stream LANZ data to up to ***100*** clients via TCP through port ***50001***.

Streamed data is in Google protocol buffer format, and includes both over-threshold events and LANZ configuration information.

20.2.3 Configuring LANZ

LANZ is disabled by default and must be enabled to function. Upper and lower queue-length thresholds can be defined for individual interfaces.

These sections describe the basic LANZ configuration steps:

- Enabling and Disabling LANZ
- Specifying the LANZ Mode on Jericho and Arad Switches
- Configuring LANZ Congestion Thresholds
- Setting LANZ Traffic Sampling
- Logging LANZ Congestion Events
- Viewing LANZ Data
- Streaming LANZ Data

20.2.3.1 Enabling and Disabling LANZ

LANZ is enabled by default on Sand platform switches (those with Arad, Jericho, or Qumran chipsets), allowing the switch to collect and display latency information. LANZ is disabled by default on all other switches. If LANZ is disabled, the **queue-monitor length (global configuration mode)** command enables LANZ with the current settings, or with the default settings if none have been configured.

When LANZ is enabled, the switch monitors queue lengths on all front-panel ports, and on CPU and fabric ports on selected platforms. Queue length data is available in the following forms:

- syslog data (see **queue-monitor length log**)
- CLI display or CSV-format output (see **show queue-monitor length**)
- data stream (see **queue-monitor streaming**)

To disable LANZ globally, enter the **`no queue-monitor length`** command in global configuration mode. Disabling LANZ globally also discards LANZ log data, but retains settings. To disable LANZ on an individual interface, enter the **`no queue-monitor length`** command in interface Ethernet configuration mode.

Examples

- This command enables LANZ on the switch.

```
switch(config)# queue-monitor length
switch(config)#
```

- This command disables LANZ on the switch.

```
switch(config)# no queue-monitor length
switch(config)#
```

- These commands disable LANZ on Ethernet interface 7.

```
switch(config)# interface ethernet 7
```

```
switch(config-if-Et7)# no queue-monitor length
switch(config-if-Et7)#
```

20.2.3.2 Specifying the LANZ Mode on Jericho and Arad Switches

On Jericho and Arad switches, LANZ operates by default in Polling Mode, which provides congestion data for the most congested queue per ASIC. They also support Notifying Mode (the LANZ mode used on all other switches), which generates Start, Update, and End events for all queues.

Notifying Mode is enabled using the **queue-monitor length notifying** command.

Examples

- This command enables Notifying Mode on a Jericho or Arad switch.

```
switch(config)# queue-monitor length notifying
```

- This command disables Notifying Mode on a Jericho or Arad switch, returning LANZ to the default Polling Mode.

```
switch(config)# no queue-monitor length notifying
```

20.2.3.3 Configuring LANZ Congestion Thresholds

When LANZ is enabled on the switch, it generates over-threshold events when queue lengths on any monitored interface exceed the upper threshold value and continues generating them until all the queue lengths on that interface drop back below the lower threshold.

20.2.3.3.1 Setting the Congestion Update Interval

The **queue-monitor length update-interval** command specifies the frequency with which congestion information is updated in microseconds.

- This command sets the time between congestion updates to ***10*** seconds.

```
switch(config)# queue-monitor length update-interval 10000000
switch(config)#
```

- This command resets the time between congestion updates to its default value of ***5*** seconds.

```
switch(config)# default queue-monitor length update-interval
switch(config)#
```

20.2.3.3.2 Congestion Thresholds on FM6000, Trident II, Trident 3, and Tomahawk Switches

Queue lengths are measured in 480-byte segments on FM6000 switches, in 208-byte segments on Trident II and Tomahawk switches, and in 256-byte segments on Trident 3 switches. The default threshold values are ***512*** segments and ***256*** segments. To change the threshold values for a specific interface, use the **queue-monitor length thresholds** command.

FM6000 switches can also monitor global buffer usage. Global buffers are measured in 160-byte segments; the default threshold values are ***10940*** segments and ***4376*** segments. To enable global buffer monitoring, use the **queue-monitor length global-buffer** command. To change the threshold values for global buffer usage monitoring on the switch, use the **queue-monitor length global-buffer thresholds** command.

Examples

- These commands set the upper and lower queue-length thresholds on ***interface ethernet 5*** to ***300*** segments and ***200*** segments.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# queue-monitor length thresholds 300 200
switch(config-if-Et5)#
```

- These commands enable global buffer monitoring on the switch and set the upper and lower thresholds to ***9000*** segments and ***4000*** segments.

```
switch(config)# queue-monitor length global-buffer
switch(config)# queue-monitor length global-buffer thresholds 9000 4000
switch(config)#
```

20.2.3.3.3 Congestion Thresholds on Arad Switches

Queue lengths are measured in bytes. The top threshold value can be between ***2*** and ***52428800*** bytes (the default value is ***52428800*** bytes). To change the upper threshold value for a specific interface, use the **queue-monitor length threshold (Arad/Jericho/Qumran)** command.

Example

These commands set the upper queue-length threshold on ***interface ethernet 5*** to ***2614400*** bytes.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# queue-monitor length thresholds 2614400
switch(config-if-Et5)#
```

20.2.3.3.4 CPU Port Congestion Thresholds

CPU port queue lengths are also monitored on selected platforms. All CPU ports share common threshold values, which are configured using the **queue-monitor length cpu thresholds** command. Individual CPU port congestion thresholds cannot be separately configured.

Examples

- This command sets the upper queue-length threshold for congestion monitoring on all CPU ports to ***1000*** segments and the lower limit to ***300*** segments.

```
switch(config)# queue-monitor length cpu thresholds 1000 300
switch(config)#
```

- This command resets the queue-length thresholds for CPU port congestion to the default values of ***512*** and ***256***.

```
switch(config)# default queue-monitor length cpu thresholds
switch(config)#
```

20.2.3.3.5 Fabric Port Congestion Thresholds

Fabric port queue lengths are also monitored on selected platforms. All fabric ports share common threshold values, which are configured using the **queue-monitor length fabric thresholds** command. Individual fabric port congestion thresholds cannot be separately configured.

Examples

- This command sets the upper queue-length threshold for congestion monitoring on all fabric ports to ***1000*** segments and the lower limit to ***300*** segments.

```
switch(config)# queue-monitor length fabric thresholds 1000 300
switch(config)#
```

- This command resets the queue-length thresholds for fabric port congestion to the default values of ***512*** and ***256***.

```
switch(config)# default queue-monitor length fabric thresholds
switch(config)#
```

20.2.3.4 Setting LANZ Traffic Sampling

The switch can be configured to automatically send congested traffic to either the CPU or an Ethernet egress interface destination when a queue threshold is crossed by enabling LANZ mirroring through the command **queue-monitor length mirror**. The CPU or an egress interface mirror destination is then configured through the command **queue-monitor length mirror destination**. LANZ traffic sampling includes exporting congested traffic to a packet capture device or another tool for analysis, or directly to the switch CPU for inspection through the command **tcpdump queue-monitor**.

Examples

- This command enables LANZ traffic sampling.

```
switch(config)# queue-monitor length mirror
switch(config)#
```

- This command disables LANZ traffic sampling.

```
switch(config)# no queue-monitor length mirror
switch(config)#
```

- This command configures LANZ traffic sampling for a CPU interface mirror destination.

```
switch(config)# queue-monitor length mirror destination cpu
switch(config)#
```

- This command configures LANZ traffic sampling for an Ethernet interface mirror destination for ports ***1*** through ***5***.

```
switch(config)# queue-monitor length mirror destination Ethernet 1-5
switch(config)#
```

- This command configures LANZ traffic sampling for an Ethernet interface mirror destination for ports ***6***, ***10***, and ***12*** through ***14***.

```
switch(config)# queue-monitor length mirror destination Ethernet 6,10,12-14
switch(config)#
```

- This command inspects traffic on the switch.

```
switch(config)# tcpdump queue-monitor
tcpdump: WARNING: lanz: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lanz, link-type EN10MB (Ethernet), capture size 65535 bytes
...
0 packets captured
0 packets received by filter
```

```
0 packets dropped by kernel
switch(config)#
```

20.2.3.5 Logging LANZ Congestion Events

To generate syslog messages when queue lengths on an interface exceed its upper threshold, enable logging with the **queue-monitor length log** command. When logging is enabled, a log message is generated each time one or more queues on an interface exceed the upper threshold value for that interface (see **queue-monitor length threshold (Arad/Jericho/Qumran)** or **queue-monitor length thresholds**). Once an interface is over threshold, additional messages are generated at a maximum rate of one per interval as long as the queue length remains above the lower threshold for that interface. No syslog message is generated when queue length drops back under threshold.

Queue length information is not included in log messages, but can be accessed by displaying LANZ data or exporting reports.

On FM6000 platforms, log messages can also be created whenever global buffer usage exceeds its upper threshold value (see **queue-monitor length global-buffer thresholds**). To enable global buffer monitoring, use the **queue-monitor length global-buffer** command. To log over-threshold events for the global buffer, use the **queue-monitor length global-buffer log** command.

Examples

- This command enables queue-length over-threshold logging with a minimum interval of ***10*** seconds between messages for a given interface.

```
switch(config)# queue-monitor length log 10
```

- This command disables queue-length over-threshold logging on the switch.

```
switch(config)# queue-monitor length log 0
```

- This is an example of a queue-length log message.

```
Oct 27 12:48:22 switch QUEUE_MONITOR-6-LENGTH_OVER_THRESHOLD:
 Interface
Ethernet6 queue length is over threshold of 512, current length is
 1024.
```

- This command enables global buffer over-threshold logging on the switch with a minimum interval of ***60*** seconds between messages.

```
witch(config)# queue-monitor length global-buffer log 60
```

20.2.3.6 Viewing LANZ Data

LANZ status, and the data stored in the LANZ data buffer, can be viewed using the CLI. Output varies by switch platform, and can be limited to a specified number of records.

20.2.3.6.1 Viewing LANZ Data on Arad Platform Switches

When LANZ is enabled on an Arad platform switch, the **show queue-monitor length** command displays a report of recent over-threshold events for a range of interfaces or for all interfaces. By default, the command displays data for all interfaces, limited to the last 1000 records, with the most recent events listed first. To view a subset of the LANZ data, limited to a specified number of records, use the **show queue-monitor length limit** command.

Example

This command displays the last ***100*** records for Ethernet interfaces ***6*** through ***8***.

```
switch# show queue-monitor length ethernet 6-8 limit 100
Report generated at 2010-01-01 12:56:13

Time                          Interface   Queue length (segments, 1 to 512 bytes)
-------------------------------------------------------------------------------
0:00:07.43393 ago             Et6         1049
0:00:39.22856 ago             Et7         2039
1 day, 4:33:23.12345 ago      Et6         1077
```

To view the current LANZ configuration for the switch and for each interface, use the **show queue-monitor length status** command.

Example

This command displays LANZ configuration and status information.

```
switch(config)# show queue-monitor length status
Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Maximum queue length in bytes : 52428800
Port threshold in bytes:
Port     High threshold
Et3/1           5242880
Et3/2           5242880
Et3/3           5242880
Et3/4           5242880
Et3/5           5242880
```

20.2.3.6.2 Viewing LANZ Data on FM6000, Trident II, Trident 3, and Tomahawk Platform Switches

When LANZ is enabled on an FM6000, Trident II, Trident 3, or Tomahawk platform switch, the **show queue-monitor length** command displays a report of recent over-threshold events for a range of interfaces or for all interfaces. By default, the command displays data for all interfaces, limited to the last ***1000*** records, with the most recent events listed first. To view a subset of the LANZ data, limited to a specified number of records, use the **show queue-monitor length limit** command.

Example

This command displays the last ***100*** records for Ethernet interfaces ***6*** through ***8***.

```
switch# show queue-monitor length ethernet 6-8 limit 100
Report generated at 2010-01-01 12:56:13

Time                          Interface   Queue length (segments, 1 to 512 bytes)
-------------------------------------------------------------------------------
0:00:07.43393 ago             Et6         1049
0:00:39.22856 ago             Et7         2039
1 day, 4:33:23.12345 ago      Et6         1077
```

To view the current LANZ configuration for the switch and for each interface, use the **show queue-monitor length status** command.

Example

This command displays LANZ configuration and status information.

```
switch(config)# show queue-monitor length status
queue-monitor length enabled
Global Buffer Monitoring
------------------------
Global buffer monitoring is enabled
Segment size in bytes :   160
```

```
Total buffers in segments : 36864
High threshold : 10940
Low threshold :  4376


Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Segment size in bytes :   480
Maximum queue length in segments :  3647
Port thresholds in segments:
Port     High threshold  Low threshold
Et1                 512            256
Et2                 512            256
Et3                 512            256
Et4                 512            256
Et5                 512            256
```

To view all available LANZ records, use the **show queue-monitor length all** command.

Example

This command displays all available LANZ records.

```
switch> show queue-monitor length all

Report generated at 2013-04-01 13:23:13
E-End, U-Update, S-Start, TC-Traffic Class
GH-High, GU-Update, GL-Low
Segment size for E, U and S congestion records is 480 bytes
Segment size for GL, GU and GH congestion records is 160 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-------------------------------------------------------------------------------
-
Type    Time                    Intf    Congestion     Queue       Time of Max
                                (TC)    duration       length      Queue length
                                        (usecs)        (segments)  relative to
                                                                   congestion
                                                                   start
                                                                   (usecs)
-------------------------------------------------------------------------------
-
E  0:00:00.07567 ago         Et22(7)   >=71 mins      20*         30us
GU 0:00:00.15325 ago         N/A       N/A            5695        N/A
U  0:00:00.19859 ago         Et4(1)    N/A            5693        N/A
GU 0:00:00.95330 ago         N/A       N/A            5696        N/A
U  0:00:00.99859 ago         Et4(1)    N/A            5695        N/A
E  0:00:01.28821 ago         Et44(1)   9672us         2502*       7294us
S  0:00:01.17591 ago         Et22(7)   N/A            26          N/A
U  0:00:03.08248 ago         Et44(1)   N/A            50          N/A
S  12days,8:56:44.07567 ago  Et44(1)   N/A            20          N/A switch>
```

On the FM6000, Trident II, Trident 3, and Tomahawk platforms, information is also available for the number of dropped packets (see **show queue-monitor length drops**) and transmission latency (see **show queue-monitor length tx-latency**); global buffer usage can also be viewed on FM6000 platforms (see **show queue-monitor length global-buffer**).

20.2.3.7 Streaming LANZ Data

To support analysis of latency conditions, the switch can be configured to stream LANZ congestion and configuration data. The switch streams LANZ data via TCP in Google protocol buffer format through port ***50001*** and through the management interface.

You must create a client application to receive the streaming data. By default, the switch will accept up to ***10*** client connections for streaming LANZ data. This limit can be configured up to a maximum of ***100***. Maximum connections can be configured when LANZ is disabled.

20.2.3.7.1 Enabling and Disabling LANZ Data Streaming

LANZ data streaming is disabled by default. To enable streaming, issue the **`no show queue-monitor streaming clients`** command in queue-monitor streaming configuration mode. To disable streaming, use the **show queue-monitor streaming clients** command.

When streaming is disabled, a message is sent to any connected clients and the connections are closed.

To ensure client access to LANZ data, add a rule to any relevant ACL permitting traffic destined for the LANZ port (***50001***) before initiating a client connection for streaming from a remote host. A static rule (sequence number ***130***) in the default control plane ACL permits LANZ traffic, but a similar rule must be added to any user-created ACL.

Examples

- These commands enable the streaming of LANZ data from the switch.

```
switch(config)# queue-monitor streaming
switch(config-qm-streaming)# no shutdown
switch(config-qm-streaming)#
```

- These commands disable LANZ data streaming.

```
switch(config)# queue-monitor streaming
switch(config-qm-streaming)# shutdown
switch(config-qm-streaming)#
```

20.2.3.7.2 Configuring Maximum Connections

By default, the switch will accept a maximum of ***10*** client connections for LANZ data streaming. This maximum can be configured using the **max-connections** command. If a client connects to the switch after the limit has been reached, an error message is sent and the connection is closed.

Example

This command sets the maximum number of client connections for LANZ data streaming to ***50***.

```
switch(config-qm-streaming)# max-connections 50
```

20.2.3.7.3 LANZ Streaming Messages

When streaming is enabled, LANZ sends a message whenever a congestion event or a configuration event occurs. The messages are streamed in Google protocol buffer format.

Configuration Messages

A configuration message is sent whenever a change is made to the LANZ configuration settings on the switch. The switch also sends a configuration message when a new client connection is established.

The configuration message includes the following information:

- **`timestamp`** time of change in configuration in tens of microseconds (UTC).
- **`lanzVersion`** LANZ feature version.
- **`numOfPorts`** number of ports in the switch.
- **`segmentSize`** segment size.
- **`maxQueueSize`** maximum queue size in segments.
- **`qLenInterval`** frequency of updates.
- **`intfName`** name of the port.
- **`switchId`** ID of the chip on a multi-chip system.

- **portId** ID of the port.
- **internalPort** “true” if it is an internal port.
- **highThreshold** higher threshold value.
- **lowThreshold** lower threshold value.

Congestion Messages

A congestion message is sent whenever LANZ generates an over-threshold event.

The congestion message includes the following information:

- **timestamp** time of congestion in micro-seconds (UTC).
- **intfName** name of the port.
- **switchId** ID of the chip on a multi-chip system.
- **portId** ID of the port.
- **queueSize** queue size in segments at time of congestion.

20.2.3.7.4 Creating the LANZ Client

For a client device to receive streaming data from the LANZ server, it must be running a client application designed to receive LANZ data. Client programs must be based on the Google protocol buffer schema file describing the structure of the congestion and configuration messages which LANZ streams.

Google Protocol Buffers

Google protocol buffers provide an efficient mechanism for serializing LANZ data for streaming. A protocol buffer package is needed in order to run a LANZ client.

The latest version of the Google protocol buffer source code is available at this address: **http://code.google.com/p/protobuf/downloads/list**

LANZ Message Schema

LANZ client applications must be designed based on the LANZ protocol buffer schema, which defines the format and contents of the streamed messages. The schema file is shown below, and is also available in the Extensions/LANZ directory on this page: **https://www.arista.com/en/support/software-download**

```
package LanzProtobuf;

message ConfigRecord {
required uint64 timestamp = 1; // Time of change in configuration in
micro-seconds (UTC)
required uint32 lanzVersion = 2; // LANZ feature version
required uint32 numOfPorts = 3; // Num of ports in the switch
required uint32 segmentSize = 4; // Segement size
required uint32 maxQueueSize = 5; // Maximum queue size in segments
optional uint32 qLenInterval = 10; // Frequency of update
message PortConfigRecord {
required string intfName = 1; // Name of the port
required uint32 switchId = 2; // Id of the chip on a multi-chip system
required uint32 portId = 3; // Id of the port
required bool internalPort = 4; // 'True' if it's an internal port
required uint32 highThreshold = 5; // Higher threshold
required uint32 lowThreshold = 6; // Lower threshold
}

repeated PortConfigRecord portConfigRecord = 6; // Lanz config details of each
port
}

message CongestionRecord {
required uint64 timestamp = 1; // Time of congestion in micro-seconds
```

```
(UTC)
required string intfName = 2; // Name of the port
required uint32 switchId = 3; // Id of the chip on a multi-chip system
required uint32 portId = 4; // Id of the port
required uint32 queueSize = 5; // Queue size in segments at time of
congestion
}

message ErrorRecord {
required uint64 timestamp = 1; // Time of event in micro-seconds (UTC)
required string errorMessage = 2; // Text message
}

message LanzRecord {
optional ConfigRecord configRecord = 1;
optional CongestionRecord congestionRecord = 2;
optional ErrorRecord errorRecord = 3;
}
```

Implementation Procedure

The following steps create and install a functional client to receive streamed LANZ data. This procedure assumes a functional Python programming environment.

1. Download the example client (lanz_client.py) from the Arista website. It is available in the Extensions/ LANZ directory on this page: https://www.arista.com/en/support/software-download
2. Decompress the GPB archive to a directory.
3. Run the GPB C++ compilation and install. With default flags using GCC on *nix platforms, this will produce a binary called “protoc” in your /usr/local/bin directory.
4. From the archive root, **cd** to python, and run the following commands:
   - **python setup.py build**
   - **python setup.py test**
5. Next, use the protoc compiler to convert the Lanz.proto file into a Python program called Lanz_pb2.py, used by the client. The command to do so is:
   - **protoc --python_out=. Lanz.proto**
   - The **--python_out=.** flag drops the compiled Python program in the directory where you ran the command.
6. Run **lanz_client.py -h** to activate the LANZ client.

20.2.4 LANZ Commands

LANZ Commands: Global Configuration

- clear queue-monitor length statistics
- queue-monitor length (global configuration mode)
- queue-monitor length cpu thresholds
- queue-monitor length fabric thresholds
- queue-monitor length global-buffer
- queue-monitor length global-buffer log
- queue-monitor length global-buffer thresholds
- queue-monitor length log
- queue-monitor length mirror
- queue-monitor length mirror destination
- queue-monitor length notifying
- queue-monitor length update-interval
- queue-monitor streaming

- tcpdump queue-monitor

LANZ Commands: Interface Ethernet Configuration Mode

- queue-monitor length threshold (Arad/Jericho/Qumran)
- queue-monitor length thresholds

LANZ Commands: Queue-Monitor Streaming Configuration Mode

- max-connections
- shutdown (queue-monitor-streaming configuration)

LANZ Display Commands

- show queue-monitor length
- show queue-monitor length all
- show queue-monitor length cpu
- show queue-monitor length csv
- show queue-monitor length drops
- show queue-monitor length ethernet
- show queue-monitor length global-buffer
- show queue-monitor length limit
- show queue-monitor length statistics
- show queue-monitor length status
- show queue-monitor length tx-latency
- show queue-monitor streaming clients

20.2.4.1 clear queue-monitor length statistics

The **clear queue-monitor length statistics** command resets the occurrences of all over-threshold events on the switch including global buffer information (if supported).

Command Mode

Privileged EXEC

Command Syntax

clear queue-monitor length statistics

Example

This command resets all over-threshold events and global buffer information on an FM6000 switch.

```
switch# clear queue-monitor length statistics
switch#
```

20.2.4.2 max-connections

The max-connections command sets the maximum number of client connections the switch accepts for streaming LANZ data. The default maximum is 10 connections. To stream LANZ data, you must use the **queue-monitor streaming** command to enable LANZ data streaming.

Command Mode

Queue-Monitor-Streaming Configuration

Command Syntax

max-connections *connections*

queue-monitor streaming

Parameter

connections maximum number of simultaneous LANZ streaming client connections the switch will accept. Values range from ***1*** through ***100***.

Example

This command sets the maximum number of client connections the switch accepts for LANZ data streaming to ***50***.

```
switch(config-qm-streaming)# max-connections 50
switch(config-qm-streaming)#
```

20.2.4.3 queue-monitor length (global configuration mode)

The **queue-monitor length (global configuration mode)** command enables LANZ with the current settings, or with the default settings if LANZ has not yet been configured. LANZ is enabled by default on Sand platform switches (those with Arad, Jericho, or Qumran chipsets), and disabled by default on all other switches.

When LANZ is enabled, the switch monitors queue lengths on all ports and generates over-threshold events when an output queue becomes congested. Over-threshold event data is available in the following forms:

- syslog data (see **queue-monitor length log**).
- CLI display or CSV-format output (see **show queue-monitor length**).
- data stream (see **queue-monitor streaming**).

The **no queue-monitor length** command entered in global configuration mode disables LANZ and discards LANZ log data, but retains settings. LANZ settings include:

- logging settings (see **queue-monitor length log**).
- queue length thresholds (see **queue-monitor length threshold (Arad/Jericho/Qumran)** or **queue-monitor length thresholds**).
- data streaming settings (see **queue-monitor streaming**).

To disable LANZ on an interface, use the **no queue-monitor length** command in interface configuration mode.

Command Mode

Global Configuration

Command Syntax

queue-monitor length

no queue-monitor length

default queue-monitor length

Examples

- This command enables LANZ on the switch.

```
switch(config)# queue-monitor length
switch(config)#
```

- This command disables LANZ on the switch and discards LANZ data but maintains configurations.

```
switch(config)# no queue-monitor length
switch(config)#
```

- This command disables LANZ on Ethernet interface 3/30.

```
switch(config)# interface ethernet 3/30
switch(config-if-Et3/30)# no queue-monitor length
switch(config-if-Et3/30)#
```

20.2.4.4 queue-monitor length cpu thresholds

The **queue-monitor length cpu thresholds** command sets the queue length threshold to define "congested" on all CPU ports for purposes of LANZ reporting. If LANZ is enabled (see **queue-monitor length (global configuration mode)**), an over-threshold event is generated when one or more queues on a CPU interface exceed the upper threshold, and over-threshold events continue to be generated until all queue lengths on the interface drop below the lower threshold. (To log these events, use the **queue-monitor length log** command.) Different monitoring thresholds cannot be set for individual CPU ports.

The **no queue-monitor length cpu thresholds** and **default queue-monitor length cpu thresholds** commands reset thresholds to the default values (high: ***512*** segments; low: ***256*** segments).

Command Mode

Global Configuration

Command Syntax

queue-monitor length cpu thresholds [***upper_limit*** | ***lower_limit***]

no queue-monitor length cpu thresholds

default queue-monitor length cpu thresholds

Parameters

- ***upper_limit*** is the queue length in segments that triggers an over-threshold event. Values range from ***8*** to ***16382***. Default setting is ***512***. Segment size varies by platform.
- ***lower_limit*** When logging is enabled, an over-threshold interface continues generating over-threshold events until all its queues drop back below this length. Must be lower than ***upper_limit***. Values range from ***1*** to ***16382***. Default setting is ***256***.

Examples

- This command sets the upper queue-length threshold for congestion monitoring on all CPU ports to ***1000*** segments and the lower limit to ***300*** segments.

```
switch(config)# queue-monitor length cpu thresholds 1000 300
switch(config)#
```

- This command resets the queue-length thresholds for CPU port congestion to the default values of ***512*** and ***256***.

```
switch(config)# default queue-monitor length cpu thresholds
switch(config)#
```

20.2.4.5 queue-monitor length fabric thresholds

The **queue-monitor length fabric thresholds** command sets the **queue-monitor length thresholds** to define "congested" on all fabric ports for purposes of LANZ reporting. If LANZ is enabled (see queue-monitor length (global configuration mode)), an over-threshold event is generated when one or more queues on a fabric interface exceed the upper threshold, and over-threshold events continue to be generated until all queue lengths on the interface drop below the lower threshold. (To log these events, use the **queue-monitor length log** command.) Different monitoring thresholds cannot be set for individual fabric ports.

The **no queue-monitor length fabric thresholds** and **default queue-monitor length fabric thresholds** commands reset thresholds to the default values (high: *512* segments; low: *256* segments).

Command Mode

Global Configuration

Command Syntax

queue-monitor length fabric thresholds [***upper_limit*** | ***lower_limit***

no queue-monitor length fabric thresholds

default queue-monitor length fabric thresholds

Parameters

- ***upper_limit*** is the queue length in segments that triggers an over-threshold event. Values range from ***8*** to ***16382***. Default setting is ***512***. Segment size varies by platform.
- ***lower_limit*** When logging is enabled, an over-threshold interface continues generating over-threshold events until all its queues drop back below this length. Must be lower than ***upper_limit***. Values range from ***1*** to ***16382***. Default setting is ***256***.

Examples

- This command sets the upper queue-length threshold for congestion monitoring on all fabric ports to ***1000*** segments and the lower limit to ***300*** segments.

```
switch(config)# queue-monitor length fabric thresholds 1000 300
switch(config)#
```

- This command resets the queue-length thresholds for fabric port congestion to the default values of ***512*** and ***256***.

```
switch(config)# default queue-monitor length fabric thresholds
switch(config)#
```

20.2.4.6 queue-monitor length global-buffer log

The **queue-monitor length global-buffer log** command enables logging of global buffer over-threshold events. When logging is enabled, a log message is generated each time the contents of the global buffer exceed the upper threshold value set for the switch (see **queue-monitor length global-buffer thresholds**). Once the global buffer is over the threshold, additional messages are generated at a maximum rate of one per interval as long as the buffer value remains above the lower threshold for the switch.

Global buffer logging is disabled by default.

Log messages do not include buffer usage or congestion information. To view this information, use the **show queue-monitor length global-buffer** command.

The **no queue-monitor length global-buffer log** and **default queue-monitor length global-buffer log** commands disable global buffer logging by removing the corresponding **queue-monitor length global-buffer log** command from ***running-config***. The **queue-monitor length global-buffer log** command with an interval value of ***0*** also disables global buffer logging.

Command Mode

Global Configuration

Command Syntax

queue-monitor length global-buffer log interval

no queue-monitor length global-buffer log

default queue-monitor length global-buffer log

Parameters

interval minimum interval in seconds between logged messages.

- ***0*** global buffer logging is disabled on the switch (the default setting).
- ***1*** to ***65535*** minimum logging interval (in seconds).

Guidelines

This command is available on FM6000 platform switches.

Examples

- This command enables global buffer logging with a minimum interval of ***10*** seconds between messages.

```
switch(config)# queue-monitor length global-buffer log 10
```

- This command disables global buffer logging on the switch.

```
switch(config)# no queue-monitor length global-buffer log
```

20.2.4.7 queue-monitor length global-buffer thresholds

The queue-monitor length global-buffer thresholds command sets global buffer thresholds for the switch. An over-threshold event is generated when usage of the global buffer exceeds the upper threshold, and over-threshold events continue to be generated until usage drops below the lower threshold. (To log these events, use the **queue-monitor length global-buffer log** command.)

The no queue-monitor length global-buffer and default queue-monitor length global-buffer commands disable global buffer reporting.

The **no queue-monitor length global-buffer thresholds** and **default queue-monitor length global-buffer thresholds** commands erase custom global buffer threshold settings.

Command Mode

Global Configuration

Command Syntax

queue-monitor length global-buffer thresholds ***max_segments min_segments***

no queue-monitor length global-buffer log

default queue-monitor length global-buffer log

Parameters

- ***max_segments*** upper threshold in 160-byte segments. Value ranges from ***2*** to ***36864***. Default is ***10940***.
- ***min_segments*** lower threshold in 160-byte segments. Value ranges from ***1*** to ***36864***. Default is ***4376***.

Examples

- This command sets the upper and lower global buffer thresholds to ***9000*** segments and ***3000*** segments.

```
switch(config)# queue-monitor length global-buffer thresholds 9000 3000
switch(config)#
```

- This command resets the upper and lower global buffer thresholds to their default values.

```
switch(config)# no queue-monitor length global-buffer thresholds 9000 3000
switch(config)#
```

20.2.4.8 queue-monitor length global-buffer

The **queue-monitor length global-buffer** command includes global buffer usage in LANZ reporting.

When global buffer reporting is enabled, over-threshold events are generated when global buffer usage exceeds the upper threshold. To set the threshold value, use the **queue-monitor length global-buffer thresholds** command. Usage data may be viewed using the **show queue-monitor length global-buffer** command. To view status and threshold information, use the **show queue-monitor length status** command.

Global buffer usage is measured in segments of ***160*** bytes.

The **no queue-monitor length global-buffer** and **default queue-monitor length global-buffer** commands disable global buffer usage reporting by removing the corresponding **queue-monitor length global-buffer** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

queue-monitor length global-buffer

no queue-monitor length global-buffer

default queue-monitor length global-buffer

Guidelines

This command is available on FM6000 platform switches.

Examples

- This command enables global buffer monitoring on the switch.

```
switch(config)# queue-monitor length global-buffer
switch(config)#
```

- This command disables global buffer monitoring on the switch.

```
switch(config)# no queue-monitor length global-buffer
switch(config)#
```

20.2.4.9 queue-monitor length log

The **queue-monitor length log** command enables logging of queue-length over-threshold events when LANZ is enabled on the switch (see **queue-monitor length (global configuration mode)**). When logging is enabled, a log message is generated each time one or more queues on an interface exceed the upper threshold value for that interface (see **queue-monitor length threshold (Arad/Jericho/Qumran)**). Once an interface is over threshold, additional messages are generated at a maximum rate of one per interval as long as the queue length remains above the lower threshold for that interface. No syslog message is generated when queue length drops back under threshold.

Logging is disabled by default.

Log messages do not include queue length information. To view queue length information, use the **show queue-monitor length** command.

The queue-monitor length log command with an interval value of 0 disables event logging.

Command Mode

Global Configuration

Command Syntax

queue-monitor length log interval

Parameters

interval minimum interval in seconds between logged messages from a single interface.

- ***0*** queue-length logging is disabled on the switch (the default setting).
- ***1*** to ***65535*** minimum logging interval (in seconds).

Examples

- This command enables over-threshold logging with a minimum interval of ***10*** seconds between messages for a given interface.

```
switch(config)# queue-monitor length log 10
```

- This command disables queue-length over-threshold logging on the switch.

```
switch(config)# queue-monitor length log 0
```

- This is an example of a queue-length log message.

```
Oct 27 12:48:22 switch QUEUE_MONITOR-6-LENGTH_OVER_THRESHOLD: Interface
Ethernet6 queue length is over threshold of 512, current length is 1024.
```

20.2.4.10 queue-monitor length mirror

The **queue-monitor length mirror** command enables LANZ mirroring. As a result, the switch is configured to automatically send congested traffic to either the CPU or an Ethernet egress interface destination, once a queue threshold is crossed. To set the destination for the mirrored traffic, use the **queue-monitor length mirror destination** command.

Command Mode

Global Configuration

Command Syntax

queue-monitor length mirror

Examples

- This command enables LANZ traffic sampling.

```
switch(config)# queue-monitor length mirror
switch(config)#
```

- This command disables LANZ traffic sampling.

```
switch(config)# no queue-monitor length mirror
switch(config)#
```

20.2.4.11 queue-monitor length mirror destination

The **queue-monitor length mirror destination** command results in automatically sending traffic experiencing congestion to either the CPU or an Ethernet egress interface destination, once a queue threshold is crossed. Before using this command, first enable LANZ mirroring through the command **queue-monitor length mirror**.

Command Mode

Global Configuration

Command Syntax

queue-monitor length mirror destination cpu | *ports*

Parameter

ports any combination of Ethernet ports ***1*** through ***24***.

Examples

- This command configures LANZ traffic sampling for a CPU interface mirror destination.

```
switch(config)# queue-monitor length mirror destination cpu
switch(config)#
```

- This command configures LANZ traffic sampling for an Ethernet interface mirror destination for ports ***3***, ***11***, and ***15*** through ***20***.

```
switch(config)# queue-monitor length mirror destination Ethernet 3,11,15-20
switch(config)#
```

20.2.4.12 queue-monitor length notifying

The **queue-monitor length notifying** command enables Notifying Mode on Arad and Jericho switches. By default, LANZ operates in Polling Mode on Arad and Jericho switches. On all other switches, LANZ operates only in Notifying Mode.

When Notifying Mode is enabled, the switch provides detailed congestion information including Start, Update, and End events, rather than only polling the most congested queue in each ASIC. Notifying Mode uses both upper and lower threshold values. Both can be set with the **queue-monitor length thresholds** command. While a queue is congested, the maximum queue size is updated every five seconds by default; this interval can be configured using the **queue-monitor length update-interval** command.

The **no queue-monitor length notifying** and **default queue-monitor length notifying** commands reset LANZ to the default Polling Mode.

Command Mode

Global Configuration

Command Syntax

queue-monitor length notifying

no queue-monitor length notifying

default queue-monitor length notifying

Guidelines

- Polling mode is available only on Arad and Jericho platform switches.
- On Jericho platforms, LANZ will revert to Polling Mode if there are not enough counter resources to operate in Notifying Mode.
- On Arad platforms, Notifying Mode is incompatible with SSO. Enabling SSO while Notifying Mode is enabled will cause LANZ to revert to Polling Mode.
- On Arad platforms, Notifying Mode is not available for CPU queues. Use Polling Mode when monitoring congestion on CPU queues on Arad switches.
- If the switch is rebooted while Notifying Mode is enabled, queue threshold values may be lost.

Examples

- This command enables Notifying Mode on an Arad or Jericho switch.

```
switch(config)# queue-monitor length notifying
switch(config)#
```

- This command disables Notifying Mode on an Arad or Jericho switch, returning LANZ to the default Polling Mode.

```
switch(config)# no queue-monitor length notifying
switch(config)#
```

20.2.4.13 queue-monitor length thresholds

The **queue-monitor length thresholds** command sets both upper and lower queue length thresholds to define “congested” on the command-mode interface for purposes of LANZ reporting. If LANZ is enabled (see **queue-monitor length (global configuration mode)**), an over-threshold event is generated when one or more queues on the interface exceed the upper threshold, and over-threshold events continue to be generated until all queue lengths on the interface drop below the lower threshold. (To log these events, use the **queue-monitor length log** command.)

Entering the **no queue-monitor length** command in interface configuration mode disables LANZ on the interface. Entering either the **queue-monitor length** command or the **default queue-monitor length** command in interface configuration mode enables LANZ on the interface by removing the **no queue-monitor length** command from the configuration.

The **no queue-monitor length thresholds** and **default queue-monitor length thresholds** commands in interface configuration mode both erase custom queue length threshold settings for the interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

queue-monitor length thresholds ***upper_limit lower_limit***

no queue-monitor length

default queue-monitor length

Parameters

- ***upper_limit*** queue length in segments that triggers an over-threshold event. Must be higher than ***lower_limit***. The minimum value is *2*. The maximum is the largest number of segments which can be queued before packets are dropped, and varies based on factors including flow control state and private buffer settings. Default setting is ***512***.
- ***lower_limit*** lower queue length threshold in segments. When logging is enabled, an over-threshold interface continues generating over-threshold events until all its queues drop back below this length. Must be lower than ***upper_limit***. Values range from ***1*** to ***4806***. Default setting is ***256***.

Guidelines

Queue lengths are measured in 480-byte segments on FM6000 switches, in 208-byte segments on Trident II and Tomahawk switches, in 256-byte segments on Trident 3 switches, and in bytes on Arad and Jericho switches. Default thresholds vary by platform. Both upper and lower thresholds are configurable.

Examples

- These commands set the upper and lower queue-length thresholds on ***interface ethernet 5*** to ***300*** segments and ***200*** segments.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# queue-monitor length thresholds 300 200
switch(config-if-Et5)#
```

- These commands reset the upper and lower queue-length thresholds on ***interface ethernet 5*** to their default values.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# default queue-monitor length thresholds
switch(config-if-Et5)#
```

20.2.4.14 queue-monitor length threshold (Arad/Jericho/Qumran)

The **queue-monitor length threshold** command sets the upper queue-length threshold on Arad, Jericho, and Qumran platforms to define “congested” on the command-mode interface for purposes of LANZ reporting. If LANZ is enabled (see **queue-monitor length (global configuration mode)**), an over-threshold event is generated when one or more queues on the interface exceed the upper threshold, and over-threshold events continue to be generated until all queue lengths on the interface drop below the lower threshold. (To log these events, use the **queue-monitor length log** command.)

Entering the **no queue-monitor length** command in interface configuration mode disables LANZ on the interface. Entering either the **queue-monitor length threshold** command or the **default queue-monitor length threshold** command enables LANZ on the interface by removing the **no queue-monitor length** command from the configuration.

The **no queue-monitor length threshold** and **default queue-monitor length threshold** commands erase custom queue length threshold settings for the interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

queue-monitor length threshold ***upper_limit***

no queue-monitor length

default queue-monitor length

Parameters

upper_limit is the queue length in bytes that triggers an over-threshold event. Values range from ***40962*** to ***52428800*** bytes. Default setting is ***52428800***.

Guidelines

On Arad, Jericho, and Qumran platforms, the queue length is measured in bytes. Only the upper threshold is configurable using this command, and it is set at a default value of ***52428800*** bytes. Both upper and lower thresholds can be set using the **queue-monitor length thresholds** command.

Examples

- These commands set the upper queue-length threshold on ***interface ethernet 3/30*** to ***40000000*** bytes.

```
switch(config)# interface ethernet 3/30
switch(config-if-Et3/30)# queue-monitor length threshold 40000000
switch(config-if-Et3/30)#
```

- These commands reset the upper queue-length threshold on ***interface ethernet 3/30*** to its default value of ***52428800*** bytes.

```
switch(config)# interface ethernet 3/30
switch(config-if-Et3/30)# default queue-monitor length threshold
switch(config-if-Et3/30)#
```

20.2.4.15 queue-monitor length update-interval

The **queue-monitor length update-interval** command sets the interval between congestion updates when LANZ is in Notifying Mode.

The **no queue-monitor length update-interval** and **default queue-monitor length update-interval** commands reset the update interval to its default value of ***5000000*** (***5*** seconds).

Command Mode

Global Configuration

Command Syntax

queue-monitor length update-interval ***interval***

no queue-monitor length

default queue-monitor length

Parameters

interval is the time in microseconds between congestion updates. Values range from ***80***-***10000000***; default setting is ***5000000*** (***5*** seconds).

Examples

- This command sets the time between congestion updates to ***10*** seconds.

```
switch(config)# queue-monitor length update-interval 10000000
switch(config)#
```

- This command resets the time between congestion updates to its default value of ***5*** seconds.

```
switch(config)# default queue-monitor length update-interval
switch(config)#
```

20.2.4.16 queue-monitor streaming

The **queue-monitor streaming** command places the switch in queue-monitor-streaming configuration mode. Queue-monitor-streaming configuration mode is not a group change mode; ***running-config*** is changed immediately upon command entry. The exit command does not affect ***running-config***.

To enable LANZ data streaming on the switch, use the **no show queue-monitor streaming clients** command.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

queue-monitor streaming

Commands Available in queue-monitor streaming Configuration Mode

- **max-connections**
- **show queue-monitor streaming clients**

Example

This command places the switch in the ***queue-monitor streaming*** configuration mode.

```
switch(config)# queue-monitor streaming
switch(config-qm-streaming)#
```

20.2.4.17 show queue-monitor length

The **`show queue-monitor length`** command displays a report of recent over-threshold events for all interfaces, limited to the last ***1000*** records, with the newest events listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

To limit the output to a specified number of seconds and/or records, use the **show queue-monitor length limit** command.

Command Mode

EXEC

Command Syntax

`show queue-monitor length`

Examples

- This command displays the last ***1000*** LANZ records on an Arad or Jericho platform switch in Polling Mode.

```
switch> show queue-monitor length
Report generated at 2017-03-10 16:04:28
E-End, S-Start, P-Polling, TC-Traffic Class
* Max queue length during period of congestion
Type       Time                    Intf(TC)      Queue       Duration    Ingress
                                                 Length                  Port-set
                                                 (bytes)     (usecs)
---------- ----------------------- ------------- ----------- ----------- --------
P          0:00:04.81587 ago       Et15(3)       36126720     4030092     Et1-24 switch>
```

- This command displays the last ***1000*** LANZ records on an Arad or Jericho platform switch in Notifying Mode.

```
switch> show queue-monitor length
Report generated at 2017-03-10 16:08:58
E-End, S-Start, P-Polling, TC-Traffic Class
* Max queue length during period of congestion
Type       Time                    Intf(TC)       Queue       Duration    Ingress
                                                  Length                  Port-set
                                                  (bytes)     (usecs)
---------- ----------------------- -------------- ----------- ----------  --------
E          0:00:03.11739 ago       Et24(2)        36126720*  20700629    Et9-20,54/1-4
U          0:00:04.01513 ago       Et24(2)        36126720    N/A         Et9-20,54/1-4
U          0:00:08.94918 ago       Et24(2)        36126720    N/A         Et9-20,54/1-4
U          0:00:13.88323 ago       Et24(2)        36126720    N/A         Et9-20,54/1-4
U          0:00:18.81728 ago       Et24(2)        36126720    N/A         Et9-20,54/1-4
U          0:00:23.74758 ago       Et24(2)        36126720    N/A         Et9-20,54/1-4
S          0:00:23.81802 ago       Et24(2)        36126720    N/A         Et9-20,54/1-4 switch>
```

- This command displays the last ***1000*** LANZ records on an FM6000, Trident II, Trident 3, or Tomahawk platform switch.

```
switch> show queue-monitor length
Report generated at 2017-03-10 14:57:12
E-End, U-Update, S-Start, TC-Traffic Class
Segment size for E, U and S congestion records is 208 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-------------------------------------------------------------------------------------------
Type    Time               Interface      Congestion      Queue      Time of Max   Fabric
                           (TC)           duration        length     Queue length  Peer
                                          (usecs)         (segments) relative to
                                                                      congestion
                                                                      start
                                                                      (usecs)
-------------------------------------------------------------------------------------------
 E   0:08:04.45352 ago     Et23/3(13)     22704753        5743*      0
 U   0:08:07.10807 ago     Et23/3(13)     N/A             5742       N/A
 U   0:08:12.10808 ago     Et23/3(13)     N/A             5742       N/A
 U   0:08:17.10809 ago     Et23/3(13)     N/A             5742       N/A
 U   0:08:22.10810 ago     Et23/3(13)     N/A             5742       N/A
```

```
U   0:08:27.10810 ago     Et23/3(13)     N/A            5742       N/A
U   0:08:27.11311 ago     Et23/3(13)     N/A            5741       N/A
U   0:08:27.11811 ago     Et23/3(13)     N/A            5742       N/A
U   0:08:27.12312 ago     Et23/3(13)     N/A            5742       N/A
U   0:08:27.12812 ago     Et23/3(13)     N/A            5743       N/A
U   0:08:27.13315 ago     Et23/3(13)     N/A            5743       N/A
U   0:08:27.13816 ago     Et23/3(13)     N/A            5743       N/A
U   0:08:27.14319 ago     Et23/3(13)     N/A            5742       N/A
U   0:08:27.14822 ago     Et23/3(13)     N/A            5743       N/A
U   0:08:27.15322 ago     Et23/3(13)     N/A            5742       N/A
S   0:08:27.15828 ago     Et23/3(13)     N/A            2064       N/A switch>
```

20.2.4.18 show queue-monitor length all

The **show queue-monitor length all** command displays all available over-threshold event records on the switch including global buffer information, with the most recent events listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

show queue-monitor length all

Guidelines

This command is available on FM6000 platform switches.

Example

This command displays all available LANZ records from the switch.

```
switch> show queue-monitor length all

Report generated at 2013-04-01 13:23:13
E-End, U-Update, S-Start, TC-Traffic Class
GH-High, GU-Update, GL-Low
Segment size for E, U and S congestion records is 480 bytes
Segment size for GL, GU and GH congestion records is 160 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-------------------------------------------------------------------------------
Type    Time                   Intf    Congestion     Queue       Time of Max
                               (TC)    duration       length      Queue length
                                       (usecs)        (segments)  relative to
                                                                  congestion
                                                                  start
                                                                  (usecs)
-------------------------------------------------------------------------------
E  0:00:00.07567 ago         Et22(7)   >=71 mins      20*         30us
GU 0:00:00.15325 ago         N/A       N/A            5695        N/A
U  0:00:00.19859 ago         Et4(1)    N/A            5693        N/A
GU 0:00:00.95330 ago         N/A       N/A            5696        N/A
U  0:00:00.99859 ago         Et4(1)    N/A            5695        N/A
E  0:00:01.28821 ago         Et44(1)   9672us         2502*       7294us
S  0:00:01.17591 ago         Et22(7)   N/A            26          N/A
U  0:00:03.08248 ago         Et44(1)   N/A            50          N/A
S  12days,8:56:44.07567 ago  Et44(1)   N/A            20          N/A switch>
```

20.2.4.19 show queue-monitor length cpu

The **show queue-monitor length cpu** command displays LANZ data for CPU ports on the switch. On Trident II and Tomahawk platforms, the “Interface” column identifies the CPU port by its card slot and chip index.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

show queue-monitor length cpu

- This command displays LANZ data for CPU ports on a Trident II or Tomahawk switch.

```
switch> show queue-monitor length cpu
Report generated at 2017-03-10 15:24:11
E-End, U-Update, S-Start, TC-Traffic Class
Segment size for E, U and S congestion records is 208 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-------------------------------------------------------------------------------------------
Type    Time             Interface      Congestion     Queue      Time of Max    Fabric
                         (TC)           duration       length     Queue length   Peer
                                        (usecs)        (segments) relative to
                                                                  congestion
                                                                  start
                                                                  (usecs)
-------------------------------------------------------------------------------------------
 E   0:02:24.19153 ago    Cpu0/0(39)     16669811       271*       0
 U   0:02:25.81126 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:30.81126 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:35.81128 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.81129 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.81630 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.82130 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.82631 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.83132 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.83632 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.84132 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.84633 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.85133 ago    Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.85634 ago    Cpu0/0(39)     N/A            270        N/A
 S   0:02:40.86134 ago    Cpu0/0(39)     N/A            271        N/A
```

- This command displays LANZ data for CPU ports on an FM6000 switch.

```
switch> show queue-monitor length cpu
Report generated at 2017-03-10 15:17:57
E-End, U-Update, S-Start, TC-Traffic Class
GH-High, GU-Update, GL-Low
Segment size for E, U and S congestion records is 480 bytes
Segment size for GL, GU and GH congestion records is 160 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-------------------------------------------------------------------------------
Type    Time                  Intf    Congestion     Queue       Time of Max
                              (TC)    duration       length      Queue length
                                      (usecs)        (segments)  relative to
                                                                 congestion
                                                                 start
                                                                 (usecs)
-------------------------------------------------------------------------------
 E   0:04:41.17456 ago         Cpu(11)   29           1024*       0
 S   0:04:41.17459 ago         Cpu(11)   N/A          1024        N/A
 E   0:04:41.17463 ago         Cpu(11)   15926108     206*        33872
 U   0:04:42.09651 ago         Cpu(11)   N/A          205         N/A
 U   0:04:47.09710 ago         Cpu(11)   N/A          205         N/A
 U   0:04:52.09769 ago         Cpu(11)   N/A          205         N/A
 U   0:04:57.09826 ago         Cpu(11)   N/A          205         N/A
```

- This command displays LANZ data for CPU ports on an Arad or Jericho switch in Polling Mode.

```
switch> show queue-monitor length cpu
Report generated at 2017-03-10 16:04:28
E-End, S-Start, P-Polling, TC-Traffic Class
* Max queue length during period of congestion
Type  Time              Intf(TC)              Queue      Duration   Ingress
                                              Length                Port-set
                                              (bytes)    (usecs)
---- ------------------ --------------------- ---------- ---------- --------
```

```
P    0:00:31.48474 ago  CoppSystemL2Ucast(5)    10486080  20184965    Et1-24
```

- This command displays LANZ data for CPU ports on an Arad or Jericho switch in Notifying Mode.

```
switch> show queue-monitor length cpu
Report generated at 2017-03-10 16:08:58
E-End, S-Start, P-Polling, TC-Traffic Class
* Max queue length during period of congestion
Type  Time               Intf(TC)                Queue       Duration   Ingress
                                                 Length                 Port-set
                                                 (bytes)     (usecs)
----- ------------------ ----------------------- ----------- ---------- --------
E     0:00:03.11739 ago    CoppSystemL2Ucast(5)   10485760*   20700629   Et1-24
U     0:00:04.01513 ago    CoppSystemL2Ucast(5)   10485760     N/A       Et1-24
U     0:00:08.94918 ago    CoppSystemL2Ucast(5)   10485760     N/A       Et1-24
U     0:00:13.88323 ago    CoppSystemL2Ucast(5)   10485760     N/A       Et1-24
U     0:00:18.81728 ago    CoppSystemL2Ucast(5)   10485760     N/A       Et1-24
U     0:00:23.74758 ago    CoppSystemL2Ucast(5)   10485760     N/A       Et1-24
S     0:00:23.81802 ago    CoppSystemL2Ucast(5)   10485760     N/A       Et1-24
```

20.2.4.20 show queue-monitor length csv

The **`show queue-monitor length csv`** command displays LANZ records in Comma-Separated Value (CSV) format with the oldest samples displayed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

`show queue-monitor length csv`

Example

This command displays LANZ records in CSV format.

```
switch> show queue-monitor length csv

Report generated at 2016-02-09 22:57:50
Type,Time,Interface,Duration(usecs),Queue-Length,Time-Of-Max-Queue(usecs),Laten
cy(usecs),Tx-Drops
 S,2016-02-09 22:53:05.70596,Et29(11),N/A,2590,N/A,60.088,0
 U,2016-02-09 22:53:05.71098,Et29(11),N/A,2590,N/A,60.088,216555
 U,2016-02-09 22:53:05.71600,Et29(11),N/A,2590,N/A,60.088,215546 switch>
```

20.2.4.21 show queue-monitor length drops

The **`show queue-monitor length drops`** command displays a report of cumulative transmission drop totals for a range of interfaces or for all interfaces. Output can be limited to a specified number of seconds or records. The most recent events are listed first. By default, the command displays data for all interfaces, limited to the last 1000 records. Newest events are listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

`show queue-monitor length` [**INTERFACES**][**FACTOR**] **`drops`**

Parameters

- **INTERFACES** interface type and number for report. Values include:
  - ***no parameter*** displays information for all interfaces.
  - ***ethernet e-range*** e-range formats include a number, number range, or comma-delimited list of numbers and ranges.
- **FACTOR** limiting parameter for report. Values include:
  - ***no parameter*** displays the last ***1000*** records.
  - **limit number samples** displays the last ***number*** records.
  - **limit number seconds** displays all records generated during the last ***number*** seconds. Value of number ranges from ***1*** to ***1000000***.

Guidelines

This command is available on FM6000, Trident II, Trident 3, and Tomahawk platform switches.

Examples

- This command displays the last ***10*** records of transmission drop data on an FM6000 switch.

```
switch> show queue-monitor length limit 10 samples drops
Report generated at 2017-03-10 15:25:29
Time                                    Interface      TX Drops
--------------------------------------------------------------------
0:12:13.34425 ago                       Cpu            419
0:12:13.34428 ago                       Cpu            371
0:12:13.34433 ago                       Cpu            9913826
0:12:14.26621 ago                       Cpu            53775812
0:12:19.26680 ago                       Cpu            53775740
0:12:24.26738 ago                       Cpu            53775714
0:12:29.26796 ago                       Cpu            1073
0:12:29.26806 ago                       Cpu            1068
0:12:29.26816 ago                       Cpu            1074
0:12:29.26825 ago                       Cpu            1071
```

- This command displays the last ***10*** records of transmission drop data on a Trident II or Tomahawk switch.

```
switch> show queue-monitor length limit 10 samples drops

Report generated at 2017-03-10 15:25:34
Time                                    Interface(TC)  TX Drops
--------------------------------------------------------------------
0:03:47.06553 ago                       Cpu0/0(39)     17419818
0:03:48.68527 ago                       Cpu0/0(39)     53773707
0:03:53.68527 ago                       Cpu0/0(39)     53773770
0:03:58.68528 ago                       Cpu0/0(39)     53773763
0:04:03.68529 ago                       Cpu0/0(39)     53878
0:04:03.69030 ago                       Cpu0/0(39)     53777
0:04:03.69530 ago                       Cpu0/0(39)     53917
0:04:03.70031 ago                       Cpu0/0(39)     53880
0:04:03.70532 ago                       Cpu0/0(39)     53786
0:04:03.71032 ago                       Cpu0/0(39)     53782 switch>
```

20.2.4.22 show queue-monitor length ethernet

The **`show queue-monitor length ethernet`** command displays a report of recent over-threshold events for a range of interfaces, with the newest events listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

`show queue-monitor length ethernet` ***e-range***

Parameters

e-range the range of interfaces to be included in the report; formats include a number, number range, or comma-delimited list of numbers and ranges

Examples

- This command displays the last ***1000*** records for ***interface ethernet 9*** on an FM6000 platform switch.

```
switch> show queue-monitor length ethernet 9
Report generated at 2017-03-10 15:33:35
E-End, U-Update, S-Start, TC-Traffic Class
GH-High, GU-Update, GL-Low
Segment size for E, U and S congestion records is 480 bytes
Segment size for GL, GU and GH congestion records is 160 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
--------------------------------------------------------------------------------
Type    Time                    Intf    Congestion     Queue       Time of Max
                                (TC)    duration       length      Queue length
                                        (usecs)        (segments)  relative to
                                                                   congestion
                                                                   start
                                                                   (usecs)
--------------------------------------------------------------------------------
 E   0:00:24.82515 ago         Et9(1)    22737967      5623*       26651
 U   0:00:27.55841 ago         Et9(1)    N/A           5620        N/A
 U   0:00:32.55899 ago         Et9(1)    N/A           5620        N/A
 U   0:00:37.55957 ago         Et9(1)    N/A           5621        N/A
 U   0:00:42.56015 ago         Et9(1)    N/A           5621        N/A
 U   0:00:47.56073 ago         Et9(1)    N/A           5621        N/A
 U   0:00:47.56083 ago         Et9(1)    N/A           5620        N/A
 U   0:00:47.56093 ago         Et9(1)    N/A           5620        N/A
 U   0:00:47.56103 ago         Et9(1)    N/A           5620        N/A
 U   0:00:47.56113 ago         Et9(1)    N/A           5620        N/A switch>
```

- This command displays the last ***1000*** records for ***interface ethernet 23/3*** on a Trident II or Tomahawk platform switch.

```
switch> show queue-monitor length Ethernet 23/3
Report generated at 2017-03-10 15:38:01
E-End, U-Update, S-Start, TC-Traffic Class
Segment size for E, U and S congestion records is 208 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
--------------------------------------------------------------------------------------------------
Type    Time                    Interface      Congestion     Queue      Time of Max    Fabric
                                (TC)           duration       length     Queue length   Peer
                                               (usecs)        (segments) relative to
                                                                         congestion
                                                                         start
                                                                         (usecs)
--------------------------------------------------------------------------------------------------
 E   0:00:29.49376 ago         Et23/3(1)      22388908       7879*      365268
 U   0:00:31.83332 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:36.83288 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:41.83289 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:46.83289 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:51.83260 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:51.83760 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:51.84261 ago         Et23/3(1)      N/A            7878       N/A
 U   0:00:51.84762 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:51.85263 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:51.85763 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:51.86264 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:51.86765 ago         Et23/3(1)      N/A            7878       N/A
 U   0:00:51.87265 ago         Et23/3(1)      N/A            7877       N/A
 U   0:00:51.87766 ago         Et23/3(1)      N/A            7877       N/A
 S   0:00:51.88267 ago         Et23/3(1)      N/A            1710       N/A
switch>
```

- This command displays the last ***1000*** records for ***interface ethernet 24*** on an Arad or Jericho platform switch in Polling Mode.

```
switch> show queue-monitor length Ethernet 24
Report generated at 2017-03-10 16:04:28
E-End, S-Start, P-Polling, TC-Traffic Class
```

```
* Max queue length during period of congestion
Type       Time                    Intf(TC)        Queue          Duration    Ingress
                                                   Length                     Port-set
                                                   (bytes)        (usecs)
---------- ----------------------- --------------- -------------- ----------- --------
P          0:00:04.81587 ago       Et24(3)         36126720       4030092     Et1-24 switch>
```

- This command displays the last ***1000*** records for ***interface ethernet 24*** on an Arad or Jericho platform switch in Notifying Mode.

```
switch> show queue-monitor length Ethernet 24
Report generated at 2017-03-10 16:08:58
E-End, S-Start, P-Polling, TC-Traffic Class
* Max queue length during period of congestion
Type       Time                    Intf(TC)     Queue          Duration    Ingress
                                                Length                     Port-set
                                                (bytes)        (usecs)
---------- ----------------------- ------------ -------------- ----------- --------
E          0:00:03.11739 ago       Et24(2)      36126720*      20700629    Et9-20,54/1-4
U          0:00:04.01513 ago       Et24(2)      36126720       N/A         Et9-20,54/1-4
U          0:00:08.94918 ago       Et24(2)      36126720       N/A         Et9-20,54/1-4
U          0:00:13.88323 ago       Et24(2)      36126720       N/A         Et9-20,54/1-4
U          0:00:18.81728 ago       Et24(2)      36126720       N/A         Et9-20,54/1-4
U          0:00:23.74758 ago       Et24(2)      36126720       N/A         Et9-20,54/1-4
S          0:00:23.81802 ago       Et24(2)      36126720       N/A         Et9-20,54/1-4
 switch>
```

Note: The non-sequential listing of ingress ports shown here is specific to Jericho switches; interfaces on Arad switches are always displayed sequentially.

20.2.4.23 show queue-monitor length global-buffer

The **`show queue-monitor length global-buffer`** command displays a report of recent high usage, low usage and update events for the global buffer. Newest events are listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

`show queue-monitor length global-buffer`

Guidelines

This command is available on FM6000 platform switches.

Example

This command displays the global buffer event records for the switch.

```
switch> show queue-monitor length global buffer
Report generated at 2013-04-01 14:30:07
GH-High, GU-Update, GL-Low
Segment size = 160 bytes
* Max buffer usage during period of congestion
------------------------------------------------------------------------------
-
Type       Time                      Buffer      Congestion     Time of Max
                                     usage       duration       buffer usage
                                     (segments)  (usecs)        relative to
                                                                GH (usecs)
------------------------------------------------------------------------------
-
GE 0:04:04.49547 ago                 3121*       20786516       3418
GU 0:04:05.27967 ago                 3120        N/A            N/A
GU 0:04:10.27968 ago                 3120        N/A            N/A
GU 0:04:25.28163 ago                 3118        N/A            N/A
GU 0:04:25.28173 ago                 3118        N/A            N/A
GU 0:04:25.28182 ago                 2963        N/A            N/A
```

```
GU 0:04:25.28192 ago                  1916        N/A           N/A
GS 0:04:25.28201 ago                  913         N/A           N/A switch>
```

20.2.4.24 show queue-monitor length limit

The **show queue-monitor length limit** command displays a report of recent over-threshold events for a range of interfaces or for all interfaces, limited by a specified number of records.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

show queue-monitor length limit [**INTERFACES**] ***number***

Parameters

- **INTERFACES** interface type and number for report. Values include:
  - ***no parameter*** displays information for all interfaces.
  - **ethernet *e-range*** e-range formats include a number, number range, or comma-delimited list of numbers and ranges.
- ***number*** number of records to display. Values range from ***1*** to ***1000000***.

Example

This command displays the last ***100*** records for interface ethernet ***6*** through ***8***.

```
switch># show queue-monitor length ethernet 6-8 limit 100 samples
Report generated at 2010-01-01 12:56:13
Time                        Interface   Queue length (segments, 1 to 512 bytes)
--------------------------------------------------------------------------------
0:00:07.43393 ago           Et6         1049
0:00:39.22856 ago           Et7         2039
1 day, 4:33:23.12345 ago    Et6         1077 switch>
```

20.2.4.25 show queue-monitor length statistics

The **show queue-monitor length statistics** command displays LANZ statistics for all interfaces, showing the traffic class and number of recorded congestion events for each interface.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

show queue-monitor length statistics

Example

This command displays LANZ statistics for all interfaces on the switch.

```
switch> show queue-monitor length statistics

Report generated at 2016-02-09 22:59:56
Interface      Traffic Class  Count
-------------------------------------------------------
Et29            11             1
```

20.2.4.26 show queue-monitor length status

The **show queue-monitor length status** command displays the current LANZ configuration for the switch and for each interface. On certain platforms, the status of global buffer monitoring and per-linecard LANZ mode are displayed.

Command Mode

EXEC Global Configuration

Command Syntax

show queue-monitor length status

Guidelines

On FM6000 platform switches, this command includes status information about global buffer monitoring.

On Arad and Jericho based linecards, if LANZ is globally enabled (using the **queue-monitor length (global configuration mode)** command), the command displays the monitoring mode per-linecard.

On Arad and Jericho switches, even when all linecards are configured in Notifying Mode (using the **queue-monitor length notifying** command), some linecards may still run LANZ in Polling Mode under the following circumstances:

- On Arad-based linecards, LANZ falls back to Polling Mode when SSO redundancy mode is configured on the card (here, the mode is shown as “polling due to SSO configured”).
    - On Jericho-based linecards, LANZ may run in Polling Mode when many features that use the switch’s statistic capabilities have been configured (here, the mode is shown as “polling due to counters exhausted”).

Examples

- This command displays the current LANZ configuration on an FM6000 device.

```
switch# show queue-monitor length status
queue-monitor length enabled
queue-monitor length packet sampling is disabled
queue-monitor length update interval in micro seconds:  5000000
Global Buffer Monitoring
------------------------
Global buffer monitoring is enabled
Segment size in bytes :   160
Total buffers in segments : 49152
High threshold : 16862
Low threshold :  6745


Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Segment size in bytes :   480
Maximum queue length in segments :  5621
Port thresholds in segments:
Port     High threshold  Low threshold   Mirroring Enabled
Cpu                512            256                True
Et1                512            256                True
Et2                512            256                True
Et3                512            256                True
Et4                512            256                True
Et5                512            256                True
Et6                512            256                True
Et7                512            256                True


switch#
```

- This command displays the current LANZ configuration on a Trident II or Tomahawk device.

```
switch# show queue-monitor length status
queue-monitor length enabled
```

```
queue-monitor length packet sampling is disabled
queue-monitor length update interval in micro seconds:  5000000
Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Segment size in bytes :   208
Maximum queue length in segments : 16382
Port thresholds in segments:
Port     High threshold  Low threshold
Cpu                512            256
Et1/1              512            256
Et1/2              512            256
Et1/3              512            256
Et1/4              512            256
Et2/1              512            256
Et2/2              512            256
Et2/3              512            256 switch#
```

- This command displays the current LANZ configuration on an Arad or Jericho device.

```
switch# show queue-monitor length status
queue-monitor length enabled
queue-monitor length packet sampling is disabled
queue-monitor length update interval in micro seconds: 5000000
Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Queue length monitoring mode is notifying
Queue length monitoring status is:
Linecard3 polling due to SSO configured
Linecard4 polling due to SSO configured
Linecard5 polling due to SSO configured
Linecard6 polling due to SSO configured
Maximum queue length in bytes : 524288000
Port thresholds in bytes:
Port     High threshold    Low threshold     Warnings
Cpu               65536          32768
Et3/1/1          5242880        2621440
Et3/1/2          5242880        2621440
Et3/1/3          5242880        2621440
Et3/1/4          5242880        2621440
Et3/1/5          5242880        2621440
Et3/1/6          5242880        2621440
Et3/1/7          5242880        2621440
Et3/1/8          5242880        2621440
```

- This command displays the current LANZ configuration on an Arad or Jericho device with LANZ disabled globally. Per-linecard LANZ mode status is not displayed.

```
switch(config)# show queue-monitor length status
queue-monitor length disabled
queue-monitor length packet sampling is disabled
queue-monitor length update interval in micro seconds: 5000000
Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is disabled
Queue length monitoring mode is notifying
Maximum queue length in bytes : 524288000
Port thresholds in bytes:
Port     High threshold    Low threshold     Warnings
Cpu               65536          32768
Et3/1/1          5242880        2621440
Et3/1/2          5242880        2621440
Et3/1/3          5242880        2621440
Et3/1/4          5242880        2621440
Et3/1/5          5242880        2621440
Et3/1/6          5242880        2621440
Et3/1/7          5242880        2621440
Et3/1/8          5242880        2621440
```

20.2.4.27 show queue-monitor length tx-latency

The **show queue-monitor length tx-latency** command displays the latency data of recent LANZ events for a range of interfaces or for all interfaces. Output can be limited to a specified number of seconds

or records. The most recent events are listed first. By default, the command displays data for all interfaces, limited to the last 1000 records. Newest events are listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

show queue-monitor length [**INTERFACES**][**FACTOR**] `tx-latency`

Parameters

- **INTERFACES** interface type and number for report. Values include:
  - ***no parameter*** displays information for all interfaces.
  - ***ethernet e-range*** e-range formats include a number, number range, or comma-delimited list of numbers and ranges.
- **FACTOR** limiting parameter for report. Values include:
  - ***no parameter*** displays the last ***1000*** records.
  - **limit number samples** displays the last ***number*** records.
  - **limit number seconds** displays all records generated during the last ***number*** seconds. Value of number ranges from ***1*** to ***1000000***.

Guidelines

This command is available on FM6000, Trident II, Trident 3, and Tomahawk platform switches.

Example

This command displays transmission latency data for the last ***1000*** LANZ events on the switch.

```
switch> show queue-monitor length tx-latency

Report generated at 2017-03-10 15:40:02
Time                          Interface(TC)  Tx-Latency (usecs)
-------------------------------------------------------------------
0:02:29.99222 ago             Et23/3(1)      724.868
0:02:32.33178 ago             Et23/3(1)      724.684
0:02:37.33134 ago             Et23/3(1)      724.684
0:02:42.33135 ago             Et23/3(1)      724.684
0:02:47.33135 ago             Et23/3(1)      724.684
0:02:52.33106 ago             Et23/3(1)      730.985 switch>
```

20.2.4.28 show queue-monitor streaming clients

The **show queue-monitor streaming clients** command displays the number of presently connected clients through LANZ streaming, and also displays their host-names. Ensure that both LANZ and LANZ streaming are enabled in order to use this command.

Command Mode

EXEC

Command Syntax

show queue-monitor streaming clients

Example

This command displays the number of clients connected to LANZ streaming.

```
switch# show queue-monitor streaming clients
Number of clients connected: 3
--------------------------------
172.20.63.161:6565
172.24.17.58:42
172.38.54.142:333
```

20.2.4.29 shutdown (queue-monitor-streaming configuration)

The **shutdown** command disables the streaming of LANZ data to external clients. The **no shutdown** command enables LANZ data streaming. Streaming is disabled by default.

Command Mode

Queue-Monitor-Streaming Configuration

Command Syntax

shutdown

no shutdown

Example

These commands enable the streaming of LANZ data on the switch.

```
switch(config)# queue-monitor streaming
switch(config-qm-streaming)# no shutdown
switch(config-qm-streaming)#
```

20.2.4.30 tcpdump queue-monitor

The **tcpdump queue-monitor** command exports congested traffic to a packet capture device or another tool for analysis, or directly to the switch CPU for inspection.

Command Mode

Global Configuration

Command Syntax

tcpdump queue-monitor [file | filecount | filter | lookup-names | max-file-size | packet-count | size | verbose]

Parameters

- ***file*** output file.
  - **certificate:** certificate file.
    - **file:** standard file.
    - **flash:** flash file.
    - **sslkey:** sslkey file.
    - **usb1:** usb1 file.
- **filecount** specify the number of output files: ***1*** to ***100***.
- **filter** set the filtering expression to select which packets will be dumped.
- **lookup-names** enable reverse DNS lookups.
- **max-file-size** specify the maximum file size by entering *1* to ***100*** million bytes.
- **packet-count** specify ***1*** to ***10000*** packets to capture.
- **size** specify the maximum number of bytes to dump per packet with a size of ***1*** to ***65536*** bytes.

- **verbose** enable verbose mode.

Example

This command inspects traffic on the switch.

```
switch(config)# tcpdump queue-monitor
tcpdump: WARNING: lanz: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lanz, link-type EN10MB (Ethernet), capture size 65535 bytes
...
0 packets captured
0 packets received by filter
0 packets dropped by kernel
switch(config)#
```

20.3 Sampled Flow Tracking

This chapter describes Arista's implementation of sampled flow tracking, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Sampled Flow Tracking Overview
- Configuring Sampled Flow Tracking
- Hardware Flow Tracking with IPFIX Export
- Postcard Telemetry
- Inband Network Telemetry (INT) Support
- Sampled Flow Tracking Configuration Examples
- Sampled Flow Tracking Commands

20.3.1 Sampled Flow Tracking Overview

Network administrators require access to flow information that passes through various network elements to analyze and monitor networks. Sampled flow tracking provides access to IP flow information by sampling traffic flows in ingress direction on the interfaces on which it is configured. The samples are then used to create flow records that are exported to the configured collectors in the Internet Protocol Flow Information Export (IPFIX) format.

Sampled flow tracking terminology:

- **Flow tracker**: It is a collection of interfaces that collect samples and create flow records. The flow tracker has one or more exporters.
- **Exporter**: It sends flow records to one or more collectors.
- **Collector**: It receives flow records from one or more exporters.
- **Data record**: It contains values of the parameters corresponding to a template record.
- **Template record**: It defines the structure and interpretation of fields in a data record. It is an ordered sequence of type and length pairs.
- **Options template record**: It is a type of template record that defines the structure and interpretation of fields in a data record, including how to scope the applicability of the data record.

Data records are created based on the following flow key fields: source IP address, destination IP address, IP protocol, source port, destination port, VRF, and VLAN. These records support IPv4 flow data record and IPv6 flow data record.

Sampled flow tracking supports the following options data records:

- **VRF record**: mapping of VRF ID to VRF name.
- **Interface record**: mapping of interface ID to interface name.
- **Flow key indicator record**: mapping of template ID to flow key indicator.

- **Flow tracker record**: contains information about configured flow tracker.

20.3.1.1 Sampled Flow Tracking Limitations

The limitations of Sampled flow tracking are:

- Sampled flow tracking is active only when sFlow is disabled on the device.
- Sampled flow tracking does not support export of IPFIX messages over ECMP paths.
- Sampled flow tracking route simulation is not supported for ECMP paths.

20.3.2 Configuring Sampled Flow Tracking

These sections describe sampled flow tracking configurations.

Configuring Sampled Flow Tracking

Use the **flow tracker sampled** command to enable sampled flow tracking on a tracker. Each tracker should have a minimum of one exporter configured.

Example

This command enables sampled flow tracking on an interface ***Eth1*** and the flow tracker ***ftr1***.

```
switch(config)# interface Eth1
switch(config-if-Et1)# flow tracker sampled ftr1
```

Use the **sample** command to enable the sample rate for a specific sampled flow tracker. The default sample rate is ***1048576***.

Example

These commands configure a sample rate of ***1024*** for the sampled flow tracker.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# sample 1024
```

20.3.2.1 Configuring the Sampled Flow Tracker

Use the **tracker** command to configure a sampled flow tracker for a device.

Example

This command configures a sampled flow tracker named ***ftr1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
```

Use the **record export on interval** command to configure the interval at which active flow records are exported. The default interval is ***300000*** milliseconds.

Example

These commands configure an active record interval of ***7000*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# record export on interval 7000
```

Use the **record export on inactive timeout** command to configure the interval at which timed-out inactive flow records are exported. The default interval is ***15000*** milliseconds.

These commands configure an inactive record interval of ***4000*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# record export on inactive timeout
 4000
```

20.3.2.2 Configuring Exporter for Sampled Flow Tracker

Use the **`exporter`** command to configure or unconfigure an exporter for a specific tracker.

Example

This command configures exporter ***exp1*** for the specific tracker ***ftr1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
```

Use the **`collector`** command to configure the collector for the specific exporter.

Example

These commands configure a collector for the IPv4 address ***192.0.2.0*** and collector port number ***10***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# collector 192.0.2.0 port 10
```

Use the **`local interface`** command to configure the local source interface for the specific exporter.

Example

These commands configure the local source ***interface Ethernet1*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# local interface Ethernet1
```

Use the **`dscp`** command to configure the DSCP value for the specific exporter. The default DSCP value is ***0***.

Example

These commands configure a DSCP value of ***10*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# dscp 10
```

Use the **`format ipfix version`** command to configure the IPFIX version and maximum packet size for the specific exporter. The default IPFIX version is ***10*** and the default maximum packet size is ***9152***.

Example

These commands configure an IPFIX version of ***10*** and a maximum packet size of ***854*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
```

```
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# format ipfix version 10 max-
packet-size 854
```

Use the **template interval** command to configure the interval at which templates are exported for the specific exporter. The default template interval is ***3600000*** milliseconds.

Example

This command configures the interval of ***3400000*** milliseconds for the exporter ***exp1***.

```
switch(config-ftr-sampled-tr-exp-ftr1-exp1)# template interval 3400000
```

20.3.3 Hardware Flow Tracking with IPFIX Export

Hardware flow tracking uses match criteria to collect data from packets based on defined flow profiles. The data collected is sent to an external node called Collector using IPFIX flow export protocol. The flow tracking engine tracks up to 32K flows going through a given set of ports in a switch with IPFIX-capability. Flow tracking is configured on physical interfaces or LAG interfaces. The matching fields in the packet header are:

- IP source
- IP destination
- IP protocol
- IP protocol's source port
- IP protocol's destination port

The information collected is:

- Byte count (4 bytes)
- Packet count (4 bytes)
- New-learn timestamp
- Flow start timestamp
- Flow end timestamp

Configuring Hardware Flow-tracking

This file extract show a hardware flow-tracking configuration.

```
! Define a loopback interface to act as the local source interface for
! IPFIX export
int Loopback0
 ip address 1.2.3.4/32

@ Enable IP routing for IPFIX packet to be routed to the collector
ip routing

! The flow tracker definition
flow tracking hardware
   tracker myFtr
      record export on inactive timeout 60000
      record export on interval 30000
      !
      exporter myExporter
         local interface Loopback0
         template interval 5000
         collector 172.28.130.153
   no shutdown

! Flow tracked interface/port
int Ethernet48
```

```
flow tracker hardware myFtr
no shutdown
```

This command shows general information about hardware flow tracking.

```
switch# show flow tracking hardware
Flow Tracking Status
  Type: Hardware
  Running: yes
  Tracker: myFtr
    Active interval: 30000ms
    Inactive timeout: 60000ms
    Groups: IPv4, IPv6, VxlanIPv4, VxlanIPv6
    Exporter: myExpoter
      VRF: default
      Local interface: Loopback0 (1.2.3.4)
      Export format: IPFIX version 10, MTU 1500
      DSCP: 0
      Template interval: 5000ms
      Collectors:
        172.28.130.153 port 4739
    Active interfaces:
      Et48
```

This command shows hardware flow tracking **IPFIX** template.

```
switch# show flow tracking hardware ipfix template
Tracker: myFtr
  Data Template, Group: IPv4, Fields: 16, Template ID: 263
    paddingOctets (210), 4 bytes
    aristaBscanExportReason[E] (1036), 2 bytes
    paddingOctets (210), 38 bytes
    destinationTransportPort (11), 2 bytes
    sourceTransportPort (7), 2 bytes
    protocolIdentifier (4), 1 bytes
    destinationIPv4Address (12), 4 bytes
    sourceIPv4Address (8), 4 bytes
    ingressVRFID (234), 2 bytes
    paddingOctets (210), 1 bytes
    aristaBscanTsNewLearn[E] (1040), 6 bytes
    aristaBscanTsFlowStart[E] (1038), 6 bytes
    aristaBscanTsFlowEnd[E] (1039), 6 bytes
    octetDeltaCount (1), 4 bytes
    packetDeltaCount (2), 4 bytes
    paddingOctets (210), 38 bytes

  Data Template, Group: IPv6, Fields: 17, Template ID: 264
    paddingOctets (210), 4 bytes
    aristaBscanExportReason[E] (1036), 2 bytes
    paddingOctets (210), 9 bytes
    sourceIPv6Address (27), 16 bytes
    paddingOctets (210), 5 bytes
    destinationTransportPort (11), 2 bytes
    sourceTransportPort (7), 2 bytes
    protocolIdentifier (4), 1 bytes
    ingressVRFID (234), 2 bytes
    destinationIPv6Address (28), 16 bytes
    paddingOctets (210), 1 bytes
```

```
    aristaBscanTsNewLearn[E] (1040), 6 bytes
    aristaBscanTsFlowStart[E] (1038), 6 bytes
    aristaBscanTsFlowEnd[E] (1039), 6 bytes
    octetDeltaCount (1), 4 bytes
    packetDeltaCount (2), 4 bytes
    paddingOctets (210), 38 bytes

  Data Template, Group: VxlanIPv4, Fields: 16, Template ID: 265
    paddingOctets (210), 4 bytes
    aristaBscanExportReason[E] (1036), 2 bytes
    paddingOctets (210), 38 bytes
    destinationTransportPort (11), 2 bytes
    sourceTransportPort (7), 2 bytes
    protocolIdentifier (4), 1 bytes
    destinationIPv4Address (12), 4 bytes
    sourceIPv4Address (8), 4 bytes
    ingressVRFID (234), 2 bytes
    paddingOctets (210), 1 bytes
    aristaBscanTsNewLearn[E] (1040), 6 bytes
    aristaBscanTsFlowStart[E] (1038), 6 bytes
    aristaBscanTsFlowEnd[E] (1039), 6 bytes
    octetDeltaCount (1), 4 bytes
    packetDeltaCount (2), 4 bytes
    paddingOctets (210), 38 bytes

  Data Template, Group: VxlanIPv6, Fields: 17, Template ID: 266
    paddingOctets (210), 4 bytes
    aristaBscanExportReason[E] (1036), 2 bytes
    paddingOctets (210), 9 bytes
    sourceIPv6Address (27), 16 bytes
    paddingOctets (210), 5 bytes
    destinationTransportPort (11), 2 bytes
    sourceTransportPort (7), 2 bytes
    protocolIdentifier (4), 1 bytes
    ingressVRFID (234), 2 bytes
    destinationIPv6Address (28), 16 bytes
    paddingOctets (210), 1 bytes
    aristaBscanTsNewLearn[E] (1040), 6 bytes
    aristaBscanTsFlowStart[E] (1038), 6 bytes
    aristaBscanTsFlowEnd[E] (1039), 6 bytes
    octetDeltaCount (1), 4 bytes
    packetDeltaCount (2), 4 bytes
    paddingOctets (210), 38 bytes

  Options Template, VRF Mapping, Template ID: 256
    ingressVRFID (234), 4 bytes
    VRFname (236), variable length

  Options Template, Interface Mapping, Template ID: 257
    ingressInterface (10), 4 bytes
    interfaceName (82), variable length

  Options Template, Flow Key, Template ID: 258
    templateId (145), 2 bytes
    flowKeyIndicator (173), 8 bytes

  Options Template, Tracker, Template ID: 259
    observationDomainId (149), 4 bytes
    observationDomainName (300), variable length
    flowActiveTimeout (36), 2 bytes
    flowIdleTimeout (37), 2 bytes
    selectorAlgorithm (304), 2 bytes
    samplingSize (309), 4 bytes
    samplingPopulation (310), 4 bytes
```

```
    flowTrackingType (1001), 2 bytes
```

This command shows hardware flow tracking **IPFIX** template option-table.

```
switch# show flow tracking hardware ipfix options-table
Tracker: myFtr
  Observation domain: myFtr, ID: 1
  Active interval: 5sec
  Inactive timeout: 60sec
  Selector algorithm: random(3)
  Sampling: 1/1
  Flow tracking type: hardware(2)

  VRF Table, Template ID: 256, Scope: ingressVRFID
      VRF ID        VRF Name
  -------------- ---------------
        0            default
        1             vrf1
        2             vrf2
        3        fake-management
        4            vrf500
     16777215

  Interface Table, Template ID: 257, Scope: ingressInterface
     Interface ID    Interface Name
  ------------------ --------------
          0             unknown
          1            Ethernet1
          2            Ethernet2
          3            Ethernet3
          4            Ethernet4
          5            Ethernet5
          6            Ethernet6
          7            Ethernet7
          8            Ethernet8
          9            Ethernet9
          10           Ethernet10
          11           Ethernet11
          12           Ethernet12
          13           Ethernet13
          14           Ethernet14
          15           Ethernet15
          16           Ethernet16
          17           Ethernet17
          18           Ethernet18
          19           Ethernet19
          20           Ethernet20
          21           Ethernet21
          22           Ethernet22
          23           Ethernet23
          24           Ethernet24
          25           Ethernet25
          26           Ethernet26
          27           Ethernet27
          28           Ethernet28
          29           Ethernet29
          30           Ethernet30
          31           Ethernet31
          32           Ethernet32
          33           Ethernet33
```

```
          34            Ethernet34
          35            Ethernet35
          36            Ethernet36
          37            Ethernet37
          38            Ethernet38
          39            Ethernet39
          40            Ethernet40
          41            Ethernet41
          42            Ethernet42
          43            Ethernet43
          44            Ethernet44
          45            Ethernet45
          46            Ethernet46
          47            Ethernet47
          48            Ethernet48
          49            Ethernet49
          50            Ethernet50
          51            Ethernet51
          52            Ethernet52
       53001         Ethernet53/1
       53002         Ethernet53/2
       53003         Ethernet53/3
       53004         Ethernet53/4
       54001         Ethernet54/1
       54002         Ethernet54/2
       54003         Ethernet54/3
       54004         Ethernet54/4
       999001        Management1
      2000002           Vlan2
      2000048           Vlan48
      2000049           Vlan49
      2000100          Vlan100
      7000000           Vxlan1
     1073741823          CPU
     1073741824        discard
     2147483648       multicast

Flow Keys Table, Template ID: 258, Scope: templateId
   Template ID    Flow Key Indicator
----------------- ------------------
       263              0x1f8
       264              0x3e8
       265              0x1f8
       266              0x3e8
```

20.3.4 Postcard Telemetry

The postcard telemetry gathers per flow telemetry information like path and per hop latency. The path, latency and congestion information for flows at different times help in troubleshooting and monitoring flows. Postcard telemetry samples flows at every switch, aggregates them and sends the samples to a collector with path and latency information using GRE encapsulation. For calculating latency information, switches in the network need to be in PTP sync.

The information collected is:

- Length of the truncated samples in bytes.
- 48-bit timestamp.
- SNMP OID values of Ingress and Egress ports.
- 16 bit IP payload checksum, uniquely identify the sample of the same packet from different switches at the collector.

- Sample Rate (Multiplier is 1K).
- Sample data, packet inclusive of L2 header, truncated to 256 bytes.

Configuring Postcard Telemetry for Collector

All switches have same configuration for postcard telemetry to give correct information to collector and it should enabled with PTP.

```
switch(config)# monitor telemetry postcard policy
switch(config-tele-postcard-policy)# no disabled
switch(config-tele-postcard-policy)# ingress collection gre source 10.10.10.10
 destination 172.16.1.1

switch(config)# interface Ethernet1/1
switch(config-if-Et1/1)# telemetry postcard policy profile default
```

The sample rates can be selected:

- 16384 Set sample rate to 1 in 16k packets
- 32768 Set sample rate to 1 in 32k packets
- 65536 Set sample rate to 1 in 64k packets

This example configures sample policy for matching two different flow sets.

Match Rule 1:

- Destination IP prefix ***10.1.1.0/24*** and Source IP prefix ***10.2.2.0/24***
- TCP source port number ***100*** and destination source port number ***200***

Match Rule 2:

- Destination IP prefix ***172.16.2.0/24***
- Source IP prefix ***172.16.3.0/24***

```
switch(config)# monitor telemetry postcard policy
switch(config-tele-postcard-policy)# sample policy mypolicy
switch(config-postcard-sample-policy-mypolicy)# match myrule1 ipv4
switch(config-postcard-sample-policy-match-mypolicy-myrule1-ipv4
)# destination prefix 10.1.1.0/24
switch(config-postcard-sample-policy-match-mypolicy-myrule1-ipv4
)# source prefix 10.2.2.0/24
switch(config-postcard-sample-policy-match-mypolicy-myrule1-ipv4
)# protocol tcp source port 100 destination port 200

switch(config-postcard-sample-policy-mypolicy)# match myrule2 ipv4
switch(config-postcard-sample-policy-match-mypolicy-myrule1-ipv4
)# destination prefix 172.16.2.0/24
switch(config-postcard-sample-policy-match-mypolicy-myrule1-ipv4
)# source prefix 172.16.3.0/24

switch(config-postcard-sample-policy-mypolicy)# profile myprofile
switch(config-postcard-profile-myprofile)# ingress sample policy
 mypolicy

switch(config)# interface Ethernet2/1
switch(config-if-Et1/1)# telemetry postcard policy profile myprofile
```

These actions can be configured for any match rule, sample at specified rate, sampling all packets, or no sampling for the flow. The last option is the default.

```
switch(config)# monitor telemetry postcard policy
switch(config-tele-postcard-policy)# sample policy mypolicy
switch(config-postcard-sample-policy-mypolicy)# match myrule1 ipv4
switch(config-postcard-sample-policy-match-mypolicy-myrule1-ipv4
)# actions
switch(config-postcard-sample-policy-actions-mypolicy-myrule1)# sample
```

Sampling can also be done based on user specified checksum value and mask in TCP/UDP header.

```
switch(config)# monitor telemetry postcard policy
switch(config-tele-postcard-policy)# ingress sample tcp-udp-checksum
 value <val> mask <mask>
```

Show Commands

This shows sample rate and collector IP configuration.

```
switch# show monitor telemetry postcard policy
Enabled: true
Ingress collection sample rate: 16384
Ingress collection type: GRE
Ingress collection source: 10.10.10.10
Ingress collection destination: 172.16.1.1
```

This shows information about postcard telemetry sample policies.

```
switch# show monitor telemetry postcard sample policy
Sample policy default
Total number of rules configured: 1
match ipv4 ipv4-all-default:
        Actions: sample

Sample policy mypolicy
Total number of rules configured: 3
match ipv4 myrule1:
        Source: 10.2.2.0/24
        Destination: 10.1.1.0/24
        Protocol: tcp
                Source port: 100
                Destination port: 200
match ipv4 myrule2:
        Source: 172.16.2.0/24
        Destination: 172.16.2.0/24
match ipv4 ipv4-all-default:
```

This shows different profiles configured and interfaces on which profiles are configured and active.

```
switch# show monitor telemetry postcard policy profiles
Profiles
Name: default
Sample policy: default
```

```
Configured on: Et1/1
Active on: Et1/1

Name: myprofile
Sample policy: mypolicy
Configured on: Et2/1
Active on: Et2/1

switch# show monitor telemetry postcard policy profile myprofile
Profiles
Name: myprofile
Sample policy: mypolicy
Configured on: Et2/1
Active on: Et2/1
```

20.3.4.1 Configuring TCAM Profile for Postcard Telemetry

The postcard telemetry requires the system TCAM profile to have postcard telemetry enabled. This can be achieved by creating a user defined TCAM profile.

The system TCAM profile must have the **`telemetry postcard policy ipv4`** to support postcard telemetry for IPv4 packets. This is applicable for both copied or newly created TCAM profiles.

Creating the User Defined TCAM profile

This adds the postcard telemetry to the default profile.

```
switch(config)# hardware tcam
switch(config-hw-tcam)# profile <profile name> copy default
switch(config-hw-tcam-profile-<profile>)# feature telemetry postcard policy
 ipv4 copy
switch(system-feature-source-profile) #
```

Postcard telemetry is supported for ipv4 bridged and routed packets.

```
switch(config-hw-tcam-profile-<profile>-feature-<feature>)# packet ipv4
 forwarding bridged
switch(config-hw-tcam-profile-<profile>-feature-<feature>)# packet ipv4
 forwarding routed
```

Key size is limited to 160. This is optional for feature copied from the ***system-feature-source-profile***.

```
switch(config-hw-tcam-profile-<profile>-feature-<feature>)# key size limit 160
```

This removes the unused features to ensure that the TCAM DB for postcard telemetry gets allocated.

```
switch(config-hw-tcam-profile-<profile>-feature-<feature>)# exit
switch(config-hw-tcam-profile-<profile>)# no feature mirror ip
switch(config-hw-tcam-profile-<profile>)#
```

Applying the User Defined TCAM Profile

This sets the profile as the system profile under the ***hardware tcam*** mode.

```
switch(config-hw-tcam)# system profile <profilename>
```

When the system TCAM profile is changed, it is expected for some of the agents to restart. This removes the unused features to ensure that the TCAM DB for postcard telemetry gets allocated.

Limitations

- Only IPv4 collector in default VRF is supported.
- Only IPv4 match rules are supported in sample policy.
- Postcard telemetry for VxLAN encapsulated packets is not supported.
- Postcard telemetry for PBR forwarded packets is not supported.
- Postcard telemetry for packets with IP options are not supported.
- Postcard telemetry for multi destination packets is not supported.
- Postcard telemetry for packets dropped or consumed by switch is not supported.
- DCS-7280 and DCS-7500 platforms can support at most 3 postcard policies.
- For INT to be enabled, all the other telemetry features must be in disabled state (e.g.: Sflow, Sampled flow tracking).

20.3.5 Inband Network Telemetry (INT) Support

The Inband Network Telemetry, eXport Data (INT-XD) gathers flow, queue, drop telemetry information like network path, hop latency, queue congestion, drop reasons and more which are used for network monitoring and troubleshooting.

The INT-XD supports:

- Flow telemetry report generates from flow events. Flow events include new flows, change in the attributes of flow like ingress/egress port or latency. Flow reports include information about the path that packets traverse as well as other telemetry metadata such as hop latency and queue occupancy.
- Drop reports provide visibility into the impact of packet drops on user traffic. Drop reports include information about the path that packets traversed as well as other telemetry metadata such as drop reason code and queue id.
- Queue congestion reports are generated from queue-related events, like packets exceeding the queue depth or latency. This provides visibility into the traffic causing and prolonging queue congestion.

Platform Compatibility

The following platforms support the INT-XD feature.

- DCS-7170-64C-F
- DCS-7170-64C-R
- DCS-7170-64C#
- DCS-7170-64C-M#
- DCS-7170-32C-F
- DCS-7170-32C-R
- DCS-7170-32C#
- DCS-7170-32C-M-F
- DCS-7170-32C-M-R
- DCS-7170-32C-M#
- DCS-7170-32CD-F
- DCS-7170-32CD-R
- DCS-7170-32CD#

Configuration

All switches have same configuration for INT-XD and postcard telemetry to give correct information to collector and they are supported only in default profile.

```
switch(config)# platform barefoot profile default
```

This command enters the postcard telemetry context to enable the feature.

```
switch(config)# monitor telemetry postcard int-xd
```

This command enables the postcard telemerty.

```
switch(config-tele-postcard-int-xd)# no disabled
```

This command enables flow report.

```
switch(config-tele-postcard-int-xd)# report flow
```

This command configures flow report refresh interval. By default it is set to 5 seconds.

```
switch(config-tele-postcard-int-xd)# report flow refresh-interval <value>
 seconds
```

This command enables drop report.

```
switch(config-tele-postcard-int-xd)# report drop
```

This command enables queue report.

```
switch(config-tele-postcard-int-xd)# report queue
```

This command configures queue depth threshold. By default it is set to 2 percent.

```
switch(config-tele-postcard-int-xd)# report queue depth 10 percent
```

This command configures switch hop latency threshold. By default it is set to 2048 ns.

```
switch(config-tele-postcard-int-xd)# report queue latency 1024 nanoseconds
```

This command configures queue report suppression limits. Default value is 1000 reports/sec per queue.

```
switch(config-tele-postcard-int-xd)# report queue rate-limit 1000 reports-per-s
econd
```

This command configures unique id of a switch.

```
switch(config-tele-postcard-int-xd)# device id 25
```

This command exits the mode to commit the changes.

```
switch(config-tele-postcard-int-xd)# exit
```

Show Commands

This example shows the mapping between drop reason text and drop reason code.

```
switch(config)# show platform barefoot int drop codes
Code                    Reason
------  ------------------------------------------------------------
     1    Ingress STP blocked
```

```
      2    Ingress invalid VLAN
```

This command shows the current sequence number of the INT-XD reports. Sequence number is incremented every time an INT-XD report is sent. It shows the sequence number in HEX format.

```
switch(config)# show platform barefoot registers seqNumber
```

Limitations

- Specific flow watchlist to filter flows is not supported in this release.
- A 5-tuple of outer header is used for tracking the flows.
- Collector reachability through overlay networks is not supported.
- Packets punted to the CPU are not exported to the collector.
- Collector reachability over non default vrf is not supported.
- Collector reachability via out of band management port is not supported.
- Only unicast packet tail drops are exported to collectors.
- INT reports are sent to a single collector reachable via IPv4.

20.3.6 Sampled Flow Tracking Configuration Examples

This section describes the command configurations required to configure sampled flow tracking.

20.3.6.1 Sampled Flow Tracking Basic Configuration

The following commands enable a basic configuration.

1. Enable configuration mode for the sampled flow tracking on a device.

```
switch(config)# flow tracking sampled
```

2. Configure a sampled flow tracker for a device.

```
switch(config-flow-tracking-sampled)# tracker ftr1
```

3. Configure an exporter for the specific tracker.

```
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
```

4. Configure the collector for the specific exporter.

```
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# collector 172.31.22.131
```

5. Configure the local source ***interface Ethernet1*** for the specific exporter.

```
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# local interface Ethernet1
```

6. Enable sampled flow tracking.

```
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# no shutdown
```

7. Configure the ***interface Ethernet2*** for the specific exporter.

```
switch(config)# interface Ethernet2
```

8. Configure the sampled flow tracker on ***interface Ethernet2***.

```
switch(config)# interface Ethernet2
switch(config-if-Et2)# flow tracker sampled ftr1
```

20.3.7 Sampled Flow Tracking Commands

This section contains descriptions of Sampled flow tracking commands.

Configuration Commands

- collector
- dscp
- exporter
- flow tracking sampled
- format ipfix version
- local interface
- record export on inactive timeout
- record export on interval
- sample
- shutdown (sampled flow tracking)
- template interval
- tracker

Interface Configuration Command

- flow tracker sampled

Privileged EXEC Command

- clear flow tracking sampled counters

Sampled Flow Tracking Display Commands

- show flow tracking sampled
- show flow tracking sampled counters
- show flow tracking sampled flow-table
- show flow tracking sampled ipfix options-table
- show flow tracking sampled ipfix template

20.3.7.1 clear flow tracking sampled counters

The **`clear flow tracking sampled counters`** command clears the flow tracking counters for all trackers, a specified tracker, or a specified tracker and exporter.

Command Mode

Privileged EXEC

Command Syntax

`clear flow tracking sampled counters` [tracker ***tracker_name*** [exporter ***exporter_name***]]

Parameters

- **tracker *tracker_name*** Specifies the flow tracker.
- **exporter *exporter_name*** Specifies the exporter.

Example

This command clears the flow counters for the tracker ***ftr1*** and exporter ***exp1***.

```
switch# clear flow tracking sampled counters tracker ftr1 exporter exp1
switch#
```

20.3.7.2 collector

The **collector** command configures a collector to receive flow records from a specified exporter.

The **no collector** and **default collector** commands remove the configured collector from ***running-config***.

Command Mode

Sampled Flow Tracking Exporter Configuration

Command Syntax

collector {***ipv4_address*** | ***ipv6_address***} [port ***port_number***]

no collector {***ipv4_address*** | ***ipv6_address***} [port ***port_number***]

default collector {***ipv4_address*** | ***ipv6_address***} [port ***port_number***]

Parameters

- ***ipv4_address*** Specifies the IPv4 address of the collector.
- ***ipv6_address*** Specifies the IPv6 address of the collector.
- **port *port_number*** Specifies the port number for the collector. Values range from ***1*** to ***65535***. The default value is ***4739***.

Example

These commands configure a collector for the IPv4 address ***192.0.2.0*** and collector port number ***10***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# collector 192.0.2.0 port 10
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# exit
switch(config-ftr-sampled-tr-ftr1)# exit
switch(config-flow-tracking-sampled)# exit
switch(config)#
```

20.3.7.3 dscp

The **dscp** command configures the Differentiated Services Code Point (DSCP) value for a specific exporter.

The **no dscp** and **default dscp** commands reset the DSCP value to the default of 0.

Command Mode

Sampled Flow Tracking Exporter Configuration

Command Syntax

dscp ***dscp_value***

no dscp ***dscp_value***

default dscp ***dscp_value***

Parameters

dscp_value the DSCP value assigned to the exporter. Value ranges from ***0*** to ***63***. Default value is ***0***.

Example

These commands configure a DSCP value of ***10*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# dscp 10
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# exit
switch(config-ftr-sampled-tr-ftr1)# exit
switch(config-flow-tracking-sampled)# exit
switch(config)#
```

20.3.7.4 exporter

The **exporter** command places the switch in sampled flow tracking exporter configuration mode for the specified exporter and creates the exporter if it does not yet exist.

The **no exporter** and **default exporter** commands remove the specific exporter from ***running-config***.

Command Mode

Sampled Flow Tracking Tracker Configuration

Command Syntax

exporter ***exporter_name***

no exporter ***exporter_name***

default exporter ***exporter_name***

Parameters

exporter_name the name of the exporter.

Example

These commands create exporter ***exp1*** for the flow tracker ***ftr1*** and place the switch in configuration mode for that exporter.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)#exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)#
```

20.3.7.5 flow tracker sampled

The **flow tracker sampled** command configures an interface to be part of a flow tracker. An interface can belong to only one flow tracker.

The **no flow tracker sampled** and **default flow tracker sampled** commands remove the specified interface from the specified tracker.

Command Mode

Interface-Ethernet Configuration

Command Syntax

flow tracker sampled ***tracker_name***

no flow tracker sampled ***tracker_name***

default flow tracker sampled ***tracker_name***

Parameters

tracker_name the name of the flow tracker to which the interface is to be added.

Example

This command configures ***interface ethernet 1*** to participate in the flow tracker ***ftr1***.

```
switch(config)# interface ethernet 1
switch(config-if-Et1)# flow tracker sampled ftr1
switch(config-if-Et1)#
```

20.3.7.6 flow tracking sampled

The **flow tracking sampled** command places the switch in sampled flow tracking configuration mode. Sampled flow tracking configuration mode is a group-change mode; changes made in a group-change mode are saved by exiting the mode.

The **no flow tracking sampled** and **default flow tracking sampled** commands remove all sampled flow tracking configuration from ***running-config***.

Command Mode

Global Configuration

Command Syntax

flow tracking sampled

no flow tracking sampled

default flow tracking sampled

Commands Available in Sampled Flow Tracking Configuration Mode

- **abort** exits mode without saving changes
- **exit** exits mode and saved changes
- **sample** configures sample parameters
- **shutdown (sampled flow tracking)** enables or disables sampled flow tracking
- **tracker** configures a flow tracker

Example

This command places the switch in the ***sampled flow tracking*** configuration mode.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)#
```

20.3.7.7 format ipfix version

The **format ipfix version** command configures the IPFIX version and maximum packet size for a specific exporter.

The **no format ipfix version** and **default format ipfix version** commands remove the previously configured IPFIX version and the maximum packet size value from ***running-config***.

Command Mode

Sampled Flow Tracking Configuration

Command Syntax

format ipfix version ***ipfix_version*** [max-packet-size ***max-packet-size value***]

no format ipfix version ***ipfix_version*** [max-packet-size]

default format ipfix version ***ipfix_version*** [max-packet-size]

Parameters

- ***ipfix_version*** the IPFIX version. Default value is *10*.
- **max-packet-size** ***max-packet-size value*** the IPFIX maximum packet size. Value ranges from ***512*** to ***65472***. Default value is ***9152***.

Example

These commands configure an IPFIX version of ***10*** and a maximum packet size of ***854*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# format ipfix version 10 max-
packet-size 854
```

20.3.7.8 local interface

The **local interface** command configures the local source interface for the specific exporter.

The **no local interface** and **default local interface** commands remove the local interface for the specific exporter from ***running-config***.

Command Mode

Sampled Flow Tracking Configuration

Command Syntax

local interface ***interface***

no local interface

default local interface

Parameters

interface Interface type and numbers. Options include:

- **Ethernet** ***eth_num*** displays the information of the specified Ethernet interface. The value ranges from ***1*** to ***64***.
- **Loopback** ***lb_num*** displays the information of the specified loop back interface. The value ranges from ***0*** to ***2100***.
- **Management** ***m_num*** displays the information of the specified Management interface. The management port number ranges from ***1*** to ***2***.
- **Port-Channel** ***pc_num*** displays the interface or sub-interface information of the specified port channel. The interface and sub-interface values of port channel ranges from ***1***-***1000*** and ***1***-***2000***, ***1***-***4094*** respectively.
- **Tunnel** ***t_num*** displays the information of the specified tunnel. The value ranges from ***0*** to ***255***.
- **Vlan** ***vlan_num*** displays the information of the specified VLAN interface. The value ranges from ***1*** to ***4094***.

Example

These commands configure the local source ***interface Ethernet1*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# local interface Ethernet1
```

20.3.7.9 record export on inactive timeout

The **record export on inactive timeout** command configures the interval at which inactive flow records time out and are exported for a flow tracker.

The **no record export on inactive timeout** and **default record export on inactive timeout** commands remove the timeout interval from ***running-config***.

Command Mode

Sampled Flow Tracking Configuration

Command Syntax

record export on inactive timeout ***timeout_value***

no record export on inactive timeout

default record export on inactive timeout

Parameters

timeout_value the flow record inactive export timeout value in milliseconds. Value ranges from ***3000*** to ***900000***. The default value is ***15000*** milliseconds.

Example

These commands configure an inactive record interval of ***6000*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# record export on inactive timeout
 6000
```

20.3.7.10 record export on interval

The **record export on interval** command configures the interval at which active flow records are exported for a flow tracker.

The **no record export on interval** and **default record export on interval** commands remove the interval from ***running-config***.

Command Mode

Sampled Flow Tracking Configuration

Command Syntax

record export on interval ***interval_value***

no record export on interval

default record export on interval

Parameters

interval_value the flow record export interval in milliseconds. Value ranges from ***5000*** to ***36000000***. The default value is ***300000*** milliseconds.

Example

- These commands configure an active record interval of ***9000*** for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
```

```
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# record export on interval 9000
```

20.3.7.11 sample

The **sample** command enables the sample rate for a specific sampled flow tracker.

The **no sample** and **default sample** commands remove the sample rate configured for a specific sampled flow tracker from ***running-config***.

Command Mode

Sampled Flow Configuration

Command Syntax

sample ***sample_rate***

no sample

default sample

Parameters

sample_rate the sample flow tracking rate to be assigned for a sampled flow tracker. Value ranges from ***1024*** to ***16777216***. Default value is ***1048576***.

Example

These commands configure a sample rate of ***2056*** for the sampled flow tracker.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# sample 2056
```

20.3.7.12 show flow tracking sampled

The **show flow tracking sampled tracker** command displays information about the status of a specific tracker and the status of a specified exporter within that tracker. If no tracker is specified in the command, then all information about all trackers is displayed.

Command Mode

EXEC

Command Syntax

show flow tracking sampled [tracker ***tracker_name*** [exporter ***exporter_name***]]

Parameters

- **tracker *tracker_name*** the specific flow tracker.
- **exporter *exporter_name*** the specific exporter within the tracker.

Example

This command displays the status information of the tracker ***ftr1*** and the exporter ***exp1***.

```
switch# show flow tracking sampled tracker ftr1 exporter exp1
Flow tracking status
 Type: Sampled
 Running: yes
 Sample rate: 1024
 Tracker: ftr1
 Active interval: 30000ms
 Inactive timeout: 120000ms
 Groups: IPv4, IPv6
 Exporter: exp1
```

```
  VRF: default
  Local interface: Management1 (172.30.150.179)
  Export format: IPFIX version 10, MTU 1500
  DSCP: 48
  Template interval: 3600000ms
  Collectors:
  172.31.22.131 port 4739
  Active interfaces:
  Et1
```

20.3.7.13 show flow tracking sampled counters

The **show flow tracking sampled counters** command displays information about the flow tracking counters of a specific tracker and the counters of a specified exporter within that tracker.

Command Mode

EXEC

Command Syntax

show flow tracking sampled counters [tracker ***tracker_name*** [exporter ***exporter_name***]]

Parameters

- **tracker *tracker_name*** the specific flow tracker.
- **exporter *exporter_name*** the specific exporter within the tracker.

Example

This command displays the flow tracking counter information of the tracker ***ftr1*** and the exporter ***exp1***.

```
switch# show flow tracking sampled counters tracker ftr1 exporter exp1
Tracker: ftr1
  1 flows, 22 RX packets
  Flows created: 1, expired: 0
  Group: IPv4
  1 flows, 22 RX packets
  Group: IPv6
  0 flows, 0 RX packets
  Exporter: exp1 (IPFIX)
  Collector: 172.31.24.133 port 4739
  52 messages, last sent 0:00:27 ago
  0 flow records
  2350 options data records, last sent 0:00:27 ago
  6 templates, last sent 0:12:27 ago
  Collector: 172.31.22.131 port 4739
  52 messages, last sent 0:00:27 ago
  0 flow records
  2350 options data records, last sent 0:00:27 ago
  6 templates, last sent 0:12:27 ago
```

20.3.7.14 show flow tracking sampled flow-table

The **show flow tracking sampled flow-table** command displays information about the active flows maintained in the EOS.

Command Mode

EXEC

Command Syntax

show flow tracking sampled flow-table [detail | dst-ip | dst-port | group | interface | protocol | src-ip | src-port | tracker | vlan | vrf]

Parameters

- **detail** displays detailed flow records.
- **dst-ip** displays flow records based on destination IPv4 or IPv6 address.
- **dst-port** displays flow records based on a specified destination port.
- **group** displays flow records based on IPv4 or IPv6 flow groups.
- **interface** displays flow records based on ingress interface.
- **protocol** displays flow records based on the flow IP protocol.
- **src-ip** displays flow records based on source IPv4 or IPv6 address.
- **src-port** displays flow records based on a specified source port.
- **tracker** displays flow records based on flow tracker.
- **vlan** displays flow records based on a specified flow VLAN ID.
- **vrf** displays flow records based on flow VRF.

Examples

- This command displays information about the active flows on the device.

```
switch# show flow tracking sampled flow-table
Tracker: ftr1, Flows: 1
 Group: IPv4, Flows: 1
 VRF  VLAN   Source        Destination   Protocol   Start Time            Pkts   Bytes
 ---- ------ ------------- ------------- ---------- --------------------- ------ ------
 red  42     10.10.1.1:0   10.20.1.2:0   UDP        2019-04-18 15:06:50    7      700
```

- This command displays detailed information about the active flows on the device.

```
switch# show flow tracking sampled flow-table detail
Tracker: ftr1, Flows: 1
  Group: IPv4, Flows: 1
  Flow: UDP 10.10.1.1:0 - 10.20.1.2:0, VRF: red, VLAN: 42
  Start time: 2019-04-18 15:06:50.268734, Last packet time: 2019-04-18
 15:07:03.607900
  Packets: 15, Bytes: 1500, TOS: 0, TCP flags: none
  Source MAC: 001c.73ee.bfe4, Destination MAC: 001c.7374.3b85
  Ingress Interface: 'Ethernet1', Egress VLAN: routed, Egress Interface: CPU
  Next hop: unknown, BGP next hop: unknown (AS unknown), Source AS: unknown
  Source prefix length: 24, Destination prefix length: 32
```

20.3.7.15 show flow tracking sampled ipfix options-table

The **show flow tracking sampled ipfix options-table** command displays information about the sampled IPFIX options table available.

Command Mode

EXEC

Command Syntax

show flow tracking sampled ipfix options-table tracker [flow-key | flow-tracker | interface | vrf]

Parameters

- **tracker** displays the output for a specific flow tracker.
- **flow-key** displays the flow keys options table.
- **flow-tracker** displays the flow tracker options table.
- **interface** displays the interface options table.
- **vrf** displays the VRF options table.

Example

This command displays the sampled IPFIX options table for the tracker ***ftr1***.

```
switch# show flow tracking sampled ipfix options-table
Tracker: ftr1
  Observation domain: ftr1, ID: 1
  Active interval: 30sec
  Inactive timeout: 120sec
  Selector algorithm: random(3)
  Sampling: 1/1024
  Flow tracking type: sampled(1)
  VRF Table, Template ID: 256, Scope: ingressVRFID
  VRF ID VRF Name
  -------------- --------
  0 default
  1 red
  16777215
  Interface Table, Template ID: 257, Scope: ingressInterface
  Interface ID Interface Name
  ------------------ ----------------
  0 unknown
  3013 Ethernet1
  3014 Ethernet2

  1073741823 CPU
  1073741824 discard
  1074029945 Ethernet3/36/1.1
  1074292089 Ethernet3/36/1.2
  2147483648 multicast
  Flow Keys Table, Template ID: 258, Scope: templateId
  Template ID Flow Key Indicator
  ----------------- ------------------
  261 0x7f
  262 0x7f
```

20.3.7.16 show flow tracking sampled ipfix template

The **show flow tracking sampled ipfix template** command displays information about the exported IPFIX data templates and options templates.

Command Mode

EXEC

Command Syntax

show flow tracking sampled ipfix template [data | options | tracker]

Parameters

- **data** displays the data templates.
- **options** displays the flow options template.
- **tracker** displays the flow tracker template.

Example

This command displays the sampled IPFIX options table for the tracker ***ftr1***.

```
switch# show flow tracking sampled ipfix template
Tracker: ftr1
  Data Template, Group: IPv4, Fields: 26, Template ID: 261
    ingressVRFID (234), 4 bytes
    vlanId (58), 2 bytes
```

```
    sourceIPv4Address (8), 4 bytes
    destinationIPv4Address (12), 4 bytes
    protocolIdentifier (4), 1 bytes
    sourceTransportPort (7), 2 bytes
    destinationTransportPort (11), 2 bytes
    sourceMacAddress (56), 6 bytes
    postDestinationMacAddress (57), 6 bytes
    octetDeltaCount (1), 8 bytes
    packetDeltaCount (2), 8 bytes
    flowStartMilliseconds (152), 8 bytes
    flowEndMilliseconds (153), 8 bytes
    flowEndReason (136), 1 bytes
    tcpControlBits (6), 2 bytes
    ingressInterfaceType (368), 4 bytes
    ingressInterface (10), 4 bytes
    postVlanId (59), 2 bytes
    egressInterface (14), 4 bytes
    ipClassOfService (5), 1 bytes
    bgpSourceAsNumber (16), 4 bytes
    bgpDestinationAsNumber (17), 4 bytes
    bgpNextHopIPv4Address (18), 4 bytes
    ipNextHopIPv4Address (15), 4 bytes
    sourceIPv4PrefixLength (9), 1 bytes
    destinationIPv4PrefixLength (13), 1 bytes

<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

20.3.7.17 shutdown (sampled flow tracking)

The **shutdown** command disables sampled flow tracking for the specific exporter.

The **no shutdown** command enables sampled flow tracking for the specific exporter.

Command Mode

Sampled Flow Tracking Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Example

These commands enable sampled flow tracking for the specific exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# local interface Ethernet1
switch(config-ftr-sampled-tr-ftr1-exp-exp1)# no shutdown
```

20.3.7.18 template interval

The **template interval** command configures the interval at which templates are exported for a specific exporter. The default template interval is ***3600000*** milliseconds.

The **no template interval** and **default template interval** commands reset the interval rate to the default.

Command Mode

Sampled Flow Tracking Configuration

Command Syntax

`template interval` ***interval***

`no template interval`

`default template interval`

Parameters

interval the interval rate in milliseconds. The value ranges between ***5000*** and ***3600000*** milliseconds. The default rate is ***3600000*** milliseconds.

Example

This command configures the interval of ***3400000*** milliseconds for the exporter ***exp1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)# tracker ftr1
switch(config-ftr-sampled-tr-ftr1)# exporter exp1
switch(config-ftr-sampled-tr-exp-ftr1-exp1)# template interval 3400000
```

20.3.7.19 tracker

The **`tracker`** command configures a sampled flow tracker for a device.

The **`no tracker`** and **`default tracker`** commands remove the sampled flow tracker from the ***running config***.

Command Mode

Sampled Flow Configuration

Command Syntax

`tracker` ***tracker_name***

`no tracker` ***tracker_name***

`default tracker` ***tracker_name***

Parameters

tracker_name the flow tracker name.

Example

These commands configure the sampled flow tracker ***ftr1***.

```
switch(config)# flow tracking sampled
switch(config-flow-tracking-sampled)#tracker ftr1
```

20.4 sFlow

This chapter describes Arista's implementation of sFlow, including configuration instructions and command descriptions. Topics covered by this chapter include:

- sFlow Conceptual Overview
- sFlow Configuration Procedures
- sFlow Subinterfaces
- QinQ L3 Subinterfaces
- sFlow Commands

20.4.1 sFlow Conceptual Overview

20.4.1.1 sFlow Technology

sFlow is a multi-vendor sampling technology that continuously monitors application level traffic flow at wire speed simultaneously on all interfaces. sFlow provides gigabit speed quantitative traffic measurements without impacting network performance.

sFlow has the following network traffic monitoring characteristics:

- sFlow is a statistical sampling technology that is designed to be deployed on all ports within a network to provide end to end visibility.
- sFlow exports packet samples and topology meta data to a centralized collector application
- sFlow is scalable to operate on all switch ports simultaneously.
- sFlow is implemented on all devices, without requiring additional memory or CPU and does not impact dataplane forwarding
- sFlow is an industry standard (***RFC 3176***).

An sFlow configuration consists of:

- sFlow agents, embedded on network equipment, that monitor traffic and generate data.
- sFlow collectors that receive and analyze sFlow data.

Arista switches include an sFlow agent that monitors ingress data through all Ethernet interfaces.

20.4.1.1.1 sFlow Agents

The sFlow agent is a software process that runs as part of the network management software within an Arista switch. It combines interface counters and flow samples into sFlow datagrams that are sent to an sFlow collector. Packets typically include flow samples and state information of the forwarding/routing table entries associated with each sample. Additional data can be gathered for entries originated by BGP.

The sFlow Agent performs minimal processing when packaging data into datagrams. Immediate data forwarding minimizes agent memory and CPU requirements.

20.4.1.1.2 sFlow Collector

An sFlow collector is a server that runs software that analyzes and reports network traffic. Collectors receive flow samples and counter samples respectively as sFlow datagrams from sFlow agents.

Arista switches reference a collectors IP address and UDP port as a configurable setting through a CLI command. Arista switches do not include sFlow collector software.

20.4.1.1.3 sFlow Data

The sFlow Agent uses two forms of sampling: statistical packet-based sampling of switched flows and time-based sampling of network interface statistics.

- **Switched flow sampling**: A sample is taken by either copying the packet's header or extracting feature data from the packet.
- **Interface statistics sampling**: Counter sampling extracts statistics by periodically polling each data source on the device.

sFlow implements flow sampling and counter sampling as part of an integrated system. An sFlow datagram incorporates both sample types.

20.4.1.2 Arista sFlow Implementation

Arista switches provide a single sFlow agent instance that samples ingress traffic from all Ethernet and port channel interfaces. The switch provides two levels of settings for enabling sFlow:

- a global setting that enables packet sampling on the entire switch.
- interface settings that control sampling on individual interfaces when sFlow is globally enabled.

sFlow default settings include:

- **global**: sFlow and BGP sFlow export are globally disabled.
- **Ethernet and port channel interfaces**: sFlow is enabled on all interfaces when it is enabled globally. BGP sFlow export is likewise enabled on all interfaces when it is enabled globally.

The switch performs sFlow polling when sFlow is globally enabled. The CLI provides commands that globally disable sampling while counter polling remains enabled. Sample enabling is not controllable on individual interfaces.

The switch sends sFlow datagrams to the collector located at an IP address specified by a global configuration command. If the collector destination is not configured, the switch samples data without transmitting the resulting datagrams.

Although the CLI enforces the configured sampling rate limit, it may drop samples if it cannot handle the number of samples it receives over a specified period. Under normal operation, the maximum packet sample rate is one per 16384 packets. The CLI allows for higher sampling rates by using the **dangerous** keyword.

The switch can also be configured to allow the routing agent to export BGP information to the sFlow agent. When BGP sFlow export is enabled, sFlow will add BGP information to packets whose destination is a BGP route.

The following lists describe sFlows sampling behavior relative to different packet types:

- Packets that are sampled:
  - CPU
  - IP Options and MTU violations
  - Flooded packets
  - Multicast packets
- Packets that are not sampled:
  - LACP frames
  - LLDP frames
  - STP BPDUs
  - IGMP packets
  - PAUSE frames
  - PIM hello packets
  - CRC error frames
  - Packets dropped by ACLs or due to VLAN violations

20.4.1.3 sFlow and Mirroring

The sFlow and Mirroring is supported using the same interface for both a mirroring session and sFlow at the same time. But when sFlow and mirroring is configured on the same interface, the TAP Aggregation is not supported. And when TAP Aggregation mode is enabled, the interface configured as a source for both a mirroring session and sFlow will only mirror packets, and sFlow samples are not produced.

The sFlow and Mirroring are supported on the following switch series:

DCS-7280R, DCS-7280R2, DCS-7280E, DCS-7500R, DCS-7500R2, DCS-7500E, DCS-7050X, DCS-7060X, DCS-7250X, DCS-7260X, and DCS-7300X.

However, the following switches have a limitation:

DCS-7280R, DCS-7280R2, DCS-7280E, DCS-7500R, DCS-7500R2, and DCS-7500E.

When a mirroring session and sFlow is configured on a same interface for the above devices, the following packet types are not sampled though they are sampled with only sFlow.

- STP BPDUs
- LACP frames
- LLDP frames
- OSPF packets
- PIM HELLO packets
- Packets dropped due to VLAN violations

20.4.1.4 Hardware Accelerated sFlow

The hardware-accelerated sFlow is supported on compatible platforms. Without hardware acceleration, all sFlow processing is done in software, so performance is dependent on the capabilities of the host CPU. Aggressive sampling rates also decrease the amount of processing time available for other EOS applications.

With hardware acceleration, all sFlow processing is done in hardware using specialized chips, called accelerators. These accelerators process sampled packets and send out sFlow datagrams similarly as the software agent. Involvement from the CPU is very little and the chips are dedicated to sFlow, performance is higher and the CPU has more availability for other tasks, even with high sampling rates.

The Hardware accelerated sFlow is supported on the following switch series:

DCS-7280R3, DCS-7280SR2A-48YC6, DCS-7280CR2-60, DCS-7280CR2A-60, DCS-7280CR2K-60, DCS-7280CR2-30, DCS-7280CR2A-30, DCS-7280SRAM-48C6, DCS-7280SR2K-48C6, DCS-7500R3, DCS-7500R2A-36CQ-LC, DCS-7500R2AK-36CQ-LC, DCS-7500R2AM-36CQ-LC, DCS-7500R2AK-48YCQ-LC. and DCS-7800R3

However, the following switches have a limitation:

DCS-7280R, DCS-7280R2, DCS-7280E, DCS-7500R, DCS-7500R2, and DCS-7500E.

Before enabling hardware-accelerated sFlow, the following requirements must be fulfilled:

- sFlow must be running globally.
- Routing must be enabled in any VRF. This is required for sFlow datagrams to be routed to the collector(s).

The following command enables or disables hardware-accelerated sFlow in configuration mode.

```
switch(config)# [no|default] sflow hardware acceleration
```

The following command disables hardware-accelerated sFlow on a particular LineCard.

```
switch(config)# no sflow hardware acceleration module Linecard3
```

The following command revertshardware-accelerated sFlow on a particular LineCard.

```
switch(config)# sflow hardware acceleration module Linecard3
```

Example

The following example enables routing on the switch, activates sFlow, and enables hardware acceleration globally, but forces software sFlow on Linecard3.

```
switch(config)# ip routing
switch(config)# sflow run
switch(config)# sflow hardware acceleration
switch(config)# sflow source 10.10.10.1
switch(config)# sflow destination 10.10.10.2
switch(config)# sflow hardware acceleration sample 1024
```

```
switch(config)# no sflow hardware acceleration module Linecard3
```

Example

The following example is for IPv6 configuration. Hardware accelerated sFlow enables either IPv4 or IPv6 collectors. IPv6 configuration takes precedence over IPv4 configuration and IPv4 collectors will be disabled.

```
switch(config)# ipv6 unicast-routing
switch(config)# ip routing
switch(config)# sflow run
switch(config)# sflow hardware acceleration
switch(config)# sflow source-interface Ethernet1/1
switch(config)# sflow destination 10.10.10.2
switch(config)# sflow hardware acceleration sample 1024
switch(config)# no sflow hardware acceleration module Linecard3
```

Example

The command **show sflow hardware status** displays the current status of hardware acceleration for sFlow for fixed system.

```
switch(config)# show sflow hardware status
Status
——
Hardware Acceleration On: No
       - sFlow must be enabled
       - sFlow hardware acceleration must be enabled in the CLI
       - routing must be enabled in any VRF
Sample Rate: None
```

Example

The command **show sflow hardware status** displays the current status of hardware acceleration for sFlow for modular system.

```
switch(config)# show sflow hardware status
Status
——
Hardware Acceleration On: Yes
Sample Rate: 1048576
                        sFlow Mode
Module          Active                  Configured              Has sFlow accelerators
——————-      —————————-            —————————         ————————-
Linecard3       Software                Hardware-accelerated        No
Linecard4       Software                Hardware-accelerated        No
Linecard5       Hardware-accelerated    Hardware-accelerated        Yes
Linecard6       Hardware-accelerated    Hardware-accelerated        No
```

Example

The command **show sflow hardware counters** displays counters that are specific to sFlow acceleration.

```
switch(config)# show sflow hardware counters
_______
SflowAccelFpga7:0
_______
Incoming Packet Count :
Outgoing Sflow Datagram Count : 0
Outgoing Flow Sample Count : 0
Incoming Processed Packet Count : 0
Receive Packet Drop Count : 0
Packet Truncated Count : 0
Incoming Packet Error Count : 0
Outgoing Processed Datagram Count : 0
Sample Pool : 0
_______
SflowAccelFpga7:1
_______
Incoming Packet Count : 0
Outgoing Sflow Datagram Count : 0
Outgoing Flow Sample Count : 0
Incoming Processed Packet Count : 0
Receive Packet Drop Count : 0
Packet Truncated Count : 0
Incoming Packet Error Count : 0
Outgoing Processed Datagram Count : 0
Sample Pool : 0
_______
Total
_______
Incoming Packet Count : 0
Outgoing Sflow Datagram Count : 0
Outgoing Flow Sample Count : 0
Incoming Processed Packet Count : 0
Receive Packet Drop Count : 0
Packet Truncated Count : 0
Incoming Packet Error Count : 0
Outgoing Processed Datagram Count : 0
Sample Pool : 0
```

Example

The command **show sflow hardware accelerators** displays a list of all hardware accelerators currently present in the system.

```
switch(config)# show sflow hardware accelerators
Slice     sFlow Accelerator    Type    PCI Address    Direct
 Connections
_____________________________-
Linecard7 SflowAccelFpga7:0   halo    0000:85:00.0     Jericho7/0
Linecard9 SflowAccelFpga9:0   halo    0000:a6:00.0     Jericho9/0
```

Example

The command **show sflow hardware mapping** displays hardware accelerator performs sFlow processing for each switch chip in the system. Hardware acceleration needs to be enabled and running, otherwise the output of the command will be empty.

```
switch(config)# show sflow hardware mapping
Chip          sFlow Accelerator       Direct Connection
——————————————————
Jericho3/0    SflowAccelFpga7:0       False
Jericho3/1    SflowAccelFpga7:0       False
Jericho4/0    SflowAccelFpga9:0       False
Jericho4/1    SflowAccelFpga9:0       False
Jericho5/0    SflowAccelFpga7:0       False
Jericho5/1    SflowAccelFpga7:0       False
Jericho6/0    SflowAccelFpga9:0       False
Jericho6/1    SflowAccelFpga9:0       False
Jericho7/0    SflowAccelFpga7:0       True
Jericho9/0    SflowAccelFpga9:0       True
```

20.4.2 sFlow Configuration Procedures

Implementing sFlow on an Arista switch consists of configuring the following agent parameters:

1. Collector location address.
2. Agent source address.
3. Polling interval.
4. Sampling rate.

Optionally, sFlow can be configured to include output interface and traffic class information in samples using the **sflow sample** command, and to include BGP information in samples whose destination is a BGP route using the **sflow extension bgp** command.

After configuring the sFlow agent, sampling is initiated by globally enabling sFlow on the switch.

Platform-specific Considerations

When BGP sFlow export is enabled on Arad platform switches (DCS-7280E and DCS-7500E), BGP information can be added to some sFlow packets with ECMP destinations.

DCS-7500E switches use actual hardware egress port information in sFlow packets. All other platforms use software simulation to determine the egress port.

Configuring the Collector Location

The **sflow destination** command specifies the IP address and UDP port of an sFlow collector. The switch supports multiple collectors.

Example

This command configures the switch to send sFlow data to collectors at ***10.42.15.12***, port ***6100*** and ***10.52.12.2*** port ***6343*** (the default sFlow port).

```
switch(config)# sflow destination 10.42.15.12 6100
switch(config)# sflow destination 10.52.12.2
switch(config)#
```

Configuring the Agent Source Address

The **sflow source** command specifies the source address that the switch places in all sFlow datagrams that it sends to the collector. This address is normally set to an IP address configured on the switch.

Example

This command configures ***10.2.9.21*** as the sFlow source address.

```
switch(config)# sflow source 10.2.9.21
switch(config)#
```

The **sflow source-interface** command can be alternatively used to specify the interface from which an IP address is derived that the switch places in all sFlow datagrams that it sends to the collector. This address is normally set to an IP address configured on the switch.

Example

This command configures ***interface vlan 25*** as the sFlow source interface. The switch enters the IP address for ***vlan 25*** in the source field of sFlow datagrams.

```
switch(config)# sflow source-interface vlan 25
switch(config)#
```

running-config cannot simultaneously contain **sflow source** and **sflow source-interface** commands.

Configuring the Polling Interval

The **sflow polling-interval** command specifies the interval for sending counter data to the sFlow collector. The default interval is two seconds.

Example

This command configures the switch to send sFlow data every ***10*** seconds.

```
switch(config)# sflow polling-interval 10
switch(config)#
```

Configuring the Sampling Rate and Sample Contents

The **sflow sample** command sets the packet sampling rate. Packets are sampled at random intervals to avoid inaccurate sampling of periodic events. A rate of ***16384*** corresponds to an average sample of one per ***16384*** packets. The default rate is ***1048576***.

Example

This command configures the sFlow sampling rate as ***65536*** (one per ***65536*** packets).

```
switch(config)# sflow sample 65536
switch(config)#
```

The **sflow sample** command can also optionally configure sample packets to include information about the traffic class of the sample. Traffic class is communicated by rewriting the DSCP field in the sample packet.

By default, samples include information about the output interface. To remove this information, use the **[no] sflow sample output interface** command.

Example

These commands configure sFlow to include traffic class information in samples but to exclude output interface data.

```
switch(config)# no sflow sample output interface
switch(config)# sflow sample rewrite dscp
switch(config)#
```

Enabling BGP sFlow Export

The **sflow extension bgp** command enables BGP sFlow export. When it is enabled, the routing agent will export the BGP routing table and autonomous system path information to the sFlow agent. When sFlow receives a sampled packets whose destination is a BGP route, it will look up the following additional BGP routing information and include it in the sample:

- next hop IP
  - AS numbers
  - AS system path to the destination
  - communities
  - local pref

On Arad platform switches (DCS-7280E and DCS-7500E), BGP sFlow export will also add the above BGP information to sample packets with ECMP destination routes unless they exit the switch via a trunk port or subinterface. When egress port is a trunk port or subinterface, the sample packet will only include AS path information from the first path of the ECMP route and a BGP next hop of ***0***.

On all other switch platforms, ECMP destination routes will include AS path information from the first path, but will identify the BGP next hop as ***0***.



Note: A BGP instance must be configured on the switch for BGP sFlow export to operate. See the Border Gateway Protocol (BGP) chapter for details.

Example

These commands configure a BGP instance in AS 50 and enable BGP sFlow export globally.

```
switch(config)# router bgp 50
switch(config-router-bgp)# exit
switch(config)# sflow extension bgp
switch(config)#
```

Extended Switch and Router Information

By default, extended switch and router information is added to sFlow sample packets.

Extended switch information includes the following:

- source and destination VLANs and priorities

Extended router information includes the following:

- IP version and address of next-hop router
  - source and destination mask lengths

The **no** form of **sflow extension switch** and **no** form of **sflow extension router** commands prevent the addition of extended switch and router information to sFlow sample packets.

Example

These commands prevent extended switch and router information from being added to sFlow sample packets.

```
switch(config)# no sflow extension switch
switch(config)# no sflow extension router
switch(config)#
```

Enabling sFlow

The **sflow run** command globally enables sFlow on the switch. The **sflow enable** command controls sFlow operation on Ethernet and port channel interfaces when sFlow is globally enabled. The **`sflow enable`** command has no effect when sFlow is globally disabled.

Example

These commands enable sFlow on the switch, then disables sFlow on Ethernet interface 10.

```
switch(config)# sflow run
switch(config)# interface ethernet 10
switch(config-if-Et10)# no sflow enable
switch(config)#
```

20.4.3 sFlow Subinterfaces

The **`sflow sample [input | output] subinterface`** command configures ifIndex values for subinterfaces on input and output ports to be included in the flow sample. These values are in place of the default parent port ifIndex value. Enabling this feature changes all sFlow samples generated by the switch from the compact to the expanded format.

Configuring sFlow Subinterface

The following configures subinterfaces on the switch for sampling.

```
switch (config)# sflow run
switch (config-if-Et1)# sflow sample input subinterface
switch (config)# sflow sample output subinterface
```

The following file extract displays the output from a **`show sflow detail`** command.

```
switch# show sflow detail
...
Status
------
...
Sample Switch Extension: Yes
Sample Router Extension: Yes
Sample Tunnel IPv4 Egress Extension: No
Sample Input Subinterface: Yes
Sample Output Subinterface: Yes
Port Channel Output Interface Index: portchannel
Sample Encoding Format: expanded
...
```

Limitations

- The feature is supported only on some hardware platforms.
- Only L3 subinterfaces and QinQ L3 subinterfaces support the sFlow output subinterface.
- Tunneled packets such as GRE, MPLSoGRE, and IPinIP are not supported.

20.4.4 QinQ L3 Subinterfaces

QinQ L3 subinterfaces divide a single Ethernet or port-channel interface into multiple logical L3 interfaces based on a combination of two 802.1q tags (VLAN ID) in the incoming traffic. QinQ L3 subinterfaces are commonly used in the L2/L3 boundary device, but they are also used to isolate traffic with a combination of two 802.1q tags between L3 peers by assigning each subinterface to a different VRF.

QinQ L3 subinterfaces are similar to regular L3 subinterfaces, with the only difference being the number of tags being used to isolate traffic. While L3 subinterfaces use a single 802.1q tag (VLAN ID) in the incoming traffic, QinQ L3 subinterfaces use a combination of two 802.1q tags outer, and inner, in the incoming traffic.

All restrictions that are applicable to L3 subinterfaces are also applicable to QinQ L3 subinterfaces including the following:

- PBR service policy.
- ACL logging.
- Tunneling features including VxLAN, MPLS EVPN, MPLS VPN, and Pseudowire.
- QinQ subinterfaces with Algomatch ACLs.
- QinQ subinterfaces with flex-route configuration with optimization for non-nibble aligned prefix length.
- QoS service policy on QinQ subinterfaces with TCAM profiles that have **`feature qos subintf ip/ipv6`**.
- On a routed port, a single tag L3 subinterface and a QinQ L3 subinterface with the same outer tag in the dot1q encapsulation configuration are not supported simultaneously.
- On a routed port ***Et1***, if ***Et1*** has a dot1q encapsulation of ***100***, and if ***Et1.2*** has a dot1q encapsulation of <***100,200*** outer tag ***100***, inner tag ***200***, then ***Et1.1*** and ***Et1.2*** are not supported simultaneously.

20.4.4.1 Configuring QinQ L3 Subinterfaces

The following commands assign packets ingressing on ***Ethernet interface 1/1*** (routed port) with two dot1q tags (VLAN ID) - outer tag ***100*** and inner tag ***200*** to ***Ethernet subinterface 1/1.1***, making ***Ethernet1/1.1*** a QinQ L3 subinterface.

1. Bring up the parent interface and ensure that it is configured as a routed port.

```
witch(config)# interface Ethernet1/1
switch(config-if-Et1/1)# no switchport
switch(config-if-Et1/1)# no shutdown
```

2. Configure a VLAN on the subinterface. The encapsulation dot1q vlan command is also used for VLAN translation, but in this context it associates a VLAN with the subinterface.

```
switch(config-if-Et1/1)# interface Ethernet1/1.1
switch(config-if-Et1/1.1)# encapsulation dot1q vlan 100 inner 200
switch(config-if-Et1/1.1)#
```

3. Configure IPv4, and IPv6 ACL on QinQ subinterface.

```
switch(config)# interface ethernet1/1.1
switch(config-if-Et1/1.1)# ip access-group acl_1 in
switch(config-if-Et1/1.1)# ipv6 access-group acl_v6_1 in
```

4. Configure the subinterface counters to ingress and egress.

```
switch(config)# hardware counter feature subinterface in
switch(config)# hardware counter feature subinterface out
```

To confirm the status of the subinterfaces you can use the **show interface status sub-interfaces** command as shown in the following example.

```
switch(config)# show interface status sub-interfaces
Port       Name    Status       Vlan     Duplex Speed  Type
 Flags  Encapsulation
Et1.1              connect      routed   full   10G    dot1q-encapsulation
   100,200
Et1.2              connect      routed   full   10G    dot1q-encapsulation
   102 switch>
```

To confirm the status of the **show interface et1.1 counters** command, use the ingress and egress counters as shown in the following example.

```
switch(config)# show interface et1.1 counters
L3 Interface                    InOctets          InPkts
Et1.1                                  0               0
L3 Interface                   OutOctets         OutPkts
Et1.1                                  0               0
```

20.4.5 sFlow Commands

Global Configuration Commands

- sflow destination
- sflow extension bgp
- sflow extension router
- sflow extension switch
- sflow polling-interval
- sflow run
- sflow sample
- [no] sflow sample output interface
- sflow source
- sflow source-interface

Interface Configuration Commands

- sflow enable

Privileged EXEC Command

- clear sflow counters

sFlow Display Commands

- show sflow
- show sflow interfaces

20.4.5.1 clear sflow counters

The **`clear sflow counters`** command resets the global sFlow statistics, which includes the number of samples and sample pool. The hardware trigger count is not reset.

The **show sflow** command displays global sFlow statistics.

Command Mode

Privileged EXEC

Command Syntax

`clear sflow counters`

Example

This command resets the sFlow counters.

```
switch# clear sflow counters
switch#
```

20.4.5.2 [no] sflow sample output interface

By default, sFlow samples include information about the output interface of the sampled packet. The **no sflow sample output interface** command prevents sFlow from including that information.

Command Mode

Global Configuration

Command Syntax

no sflow sample output interface

Example

This command configures sFlow to not include output interface information in samples.

```
switch(config)# no sflow sample output interface
switch(config)#
```

20.4.5.3 sflow destination

The **sflow destination** command specifies an sFlow collector IP address and UDP port. The switch supports sFlow collector addresses through multiple sFlow destination commands in ***running-config***.

The **no sflow destination** and **default sflow destination** commands remove the specified sFlow collector IP address by deleting the corresponding **sflow destination** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

sflow destination ***dest_addr*** [**UDP_PORT**]

no sflow destination ***dest_addr*** [**UDP_PORT**]

default sflow destination ***dest_addr*** [**UDP_PORT**]

Parameters

- ***dest_addr*** sflow collectors IP address.
- **UDP_PORT** sFlow collectors data reception port. Options include:
  - ***no parameter*** port number ***6343*** (default).
  - ***port_num*** port number. Value ranges from ***0*** to ***65535***.

Example

This command configures the switch to send sFlow data to the collector located at ***10.42.15.12***; the collector receives the data through UDP port ***6100***.

```
switch(config)# sflow destination 10.42.15.12 6100
switch(config)#
```

20.4.5.4 sflow enable

The **sflow enable** command enables sFlow on the configuration mode interface when sFlow is globally enabled. By default, sFlow is enabled on all interfaces when sFlow is globally enabled (**sflow run**). The **sflow enable** command is required only when ***running-config*** contains a **no sflow enable** statement for the specified interface.

The **no sflow enable** command disables sFlow on the configuration mode interface. When sFlow is globally disabled, this command persists in ***running-config*** but has no effect on switch operation.

The **default sflow enable** command removes the corresponding **no sflow enable** command from ***running-config*** enabling sFlow capability on the interface.

Command Mode

Interface-Ethernet Configuration

Interface-Port-Channel Configuration

Command Syntax

sflow enable

no sflow enable

default sflow enable

Examples

- These commands enable sFlow on the switch and disable sFlow on ***interface ethernet 12***.

```
switch(config)# sflow run
switch(config)# interface ethernet 12
switch(config-if-Et12)# no sflow enable
switch(config-if-Et12)#
```

- This command removes the **no sflow enable** command for ***interface ethernet 12*** from ***running-config***, enabling sFlow on the interface whenever sFlow is globally enabled.

```
switch(config-if-Et12)# sflow enable
switch(config-if-Et12)#
```

20.4.5.5 sflow extension bgp

The **sflow extension bgp** command enables BGP export to sFlow. When enabled, this feature the routing agent will export the BGP routing table and autonomous system path information to the sFlow agent. When sFlow receives a sampled packets whose destination is a BGP route, it will look up the following additional BGP routing information and include it in the sample:

next hop IP

- AS numbers
- AS system path to the destination
- communities
- local pref

The **no sflow extension bgp** and **default sflow extension bgp** commands disable BGP export to sFlow by deleting the corresponding **sflow extension bgp** command from ***running-config***.

Note: A BGP instance must be configured on the switch for BGP sFlow export to operate. See the Border Gateway Protocol chapter for details.

Command Mode

Global Configuration

Command Syntax

sflow extension bgp

no sflow extension bgp

default sflow extension bgp

Guidelines

BGP sFlow export behaves differently on different switch platforms as follows:

- DCS-7500E switches use actual hardware egress port information in sFlow packets. All other platforms use software simulation to determine the egress port.
- On Arad platform switches (DCS-7280E and DCS-7500E), BGP sFlow export works for sample packets with ECMP destination routes unless they exit the switch via a trunk port or subinterface. When egress port is a trunk port or subinterface, the sample packet will only include AS path information from the first path of the ECMP route and a BGP next hop of ***0***.
- On all other switch platforms, ECMP destination routes will include AS path information from the first path, but will identify the BGP next hop as ***0***.
- DCS-7500E switches use actual hardware egress port information in sFlow packets. All other platforms use software simulation to determine the egress port.

Example

These commands configure a BGP instance in ***AS 50*** and enable BGP sFlow export globally.

```
switch(config)# router bgp 50
switch(config-router-bgp)# exit
switch(config)# sflow extension bgp
switch(config)#
```

20.4.5.6 sflow extension router

By default, the switch provides extended router information in sFlow packets, including the IP version and address of the next-hop router and source and destination mask lengths.

The **no** version of the **`sflow extension router`** command prevents this information from being included in sFlow packets.

The **`sflow extension router`** and **`default sflow extension router`** commands restore the default behavior by deleting the corresponding **`no sflow extension router`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`sflow extension router`

`no sflow extension router`

`default sflow extension router`

Example

This command prevents the switch from including extended router information in sFlow packets.

```
switch(config)# no sflow extension router
switch(config)#
```

20.4.5.7 sflow extension switch

By default, the switch provides extended switch information in sFlow packets, including source and destination VLANs and priorities.

The **no** version of the **`sflow extension switch`** command prevents this information from being included in sFlow packets.

The **sflow extension switch** and **default sflow extension switch** commands restore the default behavior by deleting the corresponding **no sflow extension switch** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

sflow extension switch

no sflow extension switch

default sflow extension switch

Example

This command prevents the switch from including extended switch information in sFlow packets.

```
switch(config)# no sflow extension switch
switch(config)#
```

20.4.5.8 sflow polling-interval

The **sflow polling-interval** command specifies the counters polling interval. The switch uses this interval to schedule a ports counter data transmissions to the sFlow collector.

The default interval is two seconds.

The **no sflow polling-interval** and **default sflow polling-interval** commands revert the polling interval to the default of two seconds by removing the **sflow polling-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

sflow polling-interval ***interval_period***

no sflow polling-interval

default sflow polling-interval

Parameters

interval_period polling interval (seconds). Value ranges from ***0*** to ***3600*** (***60*** minutes). Default is ***2***.

Example

This command configures the switch to send sFlow counter data every ***10*** seconds.

```
switch(config)# sflow polling-interval 10
switch(config)#
```

20.4.5.9 sflow run

The **sflow run** command globally enables sFlow on the switch. The default sFlow global setting is ***disabled***. sFlow cannot be enabled on individual interfaces when it is globally disabled.

The **sflow enable** interface configuration command controls sFlow operation on individual Ethernet and port channel interfaces when sFlow is globally enabled. When sFlow is enabled globally, sFlow is also enabled on all interfaces by default.

The **no sflow run** and **default sflow run** commands globally disable sFlow on the switch.

Command Mode

Global Configuration

Command Syntax

sflow run

no sflow run

default sflow run

Examples

- This command enables sFlow on the switch.

```
switch(config)# sflow run
switch(config)#
```

- This command globally disables sFlow.

```
switch(config)# no sflow run
switch(config)#
```

20.4.5.10 sflow sample

The **sflow sample** command sets the packet sampling rate. Packets are sampled at random intervals to avoid inaccurate sampling of periodic events; the packet sampling rate defines the average number of ingress packets that pass through an interface for every packet that is sampled. A rate of ***16384*** corresponds to an average sample of one per ***16384*** packets. The switch may drop samples if it cannot handle the configured sample rate. Under normal operation, the maximum packet sample rate is one per ***16384*** packets. Higher sampling rates can be specified with the **dangerous** option.

By default, samples include information about the output interface. To remove this information, use the **[no] sflow sample output interface** command.

The **sflow sample** command can also optionally configure sample packets to include information about the traffic class of the sample. Traffic class is communicated by rewriting the DSCP field in the sample packet.

The **no sflow sample** and **default sflow sample** commands reset the packet sampling rate to the default of ***1048576*** and remove output interface and traffic class information from samples by removing the **sflow sample** command from the configuration.

Command Mode

Global Configuration

Command Syntax

sflow sample **TRUNCATE SAMPLE_RATE**[rewrite dscp]

no sflow sample

default sflow sample

Parameters

SAMPLE_RATE size of the packet sample from which one packet is selected. Default sample size is 1048576 packets. Options include:

- ***recommended_rate*** Integer between ***16384*** and ***16777215***.
  - **dangerous** ***any_rate*** permits overriding the recommended range of sampling rates. The ***any_rate*** value range varies by platform:

- **fm6000** ***1*** to ***65535***.
- **trident** ***1*** to ***16777216***.

- **`rewrite dscp`** configures sFlow to rewrite the DSCP field of sample packets to indicate the traffic class of the original packet.
- **TRUNCATE** sFlow sample truncation size between the range of ***128*** to ***512***. By default, the sFlow sample truncate size is set to ***128***. Note, this option is hidden.

Examples

- This command configures the sFlow sampling rate as ***65536*** (one per ***65536*** packets).

```
switch(config)# sflow sample 65536
switch(config)#
```

- This command configures the sFlow sampling rate as ***256*** (one per ***256*** packets).

```
switch(config)# sflow sample dangerous 256
switch(config)#
```

- This command configures sFlow to include traffic class information in samples.

```
switch(config)# sflow sample rewrite dscp
switch(config)#
```

20.4.5.11 sflow source

The **`sflow source`** command specifies the IP address used in the **Agent** address filed of the IPv4 sFlow datagram that the switch sends to the collector. This command cannot be used if ***running-config*** contains an **sflow source-interface** command.

The **`no sflow source`** and **`default sflow source`** commands remove the **`sflow source`** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`sflow source` ***source_addr***

`no sflow source`

`default sflow source`

Parameter

source_addr source IP address (dotted decimal notation).

Example

This command configures ***10.2.9.21*** as the sFlow source address.

```
switch(config)# sflow source 10.2.9.21
switch(config)#
```

20.4.5.12 sflow source-interface

The **`sflow source-interface`** command specifies the source IP address that is set to the IP's of the specified interfaces that the switch sends to the collector. Both, the Agent address in the IPv4 sFlow datagram as well as the source IP address sent to the collector are specified in sFlow packet. This command cannot be used if ***running-config*** contains an **sflow source** command.

The **no sflow source-interface** and **default sflow source-interface** commands remove the **sflow source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

sflow source-interface INT_NAME

no sflow source-interface

default sflow source-interface

Parameters

INT_NAME Interface type and number. Options include:

- **interface ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **interface loopback *l_num*** Loopback interface specified by ***l_num***.
- **interface management *m_num*** Management interface specified by ***m_num***.
- **interface port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **interface vlan *v_num*** VLAN interface specified by ***v_num***.

Example

This command configures the sFlow source address as the IP address assigned to the ***loopback 0*** interface.

```
switch(config)# sflow source-interface loopback 0
switch(config)#
```

20.4.5.13 show sflow

The **show sflow** command displays configured sFlow parameters, operational status, and statistics.

The **show sflow interfaces** command displays the interfaces where sFlow is enabled.

Command Mode

EXEC

Command Syntax

show sflow [detail]

Parameters

detail adds hardware sampling status and number of discarded samples to the information displayed.

Examples

- This command displays the base sFlow information.

```
switch# show sflow
! Displaying counters that may be stale
sFlow Configuration
-------------------
Destinations: None (default)
Source(s):
 0.0.0.0 ( default ) ( VRF: default )
 :: ( default ) ( VRF: default )
Sample Rate: 1048576 ( default )
Polling Interval (sec): 2.0 ( default )
Rewrite DSCP value: No
Status
------
```

```
Running: No
Polling On: No
Sampling On: No
Send Datagrams:
 No ( Sflow not running ) ( VRF: default )
BGP Export:
 No ( VRF: default )
Hardware Sample Rate: 1044480
Statistics
 --More--
! Displaying counters that may be stale
sFlow Configuration
-------------------
Destinations: None (default)
Source(s):
 0.0.0.0 ( default ) ( VRF: default )
  :: ( default ) ( VRF: default )
Sample Rate: 1048576 ( default )
Polling Interval (sec): 2.0 ( default )
 Rewrite DSCP value: No
Status
------
Running: No
Polling On: No
Sampling On:
 No Send Datagrams:
  No ( Sflow not running ) ( VRF: default )
BGP Export:
  No ( VRF: default )
Hardware Sample Rate: 1044480

Statistics
 ----------
Total Packets: 0
Number of Samples: 0
Sample Pool: 0
Hardware Trigger: 0
Number of Datagrams: 0
```

- This command displays the expanded sFlow information.

```
switch# show sflow detail
! Displaying counters that may be stale
sFlow Configuration
-------------------
Destinations: None (default)
Source(s):
  0.0.0.0 ( default ) ( VRF: default )
  :: ( default ) ( VRF: default )
Sample Rate: 1048576 ( default )
Polling Interval (sec): 2.0 ( default )
Rewrite DSCP value: No

Status
------
Running: No
Polling On: No
Sampling On: No
Send Datagrams:
  No ( Sflow not running ) ( VRF: default )
BGP Export:
  No ( VRF: default )
Hardware Sample Rate: 1044480
Hardware Sampling On: No
```

```
Sample Output Interface: Yes
Sample Switch Extension: Yes
Sample Router Extension: Yes

Statistics
----------
Total Packets: 0
Number of Samples: 0
Sample Pool: 0
Hardware Trigger: 0
Number of Datagrams: 0
Number of Samples Discarded: 0
```

20.4.5.14 show sflow interfaces

The **show sflow interfaces** command displays the interfaces where sFlow is enabled.

The **show sflow** command displays configured sFlow parameters, operational status, and statistics.

Command Mode

EXEC

Command Syntax

show sflow interfaces

Examples

- This command displays the show sflow interface message when sFlow is globally disabled.

```
switch# show sflow interfaces
sFlow Interface (s):
--------------------
sFlow is not running
```

- This command displays the show sflow interface message when sFlow is globally enabled and enabled on all interfaces.

```
switch(config)# sflow run
switch(config)# show sflow interfaces
Default sFlow configuration for an interface: Disable
sFlow Interface (s):
--------------------
Ethernet1  running(Counter)
Ethernet2  running(Counter)
Ethernet3  running(Flow,Counter)
Ethernet4  running(Flow,Counter)
Ethernet5  running(Flow,Counter)
```

20.5 SNMP

This chapter describes the Arista switch SNMP agent and contains these sections:

- SNMP Introduction
- SNMP Conceptual Overview
- Configuring SNMP
- SNMP Commands

20.5.1 SNMP Introduction

Arista Networks switches support many standard SNMP MIBs, making it easier to integrate these platforms into existing network management infrastructures.

With only a few configurations, many public domain and commercially available network management tools can quickly manage Arista switches out of the box. Support of SNMP V2 groups and views and V3 security allow network managers to tune switch monitoring to match the administration policy of the IT organization.

20.5.2 SNMP Conceptual Overview

Simple Network Management Protocol (SNMP) is a protocol that provides a standardized framework and a common language to monitor and manage network devices.

20.5.2.1 SNMP Structure

The SNMP framework has three parts:

- **SNMP manager**: The SNMP manager controls and monitors network host activities and is typically part of a Network Management System (NMS).
- **SNMP agent**: The SNMP agent is the managed device component that manages and reports device information to the manager.
- **Management Information Base (MIB)**: The MIB stores network management information.

The agent and MIB reside on the switch. Enabling the SNMP agent requires the definition of the manager-agent relationship. The agent contains MIB variables whose values the manager can request or change. The agent gathers data from the MIB and responds to requests for information. For a list of supported MIBs, refer to the release notes for the specific EOS version.

This chapter discusses enabling the SNMP agent on an Arista switch and controlling notification transmissions from the agent. Information on using SNMP management systems is available in the appropriate documentation for the corresponding NMS application.

20.5.2.2 SNMP Notifications

SNMP notifications are messages, sent by the agent, informing of an event or a network condition. A ***trap*** is an unsolicited notification. An ***inform*** (or inform request) is a trap that includes a request for a confirmation that the message is received. Events that a notification can indicate include improper user authentication, restart, and connection losses.

For a list of supported traps, refer to the release notes for the specific EOS version.

20.5.2.3 SNMP Versions

Arista switches support the following SNMP versions:

- **SNMPv1**: The Simple Network Management Protocol, defined in ***RFC 1157***. Security is based on community strings.
- **SNMPv2c**: Community-string based Administrative Framework for SNMPv2, defined in ***RFC 1901***, ***RFC 1905***, and ***RFC 1906***. Security is based on SNMPv1.
- **SNMPv3**: Version 3, as defined in ***RFC 2273*** to ***RFC 2275***.

20.5.2.4 SNMP Authentication and Encryption Methods

The following are the SNMP stronger Authentication and Encryption methods:

- **Authentication**
  - MD5
  - SHA-1

- SHA-224
- SHA-256
- SHA-384
- SHA-512

- **Encryption**
  - AES
  - DES
  - AES-192
  - AES-256

Note: Use the following minimums when using the stronger SNMPv3 encryption algorithm to avoid any interoperability issues.

- When using AES-192 for encryption/privacy, use a minimum of SHA-224 for authentication.
- When using AES-256 for encryption/privacy, use a minimum of SHA-256 for authentication.

20.5.3 Configuring SNMP

This section describes the steps that configure the switch SNMP agent to communicate with an SNMP manager, including the following:

- Enabling and Disabling SNMP
- Enabling SNMP in a VRF
- Configuring Community Access Control
- Configuring SNMP Parameters
- Configuring the Agent to Send Notifications
- Extending the SNMP Agent Through Runtime Scripts
- SNMP IP Address ACL Support

20.5.3.1 Enabling and Disabling SNMP

SNMP is enabled globally by issuing any **snmp-server community** or **snmp-server user** command. The **no snmp-server** command disables SNMP agent operation by removing all non-default **`snmp-server`** commands from ***running-config***.

20.5.3.2 Enabling SNMP in a VRF

By default, SNMP is enabled only in the **default** VRF. The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled.

To enable or disable SNMP in a VRF, use the **snmp-server vrf** command.

20.5.3.3 Configuring Community Access Control

SNMP community strings serve as passwords that permit an SNMP manager to access the agent on the switch. A Network Management System (NMS) can access the switch only if its community string matches at least one of the switch's community strings.

The **snmp-server community** command configures the community string.

Example

This command adds the community string ***ab_1*** to provide read-only access to the switch agent.

```
switch(config)# snmp-server community ab_1 ro
```

```
switch(config)#
```

Community statements can reference views to limit MIB objects that are available to a manager. A view is a community string object that specifies a subset of MIB objects. The **snmp-server view** command configures the community string.

Examples

- These commands create a view that includes all objects in the ***system*** group except for those in ***system.2***.

```
switch(config)# snmp-server view sys-view system include
switch(config)# snmp-server view sys-view system.2 exclude
switch(config)#
```

- This command adds the community string ***lab_1*** to provide read-only access to the switch agent for the previously defined view.

```
switch(config)# snmp-server community lab_1 view sys-view
switch(config)#
```

20.5.3.4 Configuring SNMP Parameters

This section describes these SNMP parameter configuration tasks:

- Configuring the Engine ID
- Configuring the Group
- Configuring the User
- Configuring the Host
- Enabling Link Trap Generation
- Specifying the Source Interface
- Configuring the Chassis-id String
- Configuring the Contact String
- Configuring the Location String

20.5.3.4.1 Configuring the Engine ID

The **snmp-server engineID remote** command configures the name of a Simple Network Management Protocol (SNMP) engine located on a remote device. Use the **snmp-server engineID local** command for the local engine.

A remote agent's engine ID must be configured before remote users for that agent are configured. User authentication and privacy digests are derived from the engine ID and user passwords. The configuration command fails if the remote engine ID is not configured first.

Note: When the remote engine ID is changed, all user passwords associated with the engine must be reconfigured.

Example

This command configures ***DC945798CAB4*** as the name of the remote SNMP engine located at ***12.23.104.25***, UDP ***port 162***

```
switch(config)# snmp-server engineID remote 10.23.104.25 udp-port DC945798CA
switch(config)#
```

20.5.3.4.2 Configuring the Group

An SNMP group grants specific levels of SNMP access to group users. The **snmp-server group** command configures a new SNMP group.

This command configures ***normal_one*** as an SNMPv3 group (authentication and encryption) that provides access to the ***all-items*** read view.

```
switch(config)# snmp-server group normal_one v3 priv read all-items
switch(config)#
```

20.5.3.4.3 Configuring the User

Members of SNMP groups are called ***users***. The **snmp-server user** command allows a new user to be added an SNMP group and configures that user's parameters. Remote users are configured by specifying the IP address or port number that accesses the user's SNMP agent.

Example

- This command configures the local SNMPv3 user ***tech-1*** as a member of the SNMP group ***tech-sup***.

```
switch(config)# snmp-server user tech-1 tech-sup v3
switch(config)#
```

- This command configures the remote SNMPv3 user ***tech-2*** as a member of the SNMP group ***tech-sup***. The remote user is on the agent located at ***13.1.1.4***.

```
switch(config)# snmp-server user tech-2 tech-sup remote 13.1.1.4 v3
switch(config)#
```

20.5.3.4.4 Configuring the Host

The **snmp-server host** command configures an SNMP host (to which SNMP traps will be sent). The **`snmp-server host`** command sets the community string if it was not previously configured.

Example

This command adds a v2c inform notification recipient at ***12.15.2.3*** using the community string ***comm-1***.

```
switch(config)# snmp-server host 12.15.2.3 informs version 2c comm-1
switch(config)#
```

20.5.3.4.5 Enabling Link Trap Generation

The **snmp trap link-change** command enables SNMP link trap generation on the configuration mode interface. SNMP link trap generation is enabled by default. If SNMP link trap generation was previously disabled, this command removes the corresponding **`no snmp link-status`** statement from the configuration. The **show snmp notification** command displays the SNMP link trap generation information.

Example

This command disables SNMP link trap generation on the ***interface ethernet 5***.

```
switch(config-if-Et5)# no snmp trap link-change
switch(config-if-Et5)#
```

20.5.3.4.6 Specifying the Source Interface

The **snmp-server local-interface** command specifies the interface from where an SNMP trap originates. The **show snmp local-interface** command displays the interface of the IP address for SNMP traps.

Example

This command configures the ***ethernet 1*** interface as the source of SNMP traps and informs.

```
switch(config)# snmp-server local-interface ethernet 1
switch(config)#
```

20.5.3.4.7 Configuring the Chassis-id String

The chassis ID string is typically set to the serial number of the switch. The SNMP manager uses this string to associate all data retrieved from the switch with a unique identifying label. Under normal operating conditions, editing the chassis ID string contents is unnecessary.

The **snmp-server chassis-id** command configures the chassis ID string. The default chassis ID string is the serial number of the switch. The **show snmp** command displays the chassis ID.

Example

This command configures ***xyz-1234*** as the chassis-ID string, then displays the result.

```
switch(config)# snmp-server chassis-id xyz-1234
switch(config)# show snmp
Chassis: xyz-1234  <---chassis ID
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
21 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

20.5.3.4.8 Configuring the Contact String

The SNMP contact string is information text that typically displays the name of a person or organization associated with the SNMP agent.

The **snmp-server contact** command configures the system contact string. The contact string is displayed by the **show snmp** and **show snmp v2-mib contact** commands.

Example

These commands configure ***Bonnie H at 3-1470*** as the contact string.

```
switch(config)# snmp-server contact Bonnie H at 3-1470
switch(config)#
```

20.5.3.4.9 Configuring the Location String

The location string typically provides information about the physical location of the SNMP agent. The **snmp-server location** command configures the system location string. By default, the system location string is not set.

Example

These commands configure ***lab-25*** as the location string.

```
switch(config)# snmp-server location lab_25
switch(config)# show snmp v2-mib location
Location: lab_25
switch(config)#
```

20.5.3.5 Configuring the Agent to Send Notifications

The following tasks are mandatory when setting up the SNMP agent to send notifications:

1. Configure the remote engine ID.
2. Configure the group.
3. Configure the user.
4. Configure the host.
5. Enable link trap generation on the interfaces.

Configuring SNMP Parameters describes each of these tasks.

20.5.3.6 Extending the SNMP Agent Through Runtime Scripts

The switch supports the execution of user supplied scripts to service portions of the OID space.

Scripts run under one of two operational modes:

- **Normal mode** scripts run over an indefinite period to process subsequent objects after the initial request. Maintaining an executing script avoids startup and connection delay each time an object requires processing.
- **One-shot mode** scripts process a single object, then terminate execution; requires the **one-shot** keyword.

Startup and data collection overhead is required for each request. In both modes, the SNMP server is blocked from serving other requests when waiting for script responses.

The **snmp-server extension** command configures the execution of user-supplied scripts to service portions of the OID space. Use the **one-shot** keyword to specify one-shot execution.

Examples

- This command specifies the file ***normal-example.sh***, located in flash as the script file that services the specified OID space in normal mode.

```
switch(config)# snmp-server extension .1.3.6.1.4.1.8072.2 flash:normal-
example.sh
switch(config)#
```

- Contents of the script file:

```
#!/bin/bash
```

```
while read cmd; do
   case $cmd in
      PING)
      printf "PONG\n"      ;;
      get)
      read oid
      printf "$oid\n"
      printf "integer\n"
      printf "42\n"      ;;      *)
      printf "NONE\n"
      ;;    esac
done
```

- Testing the script:

```
switch(config)# show snmp mib get .1.3.6.1.4.1.8072.2
NET-SNMP-EXAMPLES-MIB::netSnmpExamples = INTEGER: 42
switch(config)#
```

- This command specifies the file ***one-shot-example.sh***, located in flash as the script file that services the specified OID space in one-shot mode, executing once and then exiting.

```
switch(config)# snmp-server extension .1.3.6.1.4.1.8072.2 flash:one-shot-
example.sh one-shot
switch(config)#
```

- Contents of the script file:

```
#!/bin/bash
oid="$2"
printf "$oid\n"
printf "integer\n"
printf "42\n"
```

- Testing the script:

```
switch(config)# show snmp mib get .1.3.6.1.4.1.8072.2 NET-SNMP-EXAMPLES-
MIB::netSnmpExamples = INTEGER: 42
```

20.5.3.6.1 Normal Script Behavior

The first time the SNMP server requires a script result, it launches it with no arguments. The server communicates with the script through **stdin**/**stdout**. Before each request, the script is sent the string **PING\n** on **stdin**. The expected response from the script is printing **PONG\n** to **stdout**.

GET and GETNEXT Requests

For **GET** and **GETNEXT** requests, the script is passed two lines on **stdin**, the command (**get** or **getnext**) and the requested OID. The expected response from the script is the printing of three lines to **stdout**: the OID for the result varbind, the **TYPE**, and the **VALUE** itself.

Table 73: Extension Script Type and Encoding lists legal TYPE values and resulting VALUE encodings. If the command does not return an appropriate varbind, it should print **NONE\n** to **stdout** and continue running; this results in an SNMP **`noSuchName`** error or a **`noSuchInstance`** exception.

Table 73: Extension Script Type and Encoding

Type string	SNMP type	Encoding for script
integer	Integer32	integer
unsigned	Unsigned32	integer
gauge	Gauge32	integer
counter	Counter32	integer
counter64	Counter64	integer
timetick	TimeTicks	integer
ipaddress	IpAddress	a.b.c.d
objectid	ObjectID	1.3.6.1.42.99.2468
octet	OctetString	hexadecimal string
opaque	Opaque	hexadecimal string
string	OctetString	ascii string

SET Requests

For **SET** requests, script is passed three lines on stdin: the command (set), and the requested OID, and the **TYPE** and **VALUE**, both on the same line. If the assignment is successful, the expected script response is to print **DONE\n** to **stdout**. Indicated errors by writing one of the error strings described in **Set Request Error Strings** In each case, the command should continue running.

Table 74: Set Request Error Strings

authorization-error	no-access	too-big
bad-value	no-creation	undo-failed
commit-failed	no-such-name	wrong-type
gen-error	not-writable	wrong-length
inconsistent-name	read-only	wrong-encoding
inconsistent-value	resource-unavailable	wrong-value

20.5.3.6.2 One-Shot Script Behavior

The command should exit after it finishes processing a single object.

GET and GETNEXT

For each **GET** or **GETNEXT** request, the script is invoked once for each OID in the space that it serves. It receives two arguments: -g for **GET** or -n for **GETNEXT**, and the requested OID.

The expected script response is the response varbind as three separate lines printed to stdout: the result OID, the type, and the value.

If the command does not return an appropriate varbind, then the script should exit without producing any output. This results in an SNMP **noSuchName** error, or a **noSuchInstance** exception.

Possible reasons that a command would not return an appropriate varbind includes:

- The specified OID didn't correspond to a valid instance for a **GET** request.
- There were no following instances for a **GETNEXT**.

SET

A SET request results in the command being called with the arguments: -s, **OID**, **TYPE** and **VALUE**, where **TYPE** is a listed token. **Normal Script Behavior** indicates the type of the value passed as the third parameter.

When the assignment is successful, the script exits without producing any output. Errors are indicated by writing just the error name (**Normal Script Behavior**); the agent generates the appropriate error response.

20.5.3.7 SNMP IP Address ACL Support

SNMP IP address ACL support provides the ability to add access-lists to limit the source addresses that can be used to query the SNMP server. The access-lists are reachable on the switch through the access SNMP data (***port 161***). The access-lists contain standard **`permit`** and **`deny`** commands.

20.5.3.7.1 Configuration

Use the following command to add SNMP IP address ACL support:

`[no | default] snmp-server [[ ipv4 access-list *IP4_ACL*] | [ ipv6 access-list *IP6_ACL* ]][ vrf *VRF* ]`

When the ***VRF*** is not specified, **`default`** is assumed.

20.5.3.7.2 Show Commands

Use the **`show snmp ipv4 access-list summary`** command to display an abreviated output of an IPv4 access-list.

Example

```
switch# show snmp ipv4 access-list summary
IPv4 ACL Permit169
 Total rules configured: 2
Configured on VRFs: red VRF

IPv4 ACL Permit168
Total rules configured: 2
Configured on VRFs: default VRF
Active on VRFs: default VRF
```

Use the **`show snmp ipv4 access-list detail`** command to display a detailed output of an IPv4 access-list.

Example

```
switch# show snmp ipv4 access-list detail
IP Access List Permit169
10 permit ip 192.169.199.0/24 any [match 7 packets, 0:19:56 ago]
20 deny ip any any [match 13 packets, 0:03:56 ago]
Total rules configured: 2
Configured on VRFs: red VRF
```

```
IP Access List Permit168
10 permit ip 192.168.199.0/24 any [match 7 packets, 0:27:00 ago]
20 deny ip any any [match 13 packets, 0:04:30 ago]
Total rules configured: 2
Configured on VRFs: default VRF
Active on VRFs: default VRF
```

Use the **show snmp ipv4 access-list IPv4ACL** command to display a configured access-list. In this example, the configured access-list is ***Permit169***.

Example

```
switch# show snmp ipv4 access-list Permit169
IP Access List Permit169
10 permit ip 192.169.199.0/24 any [match 7 packets, 0:20:12 ago]
20 deny ip any any [match 13 packets, 0:04:12 ago]
Total rules configured: 2
Configured on VRFs: red VRF
```

Use the **show snmp ipv4 access-list summary** command to display a summary of an active access-list.

Example

```
switch# show snmp ipv4 access-list summary
! Same ACL configured in multiple VRFs. Both VRFs are listed in both
 the configured
! and the active sessions
IPv4 ACL Permit169
Total rules configured: 2
Configured on VRFs: default VRF
         red VRF
Active on VRFs: default VRF
 red VRF
```

Use the **show snmp ip access-list summary** command to display a short output of the active access-lists.

Example

```
switch# show snmp ip access-list summary
IPv4 ACL Permit169
Total rules configured: 2
Configured on VRFs: default VRF
         red VRF
Active on VRFs: default VRF
```

20.5.4 SNMP Commands

Global Configuration Commands

- no snmp-server
- snmp-server chassis-id
- snmp-server community
- snmp-server contact
- snmp-server enable traps
- snmp-server engineID local
- snmp-server engineID remote
- snmp-server extension
- snmp-server group
- snmp-server host
- snmp-server local-interface
- snmp-server location
- snmp-server qosmib counter-interval
- snmp-server user
- snmp-server view
- snmp-server vrf

Interface Configuration Commands

- snmp trap link-change

Display Commands

- show snmp
- show snmp community
- show snmp engineID
- show snmp group
- show snmp local-interface
- show snmp mib
- show snmp notification
- show snmp notification | grep bridge
- show snmp notification host
- show snmp user
- show snmp v2-mib chassis
- show snmp v2-mib contact
- show snmp v2-mib location
- show snmp view

20.5.4.1 no snmp-server

The **`no snmp-server`** and **`default snmp-server`** commands disable Simple Network Management Protocol (SNMP) agent operation by removing all **`snmp-server`** commands from ***running-config***.

SNMP is enabled with any **snmp-server community** or **snmp-server user** command.

Command Mode

Global Configuration

Command Syntax

no snmp-server

default snmp-server

Example

This command disables SNMP agent operation on the switch.

```
switch(config)# no snmp-server
switch(config)#
```

20.5.4.2 show snmp

The **show snmp** command displays SNMP information including the SNMP counter status and the chassis ID string.

Command Mode

EXEC

Command Syntax

show snmp

Example

This command displays SNMP counter status, the chassis ID, the previously configured location string, logging status and destination, and the VRFs in which the SNMP agent is operating.

```
switch> show snmp
Chassis: JFL08320162
Location: 5470ga.dc
2329135 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    38132599 Number of requested variables
    0 Number of altered variables
    563934 Get-request PDUs
    148236 Get-next PDUs
    0 Set-request PDUs
2329437 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    2329135 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to 172.22.22.20.162
SNMP agent configured in VRFs: default
SNMP agent enabled in default VRF switch>
```

20.5.4.3 show snmp community

The **show snmp community** command displays the Simple Network Management Protocol (SNMP) community access strings configured by the **snmp-server community** command.

Command Mode

EXEC

Command Syntax

show snmp community

Example

This command displays the list of community access strings configured on the switch.

```
switch> show snmp community
Community name: public switch>
```

20.5.4.4 show snmp engineID

The **show snmp engineID** command displays the local SNMP engine information configured on the switch.

Command Mode

EXEC

Command Syntax

show snmp engineID

Example

This command displays the ID of the local SNMP engine.

```
switch> show snmp engineid
Local SNMP EngineID: f5717f001c730436d700 switch>
```

20.5.4.5 show snmp group

The **show snmp group** command shows the names of configured SNMP groups along with the security model, and view status of each group.

Command Mode

EXEC

Command Syntax

show snmp group [**GROUP_LIST**]

Parameters

GROUP_LIST the name of the group.

- ***no parameter*** displays information about all groups.
- ***group_name*** the name of the group.

Field Descriptions

- **groupname** name of the SNMP group.
- **security model** security model used by the group: ***v1***, ***v2c***, ***orv3***.
- **readview** string identifying the group's read view. Refer to the **show snmp view** comaand.
- **writeview** string identifying the group's write view.
- **notifyview** string identifying the group's notify view. This command displays the groups configured on the switch.

Example

```
switch> show snmp group
groupname : normal                          security model:v3 priv
readview  : all                             writeview: <no writeview specified>
notifyview: <no notifyview specified>
switch>
```

20.5.4.6 show snmp local-interface

The **show snmp local-interface** command displays the interface whose IP address is the source address for SNMP traps.

Command Mode

EXEC

Command Syntax

show snmp local-interface

Example

This command displays the source interface for the SNMP notifications.

```
switch> show snmp local-interface
SNMP source interface: Ethernet1 switch>
```

20.5.4.7 show snmp mib

The **show snmp mib** command displays values associated with specified MIB object identifiers (OIDs) that are registered on the switch.

Command Mode

EXEC

Command Syntax

show snmp mib OBJECTS

Parameters

OBJECTS object identifiers for which the command returns data. Options include:

- **get *oid_1*** [***oid_2 ... oid_x***] values associated with each listed OID.
- **get-next *oid_1*** [***oid_2 ... oid_x***] values associated with subsequent OIDs relative to listed OIDs.
- **table *oid*** table associated with specified OID.
- **translate *oid*** object name associated with specified OID.
- **walk *oid*** objects below the specified subtree.

Example

- This command uses the get option to retrieve information about the ***sysORID.1 OID***.

```
switch# show snmp mib get sysORID.1
SNMPv2-MIB::sysORID[1] = OID: TCP-MIB::tcpMIB
```

- This command uses the get-next option to retrieve information about the OID that is after ***sysORID.8***.

```
switch# show snmp mib get-next sysORID.8
SNMPv2-MIB::sysORDescr[1] = STRING: The MIB module for managing TCP
implementations
```

20.5.4.8 show snmp notification

The **show snmp notification** command displays the SNMP trap generation information.

Command Mode

EXEC

Command Syntax

show snmp notification

Example

This command displays the SNMP traps configured on the switch.

```
switch> show snmp notification
  Type                         Name                                 Enabled
---------------------------- ------------------------------------ -------------
  entity                       entConfigChange                      Yes (default)
  entity                       entStateOperDisabled                 Yes (default)
  entity                       entStateOperEnabled                  Yes (default)
  lldp                         lldpRemTablesChange                  Yes (default)
  msdpBackwardTransition       msdpBackwardTransition               Yes
  msdpEstablished              msdpEstablished                      Yes
  snmp                         linkDown                             Yes
  snmp                         linkUp                               Yes
  snmpConfigManEvent           aristaConfigManEvent                 Yes (default)
  switchover                   aristaRedundancySwitchOverNotif      Yes
  test                         aristaTestNotification               Yes switch>
```

20.5.4.9 show snmp notification host

The **show snmp notification host** command displays information for Simple Network Management Protocol notification. Details include IP address and port number of the Network Management System, notification type, and SNMP version.

Command Mode

EXEC

Command Syntax

show snmp notification host

Field Descriptions

- **Notification host** IP address of the host.
- **udp-port** port number.
- **type** notification type.
- **user** access type of the user.
- **security model** SNMP version used.
- **traps** details of the notification.

Example

This command displays the hosts configured on the switch.

```
switch> show snmp notification host
Notification host: 172.22.22.20    udp-port: 162   type: trap
user: public                       security model: v2c switch>
```

20.5.4.10 show snmp notification | grep bridge

Use the **show snmp notification | grep bridge** command to display the enabled or disabled status of each trap type.

Command Mode

EXEC

Command Syntax

show snmp notification | grep bridge

Example

```
switch(config)# show snmp notification | grep bridge
bridge     arista-mac-age          Yes
bridge     arista-mac-learn        No
bridge     arista-mac-move         No (aristaMacMove default disabled)
```

20.5.4.11 show snmp user

The **show snmp user** command shows information about Simple Network Management Protocol (SNMP) users. Information that the command displays about each user includes their SNMP version, the engine ID of the host where they reside, and security information

Command Mode

EXEC

Command Syntax

show snmp user [**USER_LIST**]

Parameters

USER_LIST the name of the group.

- ***no parameter*** displays information about all users.
- ***user_name*** specifies name of displayed user.

Example

This command displays information about the users configured on the switch.

```
switch> show snmp user
User name: test
Security model: v3
Engine ID: f5717f001c73010e0900
Authentication protocol: SHA
Privacy protocol: AES-128
Group name: normal switch>
```

20.5.4.12 show snmp v2-mib chassis

The **show snmp v2-mib chassis** command displays the Simple Network Management Protocol (SNMP) server serial number or the chassis ID string configured by the **snmp-server chassis-id** command.

Command Mode

EXEC

Command Syntax

show snmp v2-mib chassis

Example

This command displays the chassis ID string.

```
switch> show snmp v2-mib chassis
Chassis: JFL08320162 switch>
```

20.5.4.13 show snmp v2-mib contact

The **show snmp v2-mib contact** command displays the Simple Network Management Protocol (SNMP) system contact string configured by the **snmp-server contact** command. The command has no effect if a contact string was not previously configured.

Command Mode

EXEC

Command Syntax

show snmp v2-mib contact

Example

This command displays the contact string contents.

```
switch> show snmp v2-mib contact
Contact: John Smith switch>
```

20.5.4.14 show snmp v2-mib location

The **show snmp v2-mib location** command displays the Simple Network Management Protocol (SNMP) system location string. The **snmp-server location** command configures system location details. The command has no effect if a location string was not previously configured.

Command Mode

EXEC

Command Syntax

show snmp v2-mib location

Example

This command displays the location string contents.

```
switch> show snmp v2-mib location
Location: santa clara switch>
```

20.5.4.15 show snmp view

The **show snmp view** command displays the information of a Simple Network Management Protocol configuration and the associated MIB. SNMP views are configured with the **snmp-server view** command.

Command Mode

EXEC

Command Syntax

show snmp view [**VIEW_LIST**]

Parameters

VIEW_LIST the name of the view.

- ***no parameter*** displays information about all views.
- ***view_name*** the name of the view.

Field Descriptions

- **First column** view name.
- **Second column** name of the MIB object or family.

- **Third column** inclusion level of the specified family within the view.

Example

These commands configure an SNMP view, then displays that view.

```
switch(config)# snmp-server view sys-view system include
switch(config)# snmp-server view sys-view system.2 exclude
switch(config)# show snmp view
sys-view system - included
sys-view system.2 - excluded
```

20.5.4.16 snmp trap link-change

The **snmp trap link-change** command enables Simple Network Management Protocol (SNMP) link-status trap generation on the configuration mode interface. The generation of link-status traps is enabled by default. If SNMP link-trap generation was previously disabled, this command removes the corresponding **no snmp link-status** statement from the configuration to re-enable link-trap generation.

The **no snmp trap link-change** command disables SNMP link trap generation on the configuration mode interface.

The **snmp trap link-change** and **default snmp trap link-change** commands restore the default behavior by removing the **no snmp trap link-change** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration Interface-Loopback Configuration Interface-Management Configuration Interface-Port-channel Configuration Interface-VLAN Configuration Interface-VXLAN Configuration

Command Syntax

snmp trap link-change

no snmp trap link-change

default snmp trap link-change

Guidelines

The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the **default** VRF. Enable or disable SNMP in a VRF with the **snmp-server vrf** command.

Example

This command disables SNMP link trap generation on the ***interface ethernet 5***.

```
switch(config-if-Et5)# no snmp trap link-change
switch(config-if-Et5)#
```

20.5.4.17 snmp-server chassis-id

The **snmp-server chassis-id** command configures the chassis ID string. The default chassis ID string is the serial number of the switch. The **show snmp** command displays the chassis ID.

The **no snmp-server chassis-id** and **default snmp-server chassis-id** commands restore the default chassis ID string by removing the **snmp-server chassis-id** command from the configuration.

Command Mode

Global Configuration

Command Syntax

snmp-server chassis-id ***id_text***

no snmp-server chassis-id

default snmp-server chassis-id

Parameters

id_text chassis ID string

Example

These commands configure ***xyz-1234*** as the chassis-id string, then display the result.

```
switch(config)# snmp-server chassis-id xyz-1234
switch(config)# show snmp
Chassis: xyz-1234<---chassis ID
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
21 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

20.5.4.18 snmp-server community

The **snmp-server community** command configures the community string. SNMP community strings serve as passwords that permit an SNMP manager to access the agent on the switch. The Network Management System (NMS) must define a community string that matches at least one of the switch community strings to access the switch.

The **no snmp-server community** and **default snmp-server community** commands remove the community access string from the configuration.

Command Mode

Global Configuration

Command Syntax

snmp-server community ***string_text*** [**MIB_VIEW**][**ACCESS**][**ACL_NAMES**]

no snmp-server community ***string_text***

default snmp-server community ***string_text***

Parameters

- ***string_text*** community access string.
- **MIB_VIEW** community access availability. Options include:

  - ***no parameter*** community string allows access to all objects.
  - **view *view_name*** community string allows access only to objects in the ***view_name*** view.
- **ACCESS** community access availability. Options include:
  - ***no parameter*** read-only access (default setting).
  - **ro** read-only access.
  - **rw** read-write access.
- **ACL_NAMES** community access availability. Options include:
  - ***no parameter*** community string allows access to all objects.
  - ***list_v4*** IPv4 ACL list.
  - **ipv6** ***list_v6*** IPv6 ACL list.
  - **ipv6** ***list_v6 list_v4*** IPv4 and IPv6 ACL list.

Example

This command adds the community string ***lab_1*** to provide read-only access to the switch agent.

```
switch(config)# snmp-server community lab_1 ro
switch(config)#
```

20.5.4.19 snmp-server contact

The **`snmp-server contact`** command configures the system contact string. The contact is displayed by the **show snmp** and **show snmp v2-mib contact** commands.

The **`no snmp-server contact`** and **`default snmp-server contact`** commands remove the **`snmp-server contact`** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

`snmp-server contact` ***contact_string***

`no snmp-server contact`

`default snmp-server contact`

Parameters

contact_string system contact string.

Example

These commands configure *Bonnie H* as the contact string, then display the result.

```
switch(config)# snmp-server contact Bonnie H
switch(config)# show snmp
Chassis: xyz-1234
Contact: Bonnie H.
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
```

```
24 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

20.5.4.20 snmp-server enable traps

The **snmp-server enable traps** command enables Simple Network Management Protocol (SNMP) traps. The same command also enables SNMP inform requests. To specify the recipient for notifications, use the **snmp-server host** command. Sending notifications requires the configuration of at least one host using the **snmp-server host** command.

The **snmp-server enable traps** and **no snmp-server enable traps** commands, without a trap-type parameter, specify the default notification setting for all trap types. These commands, when specifying a trap type, control notification generation for the specified trap type. The **default snmp-server enable traps** command resets notification generation to the default setting for the specified trap type.

Command Mode

Global Configuration

Command Syntax

snmp-server enable traps [***trap_type***]

no snmp-server enable traps [***trap_type***]

default snmp-server enable trap [***trap_type***]

Parameters

trap_type controls the generation of informs or traps for the specified trap type:

- ***no parameter*** controls notifications for traps not covered by specific commands.
- **entity** controls entity modification notifications.
- **lldp** controls LLDP notifications.
- **msdpBackwardTransition** controls msdpBackwardTransition notifications.
- **msdpEstablished** controls msdpEstablished notifications.
- **snmp** controls SNMP-v2 notifications.
- **switchover** controls switchover notifications.
- **snmpConfigManEvent** controls snmpConfigManEvent notifications.
- **test** controls test trap notifications.

Examples

- These commands enables notification generation for all trap types except entity traps.

```
switch(config)# snmp-server enable traps
switch(config)# no snmp-server enable traps entity
switch(config)#
```

- This command enables notification generation for all five entity traps, regardless of the default setting.

```
switch(config)# snmp-server enable traps entity
switch(config)#
```

- This command resets the entity trap notification generation to follow the default setting.

```
switch(config)# default snmp-server enable traps entity
switch(config)#
```

20.5.4.21 snmp-server engineID local

The **snmp-server engineID local** command configures the name for the local Simple Network Management Protocol (SNMP) engine. The default SNMP engineID is generated by the switch and is used when an engineID is not configured with this command. The **show snmp engineID** command displays the default or configured engine ID.

SNMPv3 authenticates users through security digests (MD5 or SHA) that are based on user passwords and the local engine ID. Passwords entered on the CLI are similarly converted, then compared to the user's security digest to authenticate the user.

Note: Changing the local engineID value invalidates SNMPv3 security digests, requiring the reconfiguration of all user passwords.

The **no snmp-server engineID** and **default snmp-server engineID** commands restore the default engineID by removing the **snmp-server engineID** command from the ***running-config***

Command Mode

Global Configuration

Command Syntax

snmp-server engineID local ***engine_hex***

no snmp-server engineID local

default snmp-server engineID

Parameter

engine_hex the switch name for the local SNMP engine (hex string).

The string must consist of at least ten characters with a maximum of ***64*** characters.

Example

This command configures ***DC945798CAB4*** as the name of the local SNMP engine.

```
switch(config)# snmp-server engineID local DC945798CAB4
switch(config)#
```

20.5.4.22 snmp-server engineID remote

The **snmp-server engineID remote** command configures the name of a Simple Network Management Protocol (SNMP) engine located on a remote device. The switch generates a default engineID; use the **show snmp engineID** command to view the configured or default engineID.

An SNMPv3 inform requires a remote engine ID to compute the security digest that authenticates and encrypts data transmitted to remote users. SNMPv3 authenticates users with MD5 or SHA through the engine ID and user passwords. CLI passwords are similarly authenticated.

Note: Changing the engineID value invalidates SNMPv3 security digests, requiring the reconfiguration of all user passwords.

The **no snmp-server engineID remote** and **default snmp-server engineID remote** commands remove the **snmp-server engineID remote** command from the configuration.

Command Mode

Global Configuration

Command Syntax

`snmp-server engineID remote` ***engine_addr*** [**PORT**] ***engine_hex***

`no snmp-server engineID remote` ***engine_addr*** [**PORT**]

`default snmp-server engineID remote` ***engine_addr*** [**PORT**]

Parameters

- ***engine_addr*** location of remote engine (IP address or host name).
- **PORT** udp port location of the remote engine. Options include:
  - ***no parameter*** port number ***161*** (default).
  - **udp-port** ***port_num*** port number. Ranges from *0* to ***65535***.
- ***engine_hex*** the switch's name for the remote SNMP engine (hex string).

  The string must have at least ten characters and can contain a maximum of ***64*** characters.

Example

This command configures ***DC945798CA*** as the engineID of the remote SNMP engine located at ***10.23.10.25***, UDP ***port 162***.

```
switch(config)# snmp-server engineID remote 10.23.10.25 udp-port 162 DC945798CA
switch(config)#
```

20.5.4.23 snmp-server extension

The **`snmp-server extension`** command configures the execution of user supplied scripts to service portions of the OID space.

The **`no snmp-server extension`** and **`default snmp-server extension`** commands deletes the **`snmp-server extension`** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

`snmp-server extension` ***OID_space*** **FILE_PATH** [**DURATION**]

Parameters

- ***OID_space*** OID branch serviced by the script, in numerical format.
- **FILE_PATH** path and name of the script file. Options include:
  - **file:** file is located in the switch file directory.
  - **flash:** file is located in flash memory.
- **DURATION** the execution scope of the script.
  - ***no parameter*** script runs after initial request to process subsequent requests.
  - **one-shot** script processes a single object (runs once), then terminates.

Example

This command specifies the file example.sh, located in flash, as the script file that services the listed OID space.

```
switch(config)# snmp-server extension .1.3.6.1.4.1.8072.2 flash:example.sh
```

20.5.4.24 snmp-server group

The **snmp-server group** command configures a new Simple Network Management Protocol (SNMP) group or modifies an existing group. An SNMP group is a data structure that user statements reference to map SNMP users to SNMP contexts and views, providing a common access policy to the specified users.

An SNMP context is a collection of management information items accessible by an SNMP entity. Each item of may exist in multiple contexts. Each SNMP entity can access multiple contexts. A context is identified by the EngineID of the hosting device and a context name.

The **no snmp-server group** and **default snmp-server group** commands delete the specified group by removing the corresponding **snmp-server group** command from the configuration.

Command Mode

Global Configuration

Command Syntax

snmp-server group ***group_name*** **VERSION** [**CNTX**][**READ**][**WRITE**][**NOTIFY**]

no snmp-server group ***group_name*** **VERSION**

default snmp-server group ***group_name*** **VERSION**

Parameters

- ***group_name*** the name of the group.
- **VERSION** the security model utilized by the group.
  - **v1** SNMPv1. Uses a community string match for authentication.
  - **v2c** SNMPv2c. Uses a community string match for authentication.
  - **v3 no auth** SNMPv3. Uses a username match for authentication.
  - **v3 auth** SNMPv3. HMAC-MD5 or HMAC-SHA authentication.
  - **v3 priv** SNMPv3. HMAC-MD5 or HMAC-SHA authentication. AES or DES encryption.
- **CNTX** associates the SNMP group to an SNMP context.
  - ***no parameter*** command does not associate group with an SNMP context.
  - **context** ***context_name*** associates group with context specified by ***context_name***.
- **READ** specifies read view for SNMP group.
  - ***no parameter*** command does not specify read view.
  - **read** ***read_name*** read view specified by ***read_name*** (string maximum 64 characters).
- **WRITE** specifies write view for SNMP group.
  - ***no parameter*** command does not specify write view.
  - **write** ***write_name*** write view specified by ***write_name*** (string maximum 64 characters).
- **NOTIFY** specifies notify view for SNMP group.
  - ***no parameter*** command does not specify notify view.
  - **notify** ***notify_name*** notify view specified by ***notify_name*** (string maximum 64 characters).

Example

This command configures ***normal_one*** as SNMP version 3 group (authentication and encryption) that provides access to the **all-items read view**.

```
switch(config)# snmp-server group normal_one v3 priv read all-items
switch(config)#
```

20.5.4.25 snmp-server host

The **snmp-server host** command configures an SNMP host (to which SNMP traps will be sent) and sets the community string if it was not previously configured. The host is denoted by host location and community string. The command also specifies the type of SNMP notifications that are sent: a trap is an unsolicited notification; an inform is a trap that includes a request for a confirmation that the message is received

The configuration can contain multiple statements to the same host location with different community strings. For instance, a configuration can simultaneously contain all of the following:

- **snmp-server host host-1 version 2c comm-1**
- **snmp-server host host-1 informs version 2c comm-2**
- **snmp-server host host-1 version 2c comm-3 udp-port 666**
- **snmp-server host host-1 version 3 auth comm-3**

The **no snmp-server host** and **default snmp-server host** commands remove the specified host by deleting the corresponding **snmp-server host** statement from the configuration. When removing a statement, the host (address and port) and community string must be specified.

Command Mode

Global Configuration

Command Syntax

snmp-server host ***host_id*** [**VRF_INST**][**MESSAGE**][**VERSION**] ***comm_str*** [**PORT**]

no snmp-server host ***host_id*** [**VRF_INST**][**MESSAGE**][**VERSION**] ***comm_str*** [**PORT**]

default snmp-server host ***host_id*** [**VRF_INST**][**MESSAGE**][**VERSION**] ***comm_str*** [**PORT**]

Parameters

- ***host_id*** hostname or IP address of the SNMP host.
- **VRF_INST** specifies the VRF instance being modified.
  - ***no parameter*** changes are made to the default VRF.
  - **vrf *vrf_name*** changes are made to the specified user-defined VRF.
- **MESSAGE** message type that is sent to the host.
  - ***no parameter*** sends SNMP traps to host (default).
  - **informs** sends SNMP informs to host.
  - **traps** sends SNMP traps to host.
- **VERSION** SNMP version. Options include:
  - ***no parameter*** SNMPv2c (default).
  - **version 1** SNMPv1; option not available with informs.
  - **version 2c** SNMPv2c.
  - **version 3 noauth** SNMPv3; enables user-name match authentication.
  - **version 3 auth** SNMPv3; enables MD5 and SHA packet authentication.
  - **version 3 priv** SNMPv3. HMAC-MD5 or HMAC-SHA authentication. AES or DES encryption.
- ***comm_str*** community string to be sent with the notification as a password.

  Arista recommends setting this string separately before issuing the **snmp-server host** command. To set the community string separately, use the **snmp-server community** command.
- **PORT** port number of the host.
  - ***no parameter*** socket number set to ***162*** (default).
  - **udp-port *p-name*** socket number specified by ***p-name***.

Guidelines

The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the **default** VRF. Enable or disable SNMP in a VRF with the **snmp-server vrf** command.

Example

This command adds a **version 2c** inform notification recipient.

```
switch(config)# snmp-server host 10.15.2.3 informs version 2c comm-1
switch(config)#
```

20.5.4.26 snmp-server local-interface

The **snmp-server local-interface** command specifies the interface where SNMP originates informs and traps.

The **no snmp-server local-interface** and **default snmp-server local-interface** commands remove the inform or trap source assignment by removing the **snmp-server local-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

snmp-server local-interface INTERFACE

no snmp-server local-interface

default snmp-server local-interface

Parameters

INTERFACE Interface type and number. Values include:

- **ethernet *e_num*** Ethernet interface specified by ***e_num***.
- **loopback *l_num*** Loopback interface specified by ***l_num***.
- **management *m_num*** Management interface specified by ***m_num***.
- **port-channel *p_num*** Port-Channel Interface specified by ***p_num***.
- **vlan *v_num*** VLAN interface specified by ***v_num***.
- **vrf *vrf_name*** The VRF in which SNMP is enabled. The keyword **default** specifies the default VRF.

Example

This command configures ***interface ethernet 1*** as the source of SNMP traps and informs.

```
switch(config)# snmp-server local-interface ethernet 1
switch(config)#
```

20.5.4.27 snmp-server location

The **snmp-server location** command configures the system location string. By default, no system location string is set.

The **no snmp-server location** and **default snmp-server location** commands delete the location string by removing the **snmp-server location** command from the configuration.

Command Mode

Global Configuration

Command Syntax

snmp-server location ***node_locate***

no snmp-server location

default snmp-server location

Parameters

node_locate system location information (string).

Example

These commands configure ***lab-east*** as the location string.

```
switch(config)# snmp-server location lab_east
switch(config)#
```

20.5.4.28 snmp-server qosmib counter-interval

The **snmp-server qosmib counter-interval** command configures the interval (in seconds) after which the QoS counters are updated periodically. By default the counter updates are disabled.

Command Mode

Global Configuration

Command Syntax

snmp-server qosmib counter-interval ***timer_interval***

no snmp-server qosmib counter-interval

default snmp-server qosmib counter-interval

Parameter

timer_interval Update interval for refreshing QoS counters (in seconds) between (***10***-***600***).

Example

The following command configures a interval of ***50*** seconds after which the QoS counters are updated periodically.

```
switch(config)# snmp-server qosmib counter-interval 50
```

20.5.4.29 snmp-server user

The **snmp-server user** command adds a user to a Simple Network Management Protocol (SNMP) group or modifies an existing user's parameters.

To configure a user, the IP address or port number of the device where the user's remote SNMP agent resides must be specified. A user's authentication come from the engine ID and the user's password. Remote user configuration commands fail if the remote engine ID is not configured first.

The **no snmp-server user** and **default snmp-server user** commands remove the user from an SNMP group by removing the user command from ***running-config***.

Note: Use the following minimums when using the stronger SNMPv3 encryption algorithm to avoid any interoperability issues.

- When using AES-192 for encryption/privacy, use a minimum of SHA-224 for authentication.
- When using AES-256 for encryption/privacy, use a minimum of SHA-256 for authentication.

Command Mode

Global Configuration

Command Syntax

snmp-server user ***user_name group_name*** [**AGENT**] **VERSION** [**ENGINE**][**SECURITY**]

no snmp-server user ***user_name group_name*** [**AGENT**] **VERSION**

default snmp-server user ***user_name group_name*** [**AGENT**] **VERSION**

Parameters

- ***user_name*** name of user.
- ***group_name*** name of group to which user is being added.
- **AGENT** Options include:
  - ***no parameter*** local SNMP agent.
  - **remote** ***addr*** [**udp-port** ***p_num***] remote SNMP agent location.
- ***addr*** denotes the IP address; ***p_num*** denotes the udp port socket (default port is ***162***).
- **VERSION** SNMP version; options include:
  - **v1** SNMPv1.
  - **v2c** SNMPv2c.
  - **v3** SNMPv3.
- **ENGINE** engine ID used to localize passwords. Available only if **VERSION** is **v3**.
  - ***no parameter*** Passwords localized by SNMP copy specified by ***agent***.
  - **localized** ***engineID*** octet string of engineID.
- **SECURITY** Specifies authentication and encryption levels. Available only if **VERSION** is **v3**. Encryption is available only when authentication is configured.
  - ***no parameter*** no authentication or encryption.
  - **auth** ***a_meth a_pass*** [**priv** ***e_meth e_pass***] authentication parameters.
  - ***a_meth*** authentication method: options are **md5** (HMAC-MD5-96) and **sha** (HMAC-SHA-96).
  - ***a-pass*** authentication string for users receiving packets.
  - ***e-meth*** encryption method: Options are **aes** (AES-128) and **des** (CBC-DES).
  - ***e-pass*** encryption string for the users sending packets.

Example

This command configures the remote SNMP user ***tech-1*** to the ***tech-sup*** SNMP group.

```
switch(config)# snmp-server user tech-1 tech-sup remote 10.1.1.2 v3
```

20.5.4.30 snmp-server view

The **snmp-server view** command defines a view.

An SNMP view defines a subset of objects from an MIB. Every SNMP access group specifies views, each associated with read or write access rights, to allow or limit the group's access to MIB objects.

The **no snmp-server view** command deletes a view entry by removing the corresponding **snmp-server view** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

snmp-server view ***view_name*** **[***family_name***] INCLUSION**

no snmp-server view ***view_name*** [***family_name***]

default snmp-server view ***view_name*** [***family_name***]

Parameters

- ***view_name*** Label for the view record that the command updates. Other commands reference the view with this label.
- ***family_name*** name of the MIB object or family.

  MIB objects and MIB subtrees can be identified by name or by the numbers representing the position of the object or subtree in the MIB hierarchy.
- **INCLUSION** inclusion level of the specified family within the view. Options include:
  - **include** view includes the specified subtree.
  - **exclude** view excludes the specified subtree.

Example

These commands create a view named **sys-view** that includes all objects in the **system subtree** except for those in ***system.2***.

```
switch(config)# snmp-server view sys-view system include
switch(config)# snmp-server view sys-view system.2 exclude
```

20.5.4.31 snmp-server vrf

The **snmp-server vrf** command enables SNMP in the specified VRF. By default, SNMP is enabled only in **default** VRF.

- User-defined VRFs: The **no snmp-server vrf** command disables SNMP in the specified VRF by removing the corresponding **snmp-server vrf** command from the ***running-config***.
- Default VRF: The **no snmp-server vrf** command disables SNMP in the VRF by adding a **no snmp-server vrf default** statement to the ***running-config*****.**

Command Mode

Global Configuration

Command Syntax

snmp-server vrf ***vrf_name***

no snmp-server vrf ***vrf_name***

default snmp-server vrf ***vrf_name***

Parameters

vrf_name The VRF in which SNMP is enabled. The keyword **default** specifies the default VRF.

Guidelines

The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the **default** VRF. Enable or disable SNMP in a VRF with the **snmp-server vrf** command.

Example

These commands disable SNMP in the default VRF, then enable it in the user-defined VRFs named ***magenta*** and ***columbia***.

```
switch(config)# no snmp-server vrf default
switch(config)# snmp-server vrf magenta
switch(config)# snmp-server vrf columbia
switch(config)#
```

20.6 VM Tracer

This chapter describes VM Tracer configuration and usage and contains these sections:

- VM Tracer Introduction
- VM Tracer Description
- VM Tracer Configuration Procedures
- VM Tracer Commands

20.6.1 VM Tracer Introduction

VM Tracer is a switch feature that determines the network configuration and requirements of connected VMware hypervisors. The switch uses VMware's SOAP XML API to discover VMware host server components, including:

- instantiated VMs with their network configuration (VLANs and distributed/virtual Switches).
- server hardware IPMI data which can be shown to the network manager.

VM Tracer also supports adaptive auto-segmentation, which automatically provisions and prunes VLANs from server-switched ports as VMs are instantiated and moved within the data center.

20.6.2 VM Tracer Description

Cloud operating systems manage large virtualized computing infrastructures, including software and hardware. Cloud operating systems consist of virtual machines and hypervisors:

- A Virtual Machine (VM) is a software implementation of a computer that operates as running on dedicated physical hardware. Multiple VMs share physical machine resources from a single physical device. Each VM is controlled by its operating system.
- A hypervisor, also called a Virtual Machine Manager (VMM), is software that manages multiple operating systems running concurrently on a physical device.

VM Tracer tracks activity of VMs that are controlled by hypervisors connected to the switch’s Ethernet or LAG ports. VM Tracer supports vSphere versions 6.0– 7.0. The vSphere features include Distributed Virtual Switches (DVS) and VM movement among VMware servers (VMotion).

vSphere components include:

- ESX and ESXi: hypervisors that run on VMware host server hardware.
- vCenter: centralized tool that manages multiple servers running VMware hypervisors.
- NSX for vSphere® (NSX-V): network virtualization platform delivering networking and security.

Monitoring VLAN based configurations requires vCenter access. Monitoring VXLAN based configurations requires access to vCenter and NSX-V. The following sections describe topologies that monitor these networks:

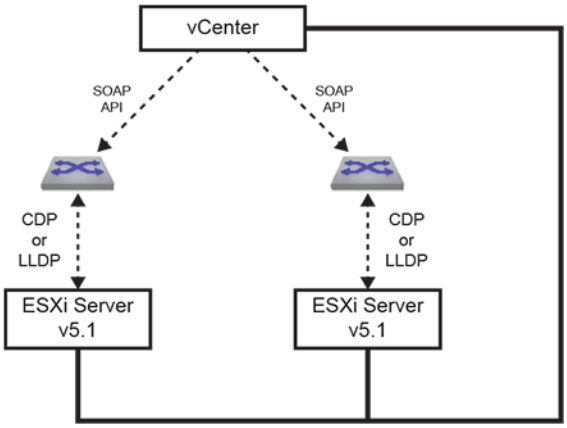
- Monitoring VLAN Based Configurations
- Monitoring VXLAN Based Configurations

20.6.2.1 Monitoring VLAN Based Configurations

vCenter manages ESX hosts and VMs through a central database. VM Tracer identifies interfaces connected to a specified ESX host and sends discovery packets (CDP or LLDP) on interfaces where VM Tracer is enabled. The ESX host updates the vCenter when it receives a discovery packet. VM Tracer reads this data

from the vCenter through a SOAP XML API to associate the ESX host to the connected switch ports. The network topology of this configuration is displayed below.

Figure 20-3: VM Tracer Topology – Monitoring VLAN Based Configurations



VM Tracer connects to a maximum of four vCenters through a SOAP (Simple Object Access Protocol) API to discover VMs in the data centers that the vCenters manage. VM Tracer maintains a list of VMs in the data center and gathers network related information about each VM, including the number of Vnics (virtual network interface card), the MAC address of each Vnic, the switch to which it connects, and the host on which it resides. VM Tracer also identifies the host NICs connected to the switch through the bridge MAC address and the interface port name. VM Tracer then searches for VMs on this host and connected to the vswitch or dvswitch whose uplink is mapped to the connected NIC.

For each connected interface, VM Tracer creates a VM Table that lists its active VMs, sorted by Vnic MAC address. Each VM entry includes its name, Vnic name, VLAN, switch name, datacenter name, and portgroup. An entry is deleted when the corresponding VM is removed, moved to a different host, or its Vnic is no longer part of the vswitch or dvswitch. An entry is added when a VM is created or moved to a host connected to the interface. VM Tracer monitors vCenter for VM management updates. If an interface goes down, all VM entries for that interface are removed from the VM Table.

20.6.2.2 Monitoring VXLAN Based Configurations

Monitoring VXLAN based configurations require access to the NSX for vSphere® (NSX-V), in addition to the configuration described in Monitoring VLAN Based Configurations. Each VM Tracer session can communicate with one NSX-V through a REST interface over XML and gathers VXLAN information by polling it on a 30 second polling cycle. VXLAN data that the switch receives from the NSX-V includes:

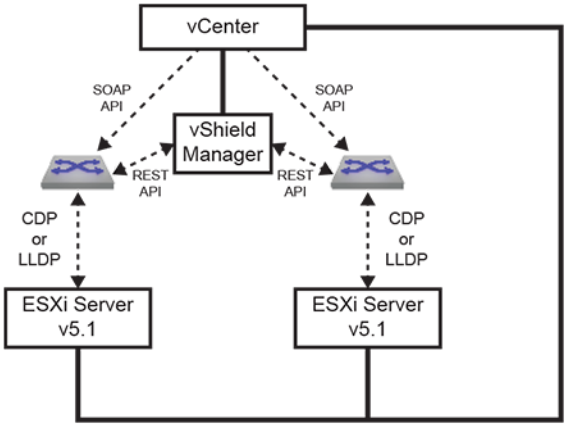
- VNI range.
- VXLAN segment.
- Multicast address range.
- network scope.

The network scope specifies the virtual address space the VXLAN segments span and is defined by the server group (cluster) collections within the segments, which in turn contain a collection of distributed virtual switches (DVS) from ESX hosts within the clusters.

VM Tracer uses this information to build a network model. Communications with NSX-V requires a single polling thread that detects network connectivity and constantly updates the local data model.

The network topology of this configuration is displayed below.

Figure 20-4: VM Tracer Topology – Monitoring VXLAN Based Configurations



20.6.3 VM Tracer Configuration Procedures

The following sections describe the session configuration process, configuring the NSX-V connection for VXLAN based configurations, and the procedure for enabling VM Tracer on individual interfaces. The switch defines the ***vmtracer*** configuration mode and VMtracer mode:

- ***vmtracer*** configuration mode is a command mode for configuring VM Tracer monitoring sessions.
- VMtracer mode is defines an interface state where discovery packets are sent to attached vSwitches.

20.6.3.1 Configuring vCenter Monitoring Sessions

A VM Tracer session connects the switch to a vCenter server for downloading data about VMs and vSwitches managed by ESX hosts connected to the switch's ports. The switch supports four VM Tracer sessions.

The switch is placed in the ***vmtracer*** configuration mode to edit session parameters, including the vCenter location and dynamic VLAN usage. Changes take effect by exiting vmtracer mode.

The **`vmtracer session`** command places the switch in the ***vmtracer*** configuration mode for a specified session. The command either creates a new session or loads an existing session for editing.

Example

This command enters the ***vmtracer*** configuration mode for the ***system_1*** session.

```
switch(config)# vmtracer session system_1
switch(vmtracer-system_1)#
```

In vmtracer configuration mode, the **`url (vmtracer mode)`**, **`username (vmtracer mode)`**, and **`password (vmtracer mode)`** commands specify the location and the account information that authenticates the switch. The URL parameter must reference a fully formed secure URL.

Example

These commands specify the IANA url along with the username and password that allow the switch to access the location.

```
switch(vmtracer-system_1)# url https://example.com/sdk
switch(vmtracer-system_1)# username a-switch_01
switch(vmtracer-system_1)# password abcde
switch(vmtracer-system_1)#
```

Default session settings allow auto-segmentation, or the dynamic allocation and pruning of VLANs when a VM managed by the ESX host connected to the switch is created, deleted, or moved to a different host. The **autovlan disable** command prevents auto-segmentation, regardless of VM activity. The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved. By default, all VLANs are allowed.

Examples

- This command disables auto-segmentation.

```
switch(vmtracer-system_1)# autovlan disable
switch(vmtracer-system_1)#
```

- These commands enable auto-segmentation and limit the list of allowed VLANs to VLAN ***1-2000***.

```
switch(vmtracer-system_1)# no autovlan disable
switch(vmtracer-system_1)# allow-vlan 1-2000
switch(vmtracer-system_1)#
```

The **exit** command returns the switch to the ***global*** configuration mode and enables the VM Tracer session. The ***vmtracer*** configuration mode can be re-entered for this session to edit session parameters.

Example

This command exits vmtracer configuration mode.

```
switch(vmtracer-system_1)# exit
switch(config)#
```

The **no vmtracer session** command disables the session and removes it from ***running-config***.

Example

This command disables and deletes the ***system_1*** VM Tracer session.

```
switch(config)# no vmtracer session system_1
switch(config)#
```

20.6.3.2 Configuring vShield Monitoring Sessions

To monitor VXLAN based VMware configurations, the switch must communicate with a NSX for vSphere® (NSX-V). The ***vmtracer-vxlan*** configuration mode specifies the location and user account data that allows the switch to access a NSX-V within the configuration mode vmtracer session.

The switch is placed in the ***vmtracer*** configuration mode to edit session parameters, including the vCenter location and dynamic VLAN usage. Changes take effect by exiting vmtracer mode.

The **vxlan (vmtracer mode)** command is executed from the ***vmtracer*** mode for a specified session and places the switch in the ***vmtracer-vxlan*** configuration mode for that session. Each VM Tracer session can be associated with one vShield instance.

Example

These commands create the vShield instance for the VMTracer session named ***vnet-1***.

```
switch(config)# vmtracer session vnet-1
switch(config-vmtracer-vnet-1)# vxlan
switch(config-vmtracer-vnet-1-vxlan)#
```

In the ***vmtracer-vxlan*** configuration mode, the **url (vmtracer-vxlan mode)**, **username (vmtracer-vxlan mode)**, and **password (vmtracer-vxlan mode)** commands specify the vShield server's location

and the account information that authenticates the switch to the vShield server. The url parameter must reference a fully formed secure url, such as **https://vcshield.democorp.com/sdk**.

Example

These commands specify the vShield's URL along with the username and password that allow the switch to access the vShield server.

```
switch(config-vmtracer-vnet-1-vxlan)# url https://vshieldserver.company1.org/
sdk
switch(config-vmtracer-vnet-1-vxlan)# username a-shield_01
switch(config-vmtracer-vnet-1-vxlan)# password home
switch(config-vmtracer-vnet-1-vxlan)#
```

20.6.3.3 Enabling VMtracer Mode

VMtracer mode is an interface setting that enables interfaces to send discovery packets to the connected vSwitch. The **vmtracer** command enables VMtracer mode on the configuration mode interface.

Examples

- These commands enable VMtracer mode on the ***interface Ethernet3***.

  ```
  switch(config)# interface Ethernet3
  switch(config-if-Et3)# vmtracer vmware-esx
  switch(config-if-Et3)#
  ```

  The **no vmtracer** command disables vmtracer mode on the configuration mode interface.
- This command disables vmtracer mode on the ***interface ethernet 3***.

  ```
  switch(config-if-Et3)# no vmtracer vmware-esx
  switch(config-if-Et3)#
  ```

20.6.3.4 Displaying VM Tracer Data

20.6.3.4.1 Displaying Session Status

The **show vmtracer session** command displays information about the specified session.

Without the **detail** parameter, the command displays connection parameters and status for the vCenter associated to the specified session.

Example

This command displays connection parameters for the vCenter associated with the system_1 session.

```
switch# show vmtracer session system_1
    vCenter URL https://vmware-vcenter1/sdk
    username arista
    password arista
    Session Status Disconnected
```

With the **detail** parameter, the command displays connection status and data concerning messages the vCenter previously received from ESX hosts connected to the switch.

Example

This command displays connection parameters and message details for the vCenter associated with the ***system_1*** session.

```
switch# show vmtracer session system_1 detail
```

```
    vCenter URL https://vmware-vcenter1/sdk
    username arista
    sessionState Connected
    lastStateChange 19 days, 23:03:59 ago
    lastMsgSent CheckForUpdatesMsg
    timeOfLastMsg 19 days, 23:14:09 ago
    resonseTimeForLastMsg 0.0
    numSuccessfulMsg 43183
    lastSuccessfulMsg CheckForUpdatesMsg
    lastSuccessfulMsgTime 19 days, 23:14:19 ago
    numFailedMsg 1076
    lastFailedMsg CheckForUpdatesMsg
    lastFailedMsgTime 19 days, 23:14:09 ago
    lastErrorCode Error -1 fault: SOAP-ENV:Client [no subcode]
    "End of file or no input: Operation interrupted or timed out after 600s
 send
or     600s receive delay"
    Detail: [no detail]
    CheckForUpdates:
```

20.6.3.4.2 Displaying VM Interfaces

The **show vmtracer interface** command displays the VM interfaces (Vnics) that are active on switch interfaces where vmtracer mode is enabled. For each Vnic, the command displays the name of the attached VM, the adapter name, its VLAN, the VM power state, and the presence status of its MAC address in the switch's MAC table.

This command displays the Vnics connected to all VM Tracer-enabled interfaces.

```
switch# show vmtracer interface

Ethernet8 : example.com
    VM Name VM Adapter VLAN Status
    esx3.aristanetworks.com vmk0 0 Up/Down
    vspheremanagement Network adapter 1 0 Up/Down

Ethernet15 : example.om
    VM Name VM Adapter VLAN Status
    Openview Network adapter 1 123 Up/Down
    VmTracerVm Network adapter 1 123 Down/Down

Ethernet23 : example.com
    VM Name VM Adapter VLAN Status

Ethernet24 : example.com
    VM Name VM Adapter VLAN Status
```

20.6.3.4.3 Displaying VMs

The **show vmtracer vm** command displays VM interfaces (Vnics) accessible to the VM Tracer-enabled interfaces. For each active listed VM, the command displays its name, adapter, and the connected hypervisor.

- This command displays the VMs connected to all VM Tracer-enabled interfaces.

```
switch# show vmtracer vm
    VM Name VM Adapter Interface VLAN
    Openview Network adapter 1 Et15 123
    vspheremanagement Network adapter 1 Et8 0
    VmTracerVm Network adapter 1 Et15 123
    example.com vmk0 Et8 0
```

- This command displays connection data for the VMs connected to all VM Tracer-enabled interfaces.

```
switch# show vmtracer vm detail
VM Name Openview
   intf : Et15
   vnic : Network adapter 1
   mac : 00:0c:29:ae:7e:90
   portgroup : dvPortGroup
   vlan : 123
   switch : vds
   host : example.com
```

20.6.4 VM Tracer Commands

Global Configuration Commands

- vmtracer session

Interface Configuration (Ethernet and Port Channel) Commands

- vmtracer

VMTracer Configuration Commands

- allowed-vlan
- autovlan disable
- password (vmtracer mode)
- source-interface
- url (vmtracer mode)
- username (vmtracer mode)
- vrf (vmtracer mode)
- vxlan (vmtracer mode)

VMTracer-VXLAN Configuration Commands

- password (vmtracer-vxlan mode)
- url (vmtracer-vxlan mode)
- username (vmtracer-vxlan mode)

VM Tracer Display Commands

- show vmtracer all
- show vmtracer interface
- show vmtracer session
- show vmtracer session vcenter
- show vmtracer session vsm
- show vmtracer vm
- show vmtracer vm detail
- show vmtracer vnic counters
- show vmtracer vxlan segment
- show vmtracer vxlan vm

20.6.4.1 allowed-vlan

The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved from the hypervisor connected to the session specified by the vmtracer mode. By default, all VLANs are allowed.

Command Mode

Vmtracer Configuration

Command Syntax

allowed-vlan [**VLAN_LIST**]

no allowed-vlan

default allowed-vlan vlan

Parameters

VLAN_LIST The VLAN list or the edit actions to the current VLAN list. Valid ***v_range*** formats include number, or number range.

- ***v_range*** The list consists of the ***v_range*** VLANs.
- **add *v_range*** The ***v_range*** VLANs are added to the current VLAN list.
- **all** The list consists of all VLANs (***1-4094***).
- **except *v_range*** The list consists of all VLANs except for those specified by ***v_range***.
- **none** The list of VLANs is empty.
- **remove *v_range*** The ***v_range*** VLANs are removed from the current VLAN list.

Related Commands

vmtracer session places the switch in the ***vmtracer*** configuration mode.

Examples

- This command sets the list of allowed VLANs to ***1*** through ***2000***.

```
switch(vmtracer-system_1)# allow-vlan 1-2000
switch(vmtracer-system_1)#
```

- This command adds VLANs to ***2501*** through ***3000***.

```
switch(vmtracer-system_1)# allow-vlan add 2051-3000
switch(vmtracer-system_1)#
```

20.6.4.2 autovlan disable

Default VM Tracer session settings enable auto provisioning, which allows the dynamic assignment and pruning of VLANs when a VM attached to the ESX connected to the switch is created, deleted, or moved to a different ESX host. The autovlan setting controls auto provisioning.

The **autovlan disable** command disables auto provisioning, which prevents the creation or deletion of VLANs regardless of VM activity. The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved. By default, all VLANs are allowed.

The **no autovlan disable** command enables the creation and deletion of VLANs caused by VM activity. This is the default setting.

Command Mode

Vmtracer Configuration

Command Syntax

autovlan disable

no autovlan disable

default autovlan disable

Related Commands

vmtracer session places the switch in the ***vmtracer*** configuration mode.

Example

This command disables dynamic VLAN creation or pruning within the configuration mode VM Tracer session.

```
switch(vmtracer-system_1)# autovlan disable
switch(vmtracer-system_1)#
```

20.6.4.3 password (vmtracer mode)

The **password** command specifies the token that authorizes the username to the vCenter associated with the VM Tracer mode session.

Command Mode

Vmtracer Configuration

Command Syntax

password [**ENCRYPTION**] [*password*]

Parameters

- **ENCRYPTION** encryption level of the password.
  - ***no parameter*** password is a clear-text string.
  - ***0*** the password is a clear-text string. Equivalent to ***no parameter***.
  - **7** the password is an encrypted string.
- ***password*** text that authenticates the username.
  - ***password*** is a clear-text string if **ENCRYPTION** specifies clear text.
  - ***password*** is an encrypted string if **ENCRYPTION** specifies an encrypted string.

Related Commands

vmtracer session places the switch in the ***vmtracer*** configuration mode.

Example

This command configures **abode** as the clear-text string that authorizes the username ***a-switch_01*** located at **example.com/sdk**.

```
switch(vmtracer-system_1)# url https://example.com/sdk
switch(vmtracer-system_1)# username a-switch_01
switch(vmtracer-system_1)# password abcde
switch(vmtracer-system_1)#
```

20.6.4.4 password (vmtracer-vxlan mode)

The **password** command specifies the token that authorizes the username on the NSX for vSphere® (NSX-V) server located at the URL configured for the configuration mode VM Tracer. The switch uses this account to access NSX-V information.

The **password** statement is replaced in ***running-config*** for the configuration mode interface by a subsequent **password** command. The statement is removed by deleting the NSX-V instance through a **no vxlan (vmtracer mode)** command in vmtracer configuration mode.

Command Mode

Vmtracer-vxlan Configuration

Command Syntax

`password` [**ENCRYPTION**] ***password***

Parameters

- **ENCRYPTION** encryption level of the password.
  - ***no parameter*** password is a clear-text string.
  - ***0*** the is a clear-text string. Equivalent to ***no parameter***.
  - ***7*** the password is an encrypted string.
- ***password*** text that autorizes the username.
  - ***password*** is a clear-text string if **ENCRYPTION** specifies clear text.
  - ***password*** is an encrypted string if **ENCRYPTION** specifies an encrypted string.

Related Commands

`vxlan (vmtracer mode)` places the switch in the ***vmtracer-vxlan*** configuration mode.

Example

- This command configures ***5678*** as the clear-text string that authorizes the username **admin** to the NSX-V located at **`https://example.com/sdk`**.

```
switch(config)# vmtracer session vnet-1
switch(config-vmtracer-vnet-1)# vxlan
switch(config-vmtracer-vnet-1-vxlan)# url https://example.com/sdk
switch(config-vmtracer-vnet-1-vxlan)# username admin
switch(config-vmtracer-vnet-1-vxlan)# password 5678
switch(config-vmtracer-vnet-1-vxlan)# exit
switch(config-vmtracer-vnet-1)# show active
vmtracer session vnet-1
   allowed-vlan 1-4094
   vxlan
      url https://example.com/sdk
      username admin
      password 7 s2Xq4GSBlYU=
switch(config-vmtracer-vnet-1)#
```

20.6.4.5 show vmtracer all

The **`show vmtracer all`** command displays VM Tracer data for all switches with the vSphere scope.

Command Mode

EXEC

Command Syntax

`show vmtracer all`

Example

This command displays data for both switches in the vSphere scope.

```
switch> show vmtracer all
Switch : a109(10.10.30.109)
Ethernet49     : 10.102.28.3/10G
   VM Name             VM Adapter          VLAN      Status    State
   ABCD                Network adapter 2   native    Up/--     --

Switch : a164(10.10.30.(172.22.30.164)
```

```
Ethernet49     : 10.102.28.3/10G Storage Network/dvUplink1
   VM Name             VM Adapter          VLAN      Status    State
   WXYZ                Network adapter 2   native    Up/--     --
switch>
```

20.6.4.6 show vmtracer interface

The **show vmtracer interface** command displays the VM interfaces (Vnics) that are active on the VM Tracer enabled interface. For each Vnic, the command displays the name of the attached VM, the adapter name, its VLAN, the VM power state, and the presence status of its MAC address in the switch's MAC table.

Command Mode

EXEC

Command Syntax

show vmtracer interface [**INT_NAME**] [**INFO_LEVEL**]

Parameters

- **INT_NAME** the interfaces to be configured. Values include:
  - ***no parameter*** command returns information for all interfaces.
  - **ethernet *e_range*** Ethernet interface range.
  - **port-channel *p_range*** Port Channel interface range.

    Valid ***e_range*** and ***p_range*** formats include number, number range, or comma-delimited list of numbers and ranges.
- **INFO_LEVEL** specifies information that the command returns.
  - ***no parameter*** connection parameters and status for VM associated to specified sessions.
  - **detail** connection status and data concerning messages the VM.
  - **host** name of the connected host.

Examples

- This command displays the Vnics connected to all VM Tracer enabled interfaces.

```
switch > show vmtracer interface

Ethernet8 : example.com
    VM Name                     VM Adapter          VLAN     Status
    esx3.aristanetworks.com     vmk0                0        Up/Down
    vspheremanagement           Network adapter 1   0        Up/Down

Ethernet15 : example.com
    VM Name                     VM Adapter          VLAN      Status
    Openview                    Network adapter 1   123       Up/Down
    VmTracerVm                  Network adapter 1   123       Down/Down

Ethernet23 : example.com
    VM Name                     VM Adapter          VLAN       Status
 switch>
```

- This command displays the Vnics connected to the ***interface Ethernet8***.

```
switch> show vmtracer interface Ethernet8

Ethernet8 : example.com
    VM Name                     VM Adapter          VLAN      Status
    example.com                 vmk0                0         Up/Down
```

```
    vspheremanagement           Network adapter 1   0         Up/Down
switch>
```

20.6.4.7 show vmtracer session

The **show vmtracer session** command displays vCenter and vShield connection information for a specified VM Tracer session.

Command Mode

EXEC

Command Syntax

show vmtracer session [**SESSION_LIST**]

Parameters

SESSION_LIST VM Tracer sessions for which the command returns information.

- ***no parameter*** all configured VM Tracer sessions.
- ***session_name*** name of one VM Tracer session.

Example

This command displays connection parameters associated to the ***abcde*** session.

```
switch> show vmtracer session abcde
Session abcde
vCenter URL       https://example.com/sdk
username                      Administrator
autovlan                      enabled
allowed-vlans     1-4094
sessionState      Connected
VShield URL       https:/vmware-vshield5.1.xyz.abcde.com
username                  admin
sessionState      Connected switch>
```

20.6.4.8 show vmtracer session vcenter

The **show vmtracer session vcenter** command displays vCenter information for a specified VM Tracer session.

Command Mode

EXEC

Command Syntax

show vmtracer session ***session_name*** **vcenter** [**INFO_LEVEL**]

Parameters

- ***session_name*** VM Tracer sessions for which the command returns information.
- **INFO_LEVEL** specifies information that the command returns.
  - ***no parameter*** displays connection and status information for the specified vCenter.
  - **detail** displays connection, status, and history information for the specified vCenter.

Examples

- This command displays connection parameters for the vCenter associated to the abcde session.

```
switch> show vmtracer session abcde vcenter
```

```
Session           abcde
vCenter URL       https://vmware-vcenter5.1/sdk
username          Administrator
autovlan          enabled
allowed-vlans     1-4094
sessionState      Connected switch>
```

- This command displays connection parameters and history details from the vCenter associated to the abcde session.

```
switch> show vmtracer session abcde vcenter detail

Session                         abcde
vCenter URL                     https://vmware-vcenter5.1/sdk
username                        Administrator
autovlan                        enabled
allowed-vlans                   1-4094
SessionState                    Connected
lastStateChange                 2:46:50 ago
lastMsgSent                     Query network hint message
timeOfLastMsg                   0:00:20 ago
responseTimeForLastMsg          0.000102301000479
numSuccessfulMsg                998
lastSuccessfulMsg               Query network hint message
lastSuccessfulMsgTime           0:00:20 ago
numFailedMsg                    0
lastFailedMsg                   --
lastFailedMsgTime               never
lastErrorCode                   --
switch>
```

20.6.4.9 show vmtracer session vsm

The **show vmtracer session vsm** command displays NSX-V information for a specified VM Tracer session.

Command Mode

EXEC

Command Syntax

show vmtracer session ***session_name*** vsm [**INFO_LEVEL**]

Parameters

- ***session_name*** VM Tracer sessions for which the command returns information.
- **INFO_LEVEL** specifies information that the command returns.
  - ***no parameter*** connection and status information for the specified NSX-V.
  - **detail** connection, status, and history information for the specified NSX-V.

Examples

- This command displays connection parameters for the NSX-V associated to the ***abcde*** session.

```
switch> show vmtracer session abcde vsm

Session           abcde
VShield URL       https://example.com/sdk
username          admin
sessionState      Connected switch>
```

- This command displays connection parameters and history details from the vShield Manager associated to the ***abcde*** session.

```
switch> show vmtracer session abcde vsm detail

Session                         abcde
VShield URL                     https://vmware-vshield5.1/
username                        admin
SessionState                    Connected
LaststateChange                 19 days, 23:14:19 ago
LastMsgSent                     /api/2.0/vdn/scopes
timeOfLastMsg                   1 days, 13:22:09 ago
responseTimeForLastMsg          0.3 sec
numSuccessfulMsg                3649
lastSuccessfulMsg               /api/2.0/vdn/scopes
lastSuccessfulMsgTime           0:00:00 ago
numFailedMsg                    1
lastFailedMsg                   /api/2.0/vdn/config/segments
lastFailedMsgTime               10 days, 1:15:29 ago
lastErrorCode                   CURLE_COULDNT_RESOLVE_HOST - Couldn't resolve
 host

                                The given remote host was not resolved.
switch>
```

20.6.4.10 show vmtracer vm

The **`show vmtracer vm`** command displays VMs interfaces (Vnics) that are accessible to VM Tracer enabled interfaces. For each active VM, the command displays the name of the VM, its adapter, and the hypervisor to which it connects.

Command Mode

EXEC

Command Syntax

`show vmtracer` [**INT_NAME**] `vm` [**VM_LIST**]

Parameters

- **INT_NAME** the interfaces name Values include:
  - ***no parameter*** command returns information for all interfaces.
  - **interface ethernet *e_range*** Ethernet interface range.
  - **interface port-channel *p_range*** Port Channel interface range.

    Valid ***e_range*** and ***p_range*** formats include a number, number range, or comma-delimited list of numbers and ranges.
- **VM_LIST** The virtual machines for which the command displays information. Options include:
  - ***no parameter*** command returns information for all present VMs.
  - ***vm_name*** command returns information only for specified VM.

Related Commands

The **`show vmtracer vm detail`** command displays connection information for one or more specified VMs.

Example

This command displays the VMs connected to all VM Tracer enabled interfaces.

```
switch> show vmtracer vm
VM Name              Esx Host          Interface  VLAN     Status
```

```
vCenter1            172.22.28.8       Po45       native   Down/Down
vCenter2            172.22.28.8       Po45       native   Up/Up
vCenter3            172.22.28.8       Po45       11       Down/Down
vCenter4            172.22.28.8       Po45       native   Down/Down
VMKernel                              Po43       native   Up/Up
demo vcenter 5 clone                  Po43       native   Up/Up switch>
```

20.6.4.11 show vmtracer vm detail

The **`show vmtracer vm detail`** command displays connection data for VMs interfaces (Vnics) that are accessible to VM Tracer enabled interfaces.

Command Mode

EXEC

Command Syntax

`show vmtracer vm` [**VM_LIST**] **`detail`**

Parameters

VM_LIST The virtual machines for which the command displays information. Options include:

- ***no parameter*** command returns information for all present VMs.
- ***vm_name*** command returns information only for specified VM.

Examples

- This command displays connection data for the VMs connected to all VM Tracer enabled interfaces.

```
switch# show vmtracer vm vmcenter1
VM Name  vCenter1 Server App
  Interface   :     Po45
  vNIC        :     Network adapter 1
  MAC         :     00:31:22:8e:b8:41
  Portgroup   :     VM Network
  VLAN        :     native
  Switch      :     Switch2
  Status      :     Down/Down
  Host        :     10.22.18.28
  Data Center :     vcenter-5 switch>
```

- This command displays connection data for the VMs connected to all VM Tracer enabled interfaces.

```
switch> show vmtracer vm detail
VM Name  vCenter1 Server App
  Interface   :     Po45
  vNIC        :     Network adapter 1
  MAC         :     00:31:22:8e:b8:41
  Portgroup   :     VM Network
  VLAN        :     native
  Switch      :     Switch2
  Status      :     Down/Down
  Host        :     10.22.18.28
  Data Center :     vcenter-5

VM Name  vCenter2 Server App
  Interface   :     Po45
  vNIC        :     vmk0
  MAC         :     00:33:23:3c:e1:4e
  Portgroup   :     Management Network
  VLAN        :     native switch>
```

20.6.4.12 show vmtracer vnic counters

The **show vmtracer interface vnic counters** command displays input and output packet counts for VM interfaces (Vnics) that are active on the specified interface or VM.

Command Mode

EXEC

Command Syntax

show vmtracer [**ENTITY**] **vnic counters**

Parameters

ENTITY the virtual machine or interface over which statistics are gathered and displayed.

- ***no parameter*** command returns information for all active VMs.
- **interface ethernet *e_range*** Ethernet interface range.
- **interface port-channel *p_range*** Port Channel interface range.
- **vm *vm_name*** command returns information for specified VM.

  Valid ***e_range*** and ***p_range*** formats include a number, number range, or comma-delimited list of numbers and ranges.

Example

This command displays the Vnics connected to ***interface ethernet 24***.

```
switch> show vmtracer interface ethernet 24 vnic counters
Physical Intf: Ethernet24
Host: 10.17.28.8/site1/dvUplink1
VM Name        vNic                 Input Pkt/Byte/%         Output Pkt/Byte/%
vCenter1       Network adapter 2    2550/   187175/  0.6     6/        360/  0.0
vCenter2       Network adapter 2    418615/ 30678024/ 99.4   1904439/ 1145654613/100.0
Summary                             421165/ 30865199/100.0   1904445/ 1145654973/100.0 switch>
```

20.6.4.13 show vmtracer vxlan segment

The **show vmtracer vxlan segment** command displays information about the VXLAN segments that are managed by the connected NSX for vSphere® (NSX-V).

Command Mode

EXEC

Command Syntax

show vmtracer segment ENTITY

Parameters

ENTITY specifies the information that the command displays. Options include:

- ***no parameter*** displays information for VXLAN segments.
- **pool** displays resource pools available to segments.
- **pool *pool_name*** displays connection information about the specified pool.
- **range** displays the VNI range of the managed segments.

Examples

- This command displays the VXLAN segments managed by the NSX-V.

```
switch> show vmtracer vxlan segment
Name                  VNI     Multicast IP     Network Scope
-------------------------------------------------------------
Eng Wire              5002    237.0.0.1        abcde
```

```
HR Wire               5000    237.0.0.2        abcde switch>
```

- This command displays the resource pools available to the VXLANs.

```
switch> show vmtracer vxlan segment pool
Name                  Description                       Segments
-------------------------------------------------------------------------
abcde                 Spans Cluster 1 and Cluster 2   Eng Wire, HR Wire
 switch>
```

- This command displays connection and packet information for the ***abcde*** pool.

```
switch> show vmtracer vxlan segment pool abcde
Name:          abcde
Description:   Spans Cluster 1 and Cluster 2
Segments:      Eng Wire, HR Wire

Vxlan Segment  Cluster   Host                VTEP IP           DVS       VLAN  MTU
Eng Wire       Cluster2  test2.example.com   10.168.200.1/24   dvs-test2 200   1600
Eng Wire       Cluster1  test2.example.com   10.168.100.1/24   dvs-test1 100   1600
HR Wire        Cluster1  test2.example.com   10.168.100.1/24   dvs-test1 100   1600
HR Wire        Cluster2  test2.example.com   10.168.200.1/24   dvs-test2 200   1600 switch>
```

- This command displays the VNI range of the VXLAN segments.

```
switch> show vmtracer vxlan segment range

VNI Range             Multicast IP Range
-----------------------------------------
5000 - 5024           237.0.0.1 - 237.0.0.117

Name                  VNI     Multicast IP     Network Scope
-------------------------------------------------------------
HR Wire               5002    237.0.0.1        abcde

Eng Wire              5000    237.0.0.2        abcde switch>
```

20.6.4.14 show vmtracer vxlan vm

The **show vmtracer vxlan vm** command displays the VXLAN segments, their VTEP IP numbers, and their VM endpoints that are managed by the connected NSX for vSphere® (NSX-V).

Command Mode

EXEC

Command Syntax

show vmtracer vxlan vm

Example

This command displays the VM endpoints of the VXLAN segments managed by the NSX-V.

```
switch> show vmtracer vxlan vm
Vxlan Segment       VTEP IP             VLAN  VMs
Eng Wire            192.168.200.1/24    200   Eng VM3, Eng VM2
Eng Wire            192.168.100.1/24    100   Eng VM1
HR Wire             192.168.100.1/24    100   HR VM2, HR VM1
HR Wire             192.168.200.1/24    200   --
switch>
```

20.6.4.15 source-interface

The **source-interface** command allows you to connect to a remote vCenter endpoint by using the primary address of the interface as the source IP address. If the interface is not specified, the source IP address is determined by the routing table.

The **no source-interface** and **default source-interface** commands restore default behavior by removing the **source-interface** command from the ***running-config***.

Command Mode

Vmtracer Configuration

Command Syntax

source-interface [**INTERFACE_NAME**]

no source-interface

default source-interface

Parameters

INTERFACE_NAME specifies the interface for which the information is displayed. Options include:

- **Ethernet *e_num*** specifies the Ethernet interface number.
- **Loopback *l_num*** specifies the loopback interface number. Value ranges from ***0*** to ***2100***.
- **Management *m_num*** specifies the management interface number. The values are ***1*** or ***2***.
- **Port-Channel** {***lag_num*** | ***lag_num.sub_num***} specifies the port-channel interface number. Value of interface ranges from ***1*** to ***2000***. Value of sub-interface ranges from ***1*** to ***4094***.
- **Tunnel *tunnel_num*** specifies the tunnel interface number. Value ranges from ***1*** to ***255***.
- **UnconnectedEthernet *port_num*** specifies the unconnected Ethernet port number. Value ranges from ***1*** to ***8***.
- **VLAN *vlan_num*** specifies the VLAN interface number. Value ranges from ***1*** to ***4094***.

Related Commands

The **vmtracer session** command places the switch in the ***vmtracer*** configuration mode.

Examples

- This command configures VM Tracer to use ***interface Ethernet 17*** to derive the source address for session packets.

```
switch(config)# vmtracer session system_1
switch(config-vmtracer-session-system_1)# source-interface Ethernet 17
switch(config-vmtracer-session-system_1)#
```

- This command configures ***interface Loopback 0*** for VM Tracer session.

```
switch(config)# vmtracer session system_1
switch(config-vmtracer-session-system_1)# source-interface Loopback 0
switch(config-vmtracer-session-system_1)#
```

- This command configures ***interface management 1*** for VM Tracer session.

```
switch(config)# vmtracer session system_1
switch(config-vmtracer-session-system_1)# source-interface management 1
switch(config-vmtracer-session-system_1)#
```

- This command configures ***port-channel 10*** for VM Tracer session.

```
switch(config)# vmtracer session system_1
switch(config-vmtracer-session-system_1)# source-interface port-channel 10
```

```
switch(config-vmtracer-session-system_1)#
```

- This command configures ***interface tunnel 25*** for VM Tracer session.

```
switch(config)# vmtracer session system_1
switch(config-vmtracer-session-system_1)# source-interface tunnel 25
switch(config-vmtracer-session-system_1)#
```

- This command configures unconnected ***interface Ethernet 1*** for VM Tracer session.

```
switch(config)# vmtracer session system_1
switch(config-vmtracer-session-system_1)# source-interface unconnected
 Ethernet 1
switch(config-vmtracer-session-system_1)#
```

- This command configures ***interface vlan 25*** for VM Tracer session.

```
switch(config)# vmtracer session system_1
switch(config-vmtracer-session-system_1)# source-interface vlan 25
switch(config-vmtracer-session-system_1)#
```

20.6.4.16 url (vmtracer mode)

The **`url`** command specifies the vCenter server location that is monitored by the session being edited by the current vmtracer mode. The command must reference a fully formed secure url.

Command Mode

Vmtracer Configuration

Command Syntax

`url` ***url_name***

Parameter

url_name location of the vCenter server. Valid formats include IP address (dotted decimal notation) and fully qualified domain name.

Related Commands

The **`vmtracer session`** command places the switch in the ***vmtracer*** configuration mode.

Example

This command specifies the location of the vCenter monitored by the ***system_1*** VM Tracer session.

```
switch(vmtracer-system_1)# url https://example.com/sdk
switch(vmtracer-system_1)#
```

20.6.4.17 url (vmtracer-vxlan mode)

The **`url`** command specifies the NSX for vSphere® (NSX-V) server location that is monitored for VXLAN information by the configuration mode VM Tracer session. The command must reference a fully formed secure URL.

The **`url`** statement is replaced in running-config for the configuration mode session by a subsequent **`url`** command. The statement is removed by deleting the NSX-V instance through a **`no vxlan (vmtracer mode)`** command in ***vmtracer*** configuration mode.

Command Mode

Vmtracer-vxlan Configuration

Command Syntax

`url` ***url_name***

Parameters

url_name location of the NSX-V server. Valid formats include IP address (dotted decimal notation) and fully qualified domain name.

Related Commands

The **`vxlan (vmtracer mode)`** command places the switch in the ***vmtracer-vxlan*** configuration mode.

Example

This command configures the location of the NSX-V monitored by the ***vnet-1*** VM Tracer session.

```
switch(config)# vmtracer session vnet-1
switch(config-vmtracer-vnet-1)# vxlan
switch(config-vmtracer-vnet-1-vxlan)# url https://example.com/sdk
switch(config-vmtracer-vnet-1-vxlan)# exit
switch(config-vmtracer-vnet-1)# show active
vmtracer session vnet-1
   allowed-vlan 1-4094
   vxlan
      url https://example.com/sdk
switch(config-vmtracer-vnet-1)#
```

20.6.4.18 username (vmtracer mode)

The **`username`** command identifies the switch's account name on the vCenter server. The switch uses this user name to access vCenter information.

Command Mode

Vmtracer Configuration

Command Syntax

`username` ***name_string***

Parameters

name_string vCenter account user name. Parameter must match the user name configured on the vCenter.

Related Commands

The **`vmtracer session`** command places the switch in the ***vmtracer*** configuration mode.

Example

This command configures the user name for the vCenter associated with the ***system_1*** session. The session uses this user name to log into the vCenter server.

```
switch(vmtracer-system_1)# username a-switch_01
switch(vmtracer-system_1)#
```

20.6.4.19 username (vmtracer-vxlan mode)

The **`username`** command identifies the switch's account name on the NSX for vSphere® (NSX-V) server located at the URL configured for the configuration mode VM Tracer. The switch uses this user name to access NSX-V information.

The **username** statement is replaced in ***running-config*** for the configuration mode interface by a subsequent **username** command. The statement is removed by deleting the NSX-V instance through a **no vxlan (vmtracer mode)** command in the ***vmtracer*** configuration mode.

Command Mode

Vmtracer-vxlan Configuration

Command Syntax

username ***name_string***

Parameters

name_string NSX-V account user name. Parameter must match a user name configured on the NSX-V.

Related Commands

The **vxlan (vmtracer mode)** command places the switch in the ***vmtracer-vxlan*** configuration mode.

Example

This command configures the user name of admin for the NSX-V located at the URL specified by the URL command.

```
switch(config)# vmtracer session vnet-1
switch(config-vmtracer-vnet-1)# vxlan
switch(config-vmtracer-vnet-1-vxlan)# url https://example.com/sdk
switch(config-vmtracer-vnet-1-vxlan)# username admin
switch(config-vmtracer-vnet-1-vxlan)# exit
switch(config-vmtracer-vnet-1)# show active
vmtracer session vnet-1
   allowed-vlan 1-4094
   vxlan
      url https://example.com/sdk
      username admin
switch(config-vmtracer-vnet-1)#
```

20.6.4.20 vmtracer

The **vmtracer** command enables vmtracer mode on the configuration mode interface. Interfaces with vmtracer mode enabled send discovery packets to the connected vSwitch.

The **no vmtracer** and **default vmtracer** commands disable vmtracer mode on the configuration mode interface by removing the corresponding **vmtracer** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration Interface-Port-channel Configuration

Command Syntax

vmtracer HOST_TYPE

no vmtracer HOST_TYPE

default vmtracer HOST_TYPE

Parameters

HOST_TYPE the type of hypervisor that controls the vSwitch to which the interface connects.

- **vmware-esx** ESX or ESXI hypervisor (VMware).

Examples

- These commands enable the ***vmtracer*** mode on the ***interface Ethernet 3***.

```
switch(config)# interface Ethernet 3
switch(config-if-Et3)# vmtracer vmware-esx
switch(config-if-Et3)#
```

- This command disables the ***vmtracer*** mode on the ***interface Ethernet 3***.

```
switch(config-if-Et3)# no vmtracer vmware-esx
switch(config-if-Et3)#
```

20.6.4.21 vmtracer session

The **`vmtracer session`** command places the switch in the ***vmtracer*** mode for the specified session. The command creates a new session or loads an existing session for editing.

A VM Tracer session connects the switch to a vCenter server at a specified location, then downloads data about VMs and vSwitches managed by ESX hosts connected to switch ports. The switch supports a maximum of four VM Tracer sessions.

VM Tracer session parameters are configured in the ***vmtracer*** mode. Parameters configured in the ***vmtracer*** mode include the vCenter location and dynamic VLAN usage.

The **`no vmtracer session`** and **`default vmtracer session`** commands disable the session and remove its configuration from ***running-config***.

Command Mode

Global Configuration

Command Syntax

`vmtracer session` ***name***

`no vmtracer session` ***name***

`default vmtracer session` ***name***

Parameters

name The label assigned to the VM Tracer session.

Commands Available in vmtracer Configuration Mode

- allowed-vlan
- autovlan disable
- password (vmtracer mode)
- url (vmtracer mode)
- username (vmtracer mode)
- vxlan (vmtracer mode)

Examples

- This command enters vmtracer mode for the ***system_1*** session.

```
switch(config)# vmtracer session system_1
switch(vmtracer-system_1)#
```

- This command disables the ***system_1*** VM Tracer session. The ***system_1*** session and all of its parameters are removed from ***running-config***.

```
switch(config)#no vmtracer session system_1
switch(config)#
```

20.6.4.22 vrf (vmtracer mode)

The **vrf** command allows the switch to communicate with a vCenter server by enabling VmTracer configuration mode. By default, VmTracer is enabled only in the **default vrf** command.

Command Mode

Vmtracer Configuration

Command Syntax

vrf ***vrf_name***

Parameters

vrf_name specifies information of the corresponding VRF.

Example

These commands place the VRF ***vrf1*** in the ***vmtracer*** configuration mode.

```
switch(config)# vmtracer session system_1
switch(config-vmtracer-session-system_1)# vrf vrf1
switch(config-vmtracer-session-system_1)#
```

20.6.4.23 vxlan (vmtracer mode)

The **vxlan** command places the switch in the ***vmtracer-vxlan*** configuration mode. To monitor VXLAN based VMware configurations, the switch must communicate with a NSX for vSphere® (NSX-V). The ***vmtracer-vxlan*** configuration mode specifies the location and user account data that allows the switch to access a NSX-V within the configuration mode ***vmtracer*** session. Each VM Tracer session can be associated with one NSX-V instance.

The **no vxlan** and **default interface vxlan** commands delete the NSX-V instance from the configuration mode ***vmtracer*** session by removing all of the ***vmtracer-vxlan*** mode commands from ***running-config***.

Command Mode

Vmtracer Configuration

Command Syntax

vxlan

no vxlan

default vxlan

Related Commands

The **vmtracer session** command places the switch in the ***vmtracer*** configuration mode.

Commands Available in vmtracer-vxlan Configuration Mode

- password (vmtracer mode)
- url (vmtracer mode)
- username (vmtracer mode)

Example

These commands create the vShield instance for the VMTracer session named ***vnet-1***.

```
switch(config)# vmtracer session vnet-1
switch(config-vmtracer-vnet-1)# vxlan
switch(config-vmtracer-vnet-1-vxlan)#
```

20.7 MapReduce Tracer

This section describes Arista's implementation of MapReduce Tracer, including configuration instructions and command descriptions. Topics covered by this section include:

- MapReduce Tracer Introduction
- MapReduce Tracer Configuration
- Displaying MapReduce Tracer Results
- MapReduce Tracer Commands

20.7.1 MapReduce Tracer Introduction

MapReduce Tracer is a network tool that monitors Hadoop nodes that are directly connected to Arista switches. MapReduce Tracer requires the following:

- Hadoop clusters are deployed with a L3 design.
- The top of rack switch is the default gateway to all attached TaskTracker nodes.
- JobTracker RPC ports do not require authentication.
- Nodes cannot simultaneously belong to multiple Hadoop clusters.
- All TaskTrackers within a cluster are accessed through a common HTTP access port.
- The switch's DNS or static host configuration facilitates TaskTracker name resolution.

Map Reduce Tracer supports these Hadoop releases:

- Apache 0.20.205
- Apache 1.2.1
- Cloudera 3u6
- Cloudera 4.1.3
- Cloudera 4.3.0
- HortonWorks 1.3
- Cloudera 4.5.0

These sections briefly describe Hadoop, Hadoop data structures, and MapReduce Tracer.

20.7.1.1 Hadoop Description

Apache Hadoop is an open-source, Java-based software framework that supports large dataset storage and processing in a distributed computational environment. Hadoop is licensed under Apache License 2.0 and developed through a global community.

Hadoop facilitates application execution on systems composed of thousands of nodes utilizing petabytes of data. Its distributed file system facilitates rapid data transfer among nodes and supports continued operations when individual nodes fail or become inaccessible.

Hadoop Distributed File System (HDFS) is a distributed file-system that stores data on the commodity machines to provide high aggregate bandwidth across the cluster.

20.7.1.2 Hadoop Cluster Structure

A cluster is a group of servers that function as a single system to provide high availability through load balancing and parallel processing. A Hadoop cluster is a type of computational cluster designed for storing and analyzing large amounts of unstructured data in a distributed computing environment.

Typical Hadoop clusters include one master and multiple worker nodes. The master node consists of a TaskTracker, JobTracker, NameNode and DataNode. Worker nodes include a TaskTracker and DataNode.

20.7.1.3 Map Reduce

MapReduce is an algorithm that Hadoop implements to process large datasets by distributing parallel tasks to nodes within a cluster. The MapReduce program includes a Map procedure that filters data and a Reduce procedure that processes the data.

MapReduce manages task and data distribution to cluster nodes such that tasks are executed in parallel, and data transfers between cluster components support redundancy and fault tolerance.

The MapReduce engine consists of one JobTracker and multiple TaskTrackers – all nodes within the Hadoop cluster. The JobTracker receives MapReduce jobs from a client application and manages the completion of these jobs by submitting tasks to available TaskTracker nodes. If a TaskTracker fails to perform the assigned task, the JobTracker reschedules that part of the job to another node.

20.7.1.4 MapReduce Tracer Function

MapReduce Tracer is a feature that tracks and interacts with Hadoop nodes directly connected to Arista switches in a cluster. It communicates with a JobTracker to obtain a list of all nodes in a cluster and then queries JobTracker and TaskTrackers on these nodes for information regarding the jobs they are running and progress of those jobs. This creates a map of TaskTrackers with kinds of jobs they are running. Commands are available to display this data in tables through the CLI and EAPI.

MapReduce Tracer monitors only nodes that connect directly to the switch in L3 networks. Directly connected nodes use the top-of-rack switch as their default gateway and the switch learns ARP entries for these nodes. The list of nodes provided by JobTracker is filtered by tracking ARP entries to remove nodes that are not directly accessible.

MapReduce Tracer creates a database of nodes from the filtered list. After the database is created, the switch queries the JobTracker and TaskTrackers to obtain the following:

- The number of monitored Hadoop nodes.
- The list of monitored nodes, including their IP addresses.
- Jobs that the TaskTrackers are running.
- JobTracker and TaskTracker statistics.

MapReduce Tracer can simultaneously monitor multiple clusters. This means the directly connected TaskTracker nodes can belong to different clusters. A maximum of five clusters are supported per switch.

20.7.2 MapReduce Tracer Configuration

The ***MapReduce Tracer*** configuration commands are structured into two configuration levels:

- The ***monitor-hadoop*** configuration mode is a child of ***global*** configuration mode and controls the ***global MapReduce Tracer*** settings.
- The ***monitor-hadoop-cluster*** configuration mode is a child of the ***monitor-hadoop*** configuration mode and defines polling configurations that monitor individual Hadoop clusters.

These sections describe MapReduce Tracer configuration processes:

- MapReduce Tracer Global Configuration
- Hadoop Cluster Access Configuration
- MapReduce Tracer Example

MapReduce Tracer functions after it is enabled globally. Each polling configuration can be individually enabled after the feature is enabled globally.

20.7.2.1 MapReduce Tracer Global Configuration

MapReduce Tracer global parameters are configured in the ***monitor-hadoop*** configuration mode. Tasks performed from this mode include specifying connection parameters to Hadoop clusters and globally enabling MapReduce Tracer.

Entering the Monitor-Hadoop Configuration Mode

Monitor-hadoop configuration mode is entered using the monitor hadoop command. Monitor-hadoop configuration mode is not a group change mode; statements are immediately stored in ***running-config*** when they are entered through the CLI. The **exit** command returns the switch to global configuration mode.

Examples

- These commands place the switch in the ***monitor-hadoop*** configuration mode.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)#
```

- This command exits the ***monitor-hadoop*** mode.

```
switch(config-monitor-hadoop)# exit
switch(config)#
```

- This command deletes all previously configured ***monitor-hadoop*** configuration mode commands.

```
switch(config)# no monitor hadoop
switch(config)#
```

Globally Enabling MapReduce Tracer

MapReduce Tracer is globally enabled by the **no** version of the **shutdown (Monitor-Hadoop)** command. MapReduce Tracer is globally disabled by default.

Example

These commands globally enable MapReduce Tracer.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# no shutdown
switch(config-monitor-hadoop)# show active
 monitor hadoop
   no shutdown
switch(config-monitor-hadoop)#
```

Creating a Cluster Monitor

A cluster monitor is created by entering the ***monitor-hadoop-cluster*** mode with the **cluster (Monitor Hadoop)** command. Each monitor is labeled with a cluster ID and probes one Hadoop cluster. When the command specifies a monitor with a previously defined cluster ID, subsequent commands edit that monitor's parameters. A monitor with a new cluster ID is created by a command that specifies a nonexistent cluster ID.

Example

These commands enter the **monitor-hadoop-cluster** mode to edit a cluster monitor. The monitor's cluster-id is ***CL2***.

```
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
   cluster CL2
```

```
switch(config-monitor-hadoop-CL2)#
```

20.7.2.2 Hadoop Cluster Access Configuration

Cluster monitors are configured in the ***monitor-hadoop-cluster*** configuration mode. Each monitor corresponds to a hadoop cluster through these configurable parameters:

- JobTracker access parameters (address, port number, and username)
- TaskTracker access port
- Polling interval
- Cluster description
- Enabled setting

The minimum explicit configuration includes JobTracker address and username; default values are defined for all other parameters. By default, cluster monitors are disabled.

The **cluster (Monitor Hadoop)** command places the switch in the ***monitor-hadoop-cluster*** mode for the specified monitor, where a cluster's connection parameters are specified. The ***monitor-hadoop-cluster*** mode is not a group change mode.

A cluster monitor is enabled by using the **no** version of the **shutdown (Monitor-Hadoop)** when MapReduce Tracer is globally enabled.

20.7.2.2.1 JobTracker Configuration

A cluster's JobTracker is located on the master node and schedules work to the cluster's TaskTracker nodes. The **jobtracker (Monitor Hadoop Cluster)** command specifies connection parameters to the monitored cluster.

JobTracker parameters include its node location (IPv4 address or hostname), RPC port, and username. The default RPC port is ***8021***. Location and username parameters do not have default values and must be explicitly configured.

Example

For the ***CL2*** monitor, these commands configure connection parameters to a JobTracker node at ***10.4.4.4*** with the username ***account1***. The default RPC port (***8021***) is implicitly specified.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# jobtracker host 10.4.4.4 username
 account1
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
   cluster CL2
      jobtracker host 10.4.4.4 user account1
switch(config-monitor-hadoop-CL2)#
```

20.7.2.2.2 TaskTracker Configuration

The tasktracker (Monitor Hadoop Cluster) command specifies the HTTP port that access TaskTrackers of the Hadoop cluster probed by the configuration mode monitor. The switch compiles a list of the cluster's TaskTracker addresses by periodically polling the cluster's JobTracker.

The default TaskTracker HTTP port is ***50060***.

Examples

- For the ***CL2*** monitor, these commands configure a TaskTracker access port of ***51000***.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# tasktracker http-port 51000
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
  cluster CL2
    tasktracker http-port 51000
switch(config-monitor-hadoop-CL2)#
```

- These commands restore the default TaskTracker HTTP access port address of ***50060***.

```
switch(config-monitor-hadoop-CL2)# no tasktracker http-port
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
  cluster CL2
switch(config-monitor-hadoop-CL2)# show active all
monitor hadoop
  cluster CL2
    jobtracker rpc-port 8021
    tasktracker http-port 50060
    interval 10
    shutdown
switch(config-monitor-hadoop-CL2)#
```

20.7.2.2.3 Polling Interval Configuration

When the monitor configuration is complete, the switch polls the cluster's JobTracker to maintain the list of active TaskTracker nodes associated with the monitored cluster and compile Hadoop job statistics. The **interval (Monitor Hadoop Cluster)** command specifies the interval between polls to the JobTracker of the monitored cluster. The default interval is ***10*** seconds.

Example

This command sets the JobTracker polling interval to ***25*** seconds for the cluster monitored by the ***CL2*** MapReduce Tracer configuration.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# interval 25
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
   cluster CL2
      interval 25
switch(config-monitor-hadoop-CL2)#
```

20.7.2.3 MapReduce Tracer Example

The commands in this section create monitors that probe two Hadoop clusters, enables each monitor individually, then enables MapReduce Tracer globally. Monitor parameters for the clusters include:

- Cluster ID: ***CL_1***
  - Jobtracker: IP address: ***10.15.2.2***; RPC port: ***8021***; username: ***xyz1***

    - TaskTracker: HTTP address ***54000***
    - JobTracker polling interval: ***10*** seconds (default)
- Cluster ID: ***CL_2***
    - Jobtracker: IP address: ***10.21.5.2***; RPC port: ***9521***; username: ***qrst4***
    - TaskTracker: HTTP address ***50060*** (default)
    - JobTracker polling interval: ***5*** seconds

Example

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL_1

switch(config-monitor-hadoop-CL_1)# jobtracker host 10.15.2.2 username
 xyz1
switch(config-monitor-hadoop-CL_1)# tasktracker http-port 54000
switch(config-monitor-hadoop-CL_1)# no shutdown
switch(config-monitor-hadoop-CL_1)# exit

switch(config-monitor-hadoop)# cluster CL_2
switch(config-monitor-hadoop-CL_2)# jobtracker host 10.21.5.2 rpc-port
 9521
switch(config-monitor-hadoop-CL_2)# interval 5
switch(config-monitor-hadoop-CL_2)# no shutdown
switch(config-monitor-hadoop-CL_2)# exit

switch(config-monitor-hadoop)# no shutdown
switch(config-monitor-hadoop)# show active
 monitor hadoop
   no shutdown
   cluster CL_1
      jobtracker host 10.15.2.2 user xyz1
      tasktracker http-port 54000
      no shutdown
   !
   cluster CL_2
      jobtracker host 10.21.5.2 rpc-port 9521 user qrst4
      interval 5
      no shutdown

switch(config-monitor-hadoop)# show active all
 monitor hadoop
   no shutdown
   cluster CL_1
      jobtracker host 10.15.2.2 rpc-port 8021 user xyz1
      tasktracker http-port 54000
      interval 10
      no shutdown
   !
   cluster CL_2
      jobtracker host 10.21.5.2 rpc-port 9521 user qrst4
      tasktracker http-port 50060
      interval 5
      no shutdown
switch(config-monitor-hadoop)# exit
switch(config)#
```

20.7.3 Displaying MapReduce Tracer Results

MapReduce Tracer display commands provide information about the configuration and activity on the monitored clusters.

This section contains the following topics:

- MapReduce Tracer Status
- Cluster Configuration and Connections
- Job Lists
- Job Data
- TaskTracker Lists
- TaskTracker Connection and Activity
- Data Bursts
- Low-memory Mode

20.7.3.1 MapReduce Tracer Status

MapReduce Tracer status is accessed through **show monitor hadoop status**. Status information includes the enabled status and the number of monitored clusters, TaskTrackers, and locally running jobs.

Example

This command displays MapReduce Tracer status for all connected clusters and TaskTrackers.

```
switch> show monitor hadoop status

Last updated: 2013-10-06 18:14:23
Mapreduce Tracer status:
  Admin status                          : Enabled
  Operational status                    : Enabled
  Number of clusters configured         : 3
  Number of local TaskTrackers          : 4
  Number of jobs running locally        : 4 switch>
```

20.7.3.2 Cluster Configuration and Connections

The following cluster configuration and connection information is available through these commands:

- **show monitor hadoop cluster all**: Configuration and connection data for all monitored clusters.
- **show monitor hadoop cluster status**: Configuration and connection data for a specified cluster.
- **show monitor hadoop tasktracker status**: Connection and activity information for TaskTrackers in a specified cluster, on a specified node, or accessed through a specified interface.

Example

This command displays configuration and connection data for the ***Cluster0*** cluster.

```
switch> show monitor hadoop cluster Cluster0 status

Last updated: 2013-10-06 18:14:23
Cluster status for cluster:  Cluster0
  Admin status                : Enabled
  JobTracker host             : host0
  JobTracker RPC port         : 9000
  JobTracker user             : user0
  JobTracker polling interval : 100 seconds
```

```
  TaskTracker HTTP port       : 8800
  Operational status          : Enabled
  Active TaskTrackers         : 31
  Blacklisted TaskTrackers    : 1
  Decommissioned TaskTrackers : 1
  Tracker expiry interval     : 20.0
  Map slots (used/total)      : 10/100
  Reduce slots (used/total)   : 11/110
  JobTracker heap size        : 1.04GB (max: 2.08GB)

switch>
```

20.7.3.3 Job Lists

The following commands display rosters of currently running job or jobs that previously ran:

- show monitor hadoop: Jobs running on all monitored Hadoop clusters.
- show monitor hadoop cluster counters: Jobs running on a specified cluster and byte counter data.
- show monitor hadoop cluster history: Jobs that previously ran on a specified cluster.
  - clear monitor hadoop job-history: Includes jobs that ran since the monitor was enabled, the switch was reloaded, or the job history was cleared.
- show monitor hadoop cluster jobs: Jobs running on a specified cluster.
- show monitor hadoop history: Jobs that ran on all configured clusters is accessed through.
  - clear monitor hadoop job-history: Includes jobs that ran since the monitor was enabled, the switch was reloaded, or the job history was cleared.
- show monitor hadoop tasktracker counters: Jobs running on a specified TaskTracker and byte counter data.

Examples

- This command displays the jobs that are running on all monitored clusters.

```
switch> show monitor hadoop

Last updated: 2013-10-06 18:14:23
Currently running jobs: 4
JobId   Job Name         Cluster   Maps(#/%)  Reduces(#/%)  Start Time
------- ---------------- --------- ---------- ------------- -------------------
1       ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%      2013-10-06 17:56:03
        gNameForAJob1
2       ShortName2       Cluster0  2/24.68%   0/26.90%      2013-10-06 17:37:43
510001  ReallyAVeryLon\  Cluster1  2/12.34%   0/13.45%      2013-10-06 17:56:03
        gNameForAJob11
510002  ShortName12      Cluster1  2/24.68%   0/26.90%      2013-10-06 17:37:43 switch>
```

- This command displays data the jobs that previously ran on connected Hadoop clusters.

```
switch> show monitor hadoop history

Job history for all clusters:
JobId  Job Name         Cluster   Start Time   End Time     Bytes In  Bytes Out
------ ---------------- --------- ------------ ------------ --------- ---------
2      AReallyBigHist\  Cluster0  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
442    AReallyBigHist\  Cluster1  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
442    AReallyBigHist\  Cluster1  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
2      AReallyBigHist\  Cluster0  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
441    HistoryJob1      Cluster1  2013-10-06   2013-10-08   26.08GB   13.04GB
                                  17:57:43     00:31:03
1      HistoryJob1      Cluster0  2013-10-06   2013-10-08   26.08GB   13.04GB
                                  17:57:43     00:31:03 switch>
```

- This command displays jobs running on cluster ***Cluster0*** and byte counters for each job.

```
switch> show monitor hadoop cluster Cluster0 counters

Last updated: 2013-10-06 18:14:23
Counters for currently running jobs on cluster: Cluster0
 JobId   Job Name          User       Bytes In   Bytes Out  Start Time
------- ----------------- ---------- ---------- ---------- -------------------
 2       ShortName2        JobUser2   37.36GB    76.29MB    2013-10-06 17:37:43
 1       ReallyAVeryLon\   JobUser1   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob1 switch>
```

20.7.3.4 Job Data

The following commands display information about jobs that are running or previously ran on monitored clusters. Available data include job identifiers, JobTracker ID, start time, stop time, data consumption, and progress statistics.

- show monitor hadoop cluster history jobs: Data consumption, start and stop times, and JobTracker ID for a specific job.
- show monitor hadoop cluster jobs <job number>: Data consumption, start and stop times, priority, JobTracker ID, and progress statistics for a specified job.
- show monitor hadoop cluster jobs counter: HDFS (Hadoop Distributed File System) data consumption and and shuffle byte counters for a specified job.
- show monitor hadoop counters: Data through and start time for jobs running on all monitored clusters.
- show monitor hadoop tasktracker jobs: Progress statistics and start times are available for jobs running on specified TaskTracker.
- show monitor hadoop tasktracker running-tasks: Job progress and byte counts of jobs running on a specified Hadoop cluster .
- show monitor hadoop tasktracker running-tasks cluster job task: Progress statistics, HDFS data consumption, start time, and progress information for the specified task of a running job.
- show monitor hadoop tasktracker counters: Data consumption and start times for jobs running on a specified TaskTracker.

Examples

- This command displays information about ***job 1*** that ran on cluster ***Cluster0***.

```
switch> show monitor hadoop cluster Cluster0 history job 1

Job history data for job: HistoryJob1
  Cluster                  : Cluster0
  Job Id                   : 1
  JT Id                    : 201310110013
  User                     : HistoryUser1
  Job start time           : 2013-10-06 17:57:43
  Job end time             : 2013-10-08 00:31:03

Per Interface job counters:
Interface       TaskTracker        Bytes In       Bytes Out
--------------- ------------------ -------------- ---------
Ethernet7       TaskTracker2       26.08GB        13.04GB switch>
```

- This command displays information about ***job 1*** that is running on cluster ***Cluster0***.

```
switch> show monitor hadoop cluster Cluster0 jobs 1

Last updated: 2013-10-06 18:14:23
Information for job: ReallyAVeryLongNameForAJob1 running on cluster:
 Cluster0
```

```
  Cluster                    : Cluster0
  Id                         : 1
  Name                       : ReallyAVeryLongNameForAJob1
  User                       : JobUser1
  Priority                   : veryHigh
  Running state              : running
  Number of map tasks        : 2
  Number of reduce tasks     : 0
  Start time                 : 2013-10-06 17:56:03
  Bytes In                   : 37.36GB
  Bytes Out                  : 76.29MB
  Map Progress               : 12.34%
  Reduce Progress            : 13.45%
  Cleanup Progress           : 14.56%
  Setup Progress             : 15.67%

switch>
```

- This command displays data for jobs running on ***TaskTracker1***.

```
switch> show monitor hadoop tasktracker host TaskTracker1 jobs

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)   Reduces(#/%)  Start Time
------ ---------------- --------- ----------- ------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%    0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%    0/26.90%      2013-10-06 17:37:43 switch>
```

20.7.3.5 TaskTracker Lists

These commands display lists of TaskTrackers that are active on monitored clusters:

- show monitor hadoop cluster tasktracker: TaskTrackers on a specified cluster.
- show monitor hadoop tasktracker all: TaskTrackers on all monitored clusters.

Example

This command displays the TaskTrackers on the ***Cluster0*** cluster.

```
switch> show monitor hadoop cluster Cluster0 tasktracker

Last updated: 2013-10-06 18:14:23
Total 2 TaskTrackers on cluster Cluster0:
Node          IP Address  Interface      Maps   Reduces
------------- ----------- -------------- ------ -------
TaskTracker1  10.100.0.1   Ethernet7      4       0
TaskTracker2  10.100.0.2   Port-Channel7  4       0 switch>
```

20.7.3.6 TaskTracker Connection and Activity

The following TaskTracker connection and activity data is available through these commands:

- show monitor hadoop tasktracker status: Connection and activity information for TaskTrackers on a specified cluster or accessed through a specified interface.
- show monitor hadoop tasktracker all counters: Data consumption for TaskTrackers connected to monitored clusters.

Example

This command displays connection and activity data for TaskTracker on the ***TaskTracker1*** node.

```
switch> show monitor hadoop tasktracker host TaskTracker1 status

Last updated: 2013-10-06 18:14:23
  TaskTracker         : TaskTracker1
  IP Address          : 10.100.0.1
  Interface           : Ethernet7
  State               : active
  Running jobs        : 2
  Running tasks       : 4
  Map Tasks           : 4
  Reduce Tasks        : 0
  Total bytes read    : 2.08GB
  Total bytes written : 4.24MB switch>
```

20.7.3.7 Data Bursts

The **show monitor hadoop traffic burst** command displays the largest data bursts for jobs running on a specified cluster or accessed through a specified node or interface. A data burst is the data consumed during a polling interval.

Example

This command displays traffic burst data for all running jobs that are accessible through ***port channel interface 7***.

```
switch> show monitor hadoop traffic burst interface Port-Channel 7

Last updated: 2013-10-06 18:14:23
Bursts on Interface: 'Port-Channel7' in cluster: Cluster0
Top 2 input bursts:
JobId       Job Name              Burst     Time
----------- --------------------- ----------- -------------------
 1           ShortName             3.07GB    2013-10-06 17:57:43
 2           ReallyAVeryLon\       6.15GB    2013-10-06 17:41:03
             gNameForAJob

Top 2 output bursts:
JobId       Job Name              Burst     Time
----------- --------------------- ----------- -------------------
 1           ShortName             4.10GB    2013-10-06 17:55:13
 2           ReallyAVeryLon\       8.20GB    2013-10-06 17:36:03
             gNameForAJob
```

20.7.3.8 Low-memory Mode

Switch behavior can be improved using the **`memory exhaustion`** command when the switch gets low on available memory. This low-memory or Out-of-Memory (OOM) state is caused by a number of individual or a combination of factors including the one listed below:

- Memory leaks in EOS or customer user processes.
- Process leaks to too many processes or scripts running at one time.
- Over configuration of the system requiring more memory than available.
- Large number of temporary (tmpfs) files such as logs.
- Large number of CLI sessions.
- Any event or agents that can consume more memory than available in the system.

Once the low-memory state is reached, the system uses the **oom_score**, which depends on the value of the **oom_score_adj** parameter to kill processes to free up available memory. The value for the parameter ranges from ***-1000*** (will not be shut down) to ***1000*** (will be shut down first) during an OOM state. Non-essential agents are assigned **oom_score_adj** values of ***900*** or greater.

When enabled, the system implements the following:

- Non-essential agents shut down by Linux are not restarted.
- CLI is restarted, and a message: "Restarting CLI due to memory exhaustion on the system" is sent to all login sessions in the switch.
- CLI Scheduler is disabled.

The system requires manual intervention through **`reset system memory exhaustion`** command to exit the mode and restart the agents that were shut down.

Configuring Low-memory Mode

The following examples are applicable to all platforms.

Example

These commands configure the low-memory mode.

```
switch(config)# monitor system
switch(config-monitor-system)# memory exhaustion
switch(config-monitor-system-memory)# action agent non-critical hold-
down
```

Example

The following command disables low-memory mode in ***config-monitor-system-memory***.

```
switch(config-monitor-system-memory)# no action agent non-critical
 hold-down
```

The following command exits low-memory mode after an OOM event has occurred and agents have been shut down. The command restarts the agents being held down and reenables the CLI scheduler.

```
switch(config-monitor)# reset system memory exhaustion
```

The **`show monitor system`** command shows the state of the system with the following information:

- If the feature is enabled.
- If the system is in low-memory mode (featured enabled and system experiencing OOM).
- How many times the system entered low-memory mode.
- Last time the system entered low-memory mode (if applicable).
- Last time the system exited low-memory mode (if applicable).

Example

```
switch# show monitor system
```

The following output is for a system where the feature is disabled and the system has never entered the low-memory mode.

```
Memory Exhaustion Feature enabled: False
System currently in Memory Exhaustion: False
Number of times system entered Memory Exhaustion:
```

The following output is for a system where the feature is enabled and the system has never entered the low-memory mode.

```
Memory Exhaustion Feature enabled: True
System currently in Memory Exhaustion: False
Number of times system entered Memory Exhaustion: 0
```

The following output is for a system where the feature is enabled and the system enters the low-memory mode.

```
Memory Exhaustion Feature enabled: True
System currently in Memory Exhaustion: True
Number of times system entered Memory Exhaustion: 1
Last time entered in Memory Exhaustion: 0:00:07 ago
```

The following output is for a system where the feature is enabled and the system enters the low-memory mode once, and has exited the low-memory mode.

```
Memory Exhaustion Feature enabled: True
System currently in Memory Exhaustion: False
Number of times system entered Memory Exhaustion: 1
Last time entered in Memory Exhaustion: 0:01:00 ago
Last time exited from Memory Exhaustion: 0:00:05 ago
```

Configuring oom_score_adj for a daemon agent

To configure the ***oom_score_adj*** of any configured daemon, use the following command:

```
switch(config-daemon-<daemon_name>)# oom score adjustment <score>
```

Note: Similar to change the daemon heartbeat value, the new ***oom_score_adj*** value for the daemon is in effect after restarting the daemon by using **`shutdown`** command followed by a **`no shutdown`** command.

```
switch(config)# daemon daemonName
switch(config-daemon-daemonName)# oom score adjustment -200
switch(config-daemon-daemonName)# shutdown
switch(config-daemon-daemonName)# no shutdown
```

20.7.4 MapReduce Tracer Commands

Global Configuration Commands

- monitor hadoop

Clear Hadoop Monitor Commands

- clear monitor hadoop burst-counters
- clear monitor hadoop job-history

Display Commands

- show agent oom scores
- show monitor hadoop
- show monitor hadoop cluster all
- show monitor hadoop cluster counters
- show monitor hadoop cluster history
- show monitor hadoop cluster history jobs
- show monitor hadoop cluster jobs
- show monitor hadoop cluster jobs <job number>
- show monitor hadoop cluster jobs counter
- show monitor hadoop cluster status
- show monitor hadoop cluster tasktracker
- show monitor hadoop counters
- show monitor hadoop history
- show monitor hadoop status
- show monitor hadoop tasktracker all
- show monitor hadoop tasktracker all counters
- show monitor hadoop tasktracker counters
- show monitor hadoop tasktracker jobs
- show monitor hadoop tasktracker running-tasks
- show monitor hadoop tasktracker running-tasks cluster job task
- show monitor hadoop tasktracker status
- show monitor hadoop traffic burst
- show monitor system

Hadoop Commands

- cluster (Monitor Hadoop)
- shutdown (Monitor-Hadoop)

Hadoop-Cluster Commands

- description (Monitor Hadoop Cluster)
- interval (Monitor Hadoop Cluster)
- jobtracker (Monitor Hadoop Cluster)
- shutdown (Monitor Hadoop Cluster)
- tasktracker (Monitor Hadoop Cluster)

20.7.4.1 clear monitor hadoop burst-counters

The **`clear monitor hadoop burst-counters`** command resets MapReduce Tracer burst counters for all jobs running on specified clusters.

Command Mode

Privileged EXEC

Command Syntax

`clear monitor hadoop burst-counters` [**CLUSTERS**]

Parameters

- **CLUSTERS** Hadoop clusters for which command displays data. Options include:
  - ***no parameter*** All clusters.

- **cluster *c_name*** Cluster name.

Example

This command clears the burst counters for all jobs running on ***CL2*** cluster.

```
switch# clear monitor hadoop burst-counters cluster CL2
Cleared burst counters switch#
```

20.7.4.2 clear monitor hadoop job-history

The **`clear monitor hadoop job-history`** command resets the job history database for all specified clusters.

Command Mode

Privileged EXEC

Command Syntax

`clear monitor hadoop job-history` [**CLUSTERS**]

Parameters

- **CLUSTERS** Hadoop clusters for which command displays data. Options include:
  - ***no parameter*** All clusters.
  - **cluster *c_name*** Cluster name.

Example

This command clears the job history on the ***CL2*** cluster.

```
switch# clear monitor hadoop job-history cluster CL2
Cleared job history switch#
```

20.7.4.3 cluster (Monitor Hadoop)

The **`cluster`** command is a monitor-hadoop command that places the switch in the ***monitor-hadoop-cluster mode*** for configuring and enabling a MapReduce Tracer monitor for a Hadoop cluster. The command either accesses an existing monitor configuration or creates a monitor.

The ***monitor-hadoop-cluster*** configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting the ***monitor-hadoop-cluster*** mode does not affect ***running-config***. The **`exit`** command returns the switch to the ***monitor-hadoop*** configuration mode.

The configuration mode monitor is enabled by the **no** version of the shutdown (Monitor Hadoop Cluster). Enabling a monitor also requires that MapReduce Tracer is globally enabled (shutdown (Monitor Hadoop Cluster).

The **`no cluster`** and **`default cluster`** commands remove the specified Hadoop cluster configuration from ***running-config***.

Command Mode

Monitor-hadoop Configuration

Command Syntax

`cluster` ***cluster_name***

`no cluster` ***cluster_name***

`default cluster` ***cluster_name***

Parameters

cluster_name Hadoop cluster name.

Related Command

The monitor hadoop command places the switch in the ***monitor-hadoop*** configuration mode.

Commands Available in Monitor-hadoop-cluster Configuration Mode

- description (Monitor Hadoop Cluster)
- interval (Monitor Hadoop Cluster)
- jobtracker (Monitor Hadoop Cluster)
- shutdown (Monitor Hadoop Cluster)
- tasktracker (Monitor Hadoop Cluster)

Examples

- These commands create the CL2 monitor and enters the ***monitor-hadoop-cluster*** mode for the monitor.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
   cluster CL2
switch(config-monitor-hadoop-CL2)#
```

- These commands exit the ***monitor-hadoop-cluster*** mode.

```
switch(config-monitor-hadoop-CL2)# exit
switch(config-monitor-hadoop)# show active
 monitor hadoop
   cluster CL2
switch(config-monitor-hadoop)#
```

- These commands remove the CL2 monitor.

```
switch(config-monitor-hadoop)# no cluster CL2
switch(config-monitor-hadoop)# show active
switch(config-monitor-hadoop)#
```

20.7.4.4 description (Monitor Hadoop Cluster)

The **description** command adds a text string to the configuration mode MapReduce Tracer cluster monitor. The string has no functional impact on the monitor.

The **no description** and **default description** commands remove the text string from the configuration mode monitor by removing the corresponding **description** command from ***running-config***.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

description ***label_text***

no description

default description

Parameters

label_text Character string assigned to the monitor configuration.

Related Commands

The **cluster (Monitor Hadoop)** command places the switch in the ***monitor-hadoop-cluster*** configuration mode.

Example

These commands add description text to the ***CL2*** monitor.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# description First Cluster
monitor hadoop
   cluster CL2
      description First Cluster
      jobtracker host 10.3.3.3 user JANE
switch(config-monitor-hadoop-CL2)#
```

20.7.4.5 interval (Monitor Hadoop Cluster)

The **interval** command specifies the polling interval between queries to the Hadoop cluster JobTracker specifed by configuration mode statements. The switch polls a cluster's JobTracker to update its list of active TaskTracker nodes and the statistics of jobs running in the cluster. This command controls the frequency of these polls. The default interval is ***10*** seconds.

The **no interval** and **default interval** commands restore the default interval of ***10*** seconds by removing the **interval** command from ***running-config***.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

interval ***period***

no interval

default interval

Parameters

period Interval (seconds) between JobTracker polls. Value ranges from ***1*** to ***600***. Default is ***10***.

Related Commands

The **cluster (Monitor Hadoop)** command places the switch in the ***monitor-hadoop-cluster*** configuration mode.

Example

This command sets the JobTracker polling interval to ***25*** seconds for the ***CL2*** cluster configuration.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# interval 25
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
   cluster CL2
      interval 25
switch(config-monitor-hadoop-CL2)#
```

20.7.4.6 jobtracker (Monitor Hadoop Cluster)

The **jobtracker** command specifies JobTracker access parameters for the cluster monitored by configuration mode monitor statements. A cluster's JobTracker is located on the master node and schedules work to the cluster's TaskTracker nodes.

Parameters required to communicate with a JobTracker include its node location (IPv4 address or hostname), RPC port, and username. The default RPC port is ***8021***. Location and username parameters do not have default values and must be explicity configured. A JobTracker command that specifies a partial parameter list modifies the existing corresponding **jobtracker** statement in ***running-config***.

The **no jobtracker** and **default jobtracker** commands perform the following:

- removes the **jobtracker** statement from ***running config*** when it lists all command parameters.
- modifies the existing **jobtracker** statement when it lists a subset of command parameters.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

jobtracker [**LOCATION**][**PORT**][**USER**]

no jobtracker[**LOCATION**][**PORT**][**USER**]

default jobtracker [**LOCATION**][][**USER**]

All parameters may be placed in any order.

Parameters

- **LOCATION** Address or hostname of JobTracker node. Options include:
  - ***no parameter*** Location remains undefined or unchanged from a previous configuration.
  - **host *ipv4_addr*** IPv4 address of master (JobTracker) node.
  - **host *hostname*** Hostname of master (JobTracker) node.
- **PORT** JobTracker RPC port number. Default value is ***8021***. Options include:
  - ***no parameter*** Port number remains unchanged from previous configuration.
  - **rdp-port *port_num*** Port number of master (JobTracker) node. Value ranges from ***1*** to ***65535***.
- **USER** Username that accesses JobTracker node. Options include:
  - ***no parameter*** Username remains undefined or unchanged from previous configuration.
  - **username *name_string*** JobTracker username.

Related Command

The **cluster (Monitor Hadoop)** command places the switch in the ***monitor-hadoop-cluster*** configuration mode.

Examples

- For the CL2 cluster configuration, these commands establish a connection to the JobTracker node at ***10.4.4.4*** with the username ***account1***. The default RPC port (***8021***) is implicitly specified.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# jobtracker host 10.4.4.4 username
 account1
switch(config-monitor-hadoop-CL2)# show active

monitor hadoop
   cluster CL2
      jobtracker host 10.4.4.4 user account1
```

```
switch(config-monitor-hadoop-CL2)#
```

- These commands modify the JobTracker configuration to specify an RPC port of ***9000***.

```
switch(config-monitor-hadoop-CL2)# jobtracker rpc-port 9000
switch(config-monitor-hadoop-CL2)# show active

monitor hadoop
   cluster CL2
      jobtracker host 10.4.4.4 rpc-port 9000 user account1

switch(config-monitor-hadoop-CL2)#
```

20.7.4.7 monitor hadoop

The **`monitor hadoop`** command places the switch in the ***monitor-hadoop*** configuration mode for configuring MapReduce Tracer monitors. A MapReduce Tracer monitor interacts with Hadoop cluster nodes that are directly attached to the switch. Tasks that the switch can perform through this interaction include:

- compile a list of nodes in the cluster.
- compile a list of jobs the nodes are running.
- download progress of the running jobs.

The ***monitor-hadoop*** configuration mode is not a group change mode; the ***running-config*** is changed immediately upon entering commands. Exiting the ***monitor-hadoop*** configuration mode does not affect the ***running-config***. The **`exit`** command returns the switch to the ***global*** configuration mode. MapReduce Tracer is enabled in the ***monitor-hadoop*** mode through the **no** version of the **shutdown (Monitor Hadoop Cluster)** command.

The **`no monitor hadoop`** and **`default monitor hadoop`** commands delete previously configured **`monitor hadoop mode`** configuration commands.

Command Mode

Global Configuration

Command Syntax

`monitor hadoop`

`no monitor hadoop`

`default monitor hadoop`

Commands Available in Monitor-hadoop Configuration Mode

- **cluster (Monitor Hadoop)**
- **shutdown (Monitor Hadoop Cluster)**

Examples

- These commands place the switch in the ***monitor-hadoop*** configuration mode.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)#
```

- This command exits ythe *monitor-hadoop* mode.

```
switch(config-monitor-hadoop)# exit
switch(config)#
```

- This command deletes all previously configured ***monitor-hadoop*** configuration mode commands.

```
switch(config)# no monitor hadoop
```

```
switch(config)#
```

20.7.4.8 show agent oom scores

The **show agent oom scores** command displays information for each agent if specified.

Command Mode

EXEC

Command Syntax

show agent [agentName] ***oom scores***

Parameters

- **Display Values**
  - **Agent name** Name(s) of agent(s) - processes.
  - **RssAnon** Size of the resident anonymous memory for the given process.
  - **VmSize** Current total virtual memory size for the process.
  - **Max VmSize** Peak value of total virtual memory for the process.
  - **oom_score** Parameter used in determining which process to hold down.
  - **oom_score_adj** Parameter used to adjust ***oom_score*.**

Example

This command displays the information for SNMP and Sysdb.

```
switch# show agent Snmp Sysdb oom scores
Agent Name  RssAnon  VmSize  Max VmSize  oom_score oom_score_adj

---------- --------- ------- ----------  --------- ------------
Snmp          34.8MB  536.1MB   541.9MB      911           900
Sysdb         102.3MB 597.5MB   597.5MB      0            -500
```

20.7.4.9 show monitor hadoop

The **show monitor hadoop** command displays a list of jobs that are running on all monitored Hadoop clusters.

Command Mode

EXEC

Command Syntax

show monitor hadoop

Example

This command displays the jobs that are running on all monitored clusters.

```
switch> show monitor hadoop

Last updated: 2013-10-06 18:14:23
Currently running jobs: 4
JobId  Job Name         Cluster   Maps(#/%)  Reduces(#/%)  Start Time
-----  ---------------- --------- ---------- ------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%   0/26.90%      2013-10-06 17:37:43
510001 ReallyAVeryLon\  Cluster1  2/12.34%   0/13.45%      2013-10-06 17:56:03
       gNameForAJob11
510002 ShortName12      Cluster1  2/24.68%   0/26.90%      2013-10-06 17:37:43 switch>
```

20.7.4.10 show monitor hadoop cluster all

The **show monitor hadoop cluster all** command displays configuration and connection information for all monitored Hadoop clusters.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster all

Example

This command displays configuration and connection data for all connected Hadoop clusters.

```
switch> show monitor hadoop cluster all

Total number of clusters configured: 3
  Cluster                      : Cluster0
  Admin status                 : Enabled
  JobTracker host              : host0
  JobTracker RPC port          : 9000
  JobTracker user              : user0
  JobTracker polling interval : 100 seconds
  TaskTracker HTTP port        : 8800
  Operational status           : Enabled
  Active TaskTrackers          : 31
  Blacklisted TaskTrackers     : 1
  Decommissioned TaskTrackers : 1
  Tracker expiry interval      : 20.0
  Map slots (used/total)       : 10/100
  Reduce slots (used/total)    : 11/110
  JobTracker heap size         : 1.04GB (max: 2.08GB)

  Cluster                      : Cluster1
  Admin status                 : Enabled
  JobTracker host              : host1
  JobTracker RPC port          : 9001
  JobTracker user              : user1
  JobTracker polling interval : 101 seconds
  TaskTracker HTTP port        : 8801
  Operational status           : Enabled
  Active TaskTrackers          : 32
  Blacklisted TaskTrackers     : 0
  Decommissioned TaskTrackers : 0
  Tracker expiry interval      : 40.0
  Map slots (used/total)       : 20/200
  Reduce slots (used/total)    : 22/220
  JobTracker heap size         : 2.09GB (max: 4.15GB)

  Cluster                      : Cluster2
  Admin status                 : Disabled
  JobTracker host              : host2
  JobTracker RPC port          : 9002
  JobTracker user              : user2
  JobTracker polling interval : 102 seconds
  TaskTracker HTTP port        : 8802
  Operational status           : Disabled
```

20.7.4.11 show monitor hadoop cluster counters

The **show monitor hadoop cluster counters** command displays a list of jobs running on the specified Hadoop cluster and data consumption associated with these jobs.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster ***c_name*** counters

Parameters

c_name Cluster name.

Example

This command displays jobs running on cluster ***Cluster0***.

```
switch> show monitor hadoop cluster Cluster0 counters

Last updated: 2013-10-06 18:14:23
Counters for currently running jobs on cluster: Cluster0
JobId   Job Name          User Bytes In Bytes   Out Start  Time
------- ----------------- ---------- ---------- ----------- -------------------
2       ShortName2        JobUser2   37.36GB    76.29MB     2013-10-06 17:37:43
1       ReallyAVeryLon\   JobUser1   37.36GB    76.29MB     2013-10-06 17:56:03
        gNameForAJob1 switch>
```

20.7.4.12 show monitor hadoop cluster history

The **show monitor hadoop cluster history** command displays all jobs that ran on the specified cluster. The list includes all jobs that ran since the switch was reloaded, the job history was cleared (**clear monitor hadoop job-history**), or MapReduce Tracer was enabled.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster ***c_name*** history

Parameters

c_name Cluster name.

Example

This command displays the jobs that were ran on the cluster named ***Cluster0***.

```
switch> show monitor hadoop cluster Cluster0 history

Jobs history on cluster: Cluster0
JobId    Job Name           Start Time    End Time       Bytes In     Bytes Out
-------- ------------------ ------------- -------------- ------------ ---------
2        AReallyBigHist\    2013-10-06    2013-10-09     26.08GB      13.04GB
         oricalJobName      17:41:03      06:47:43
2        AReallyBigHist\    2013-10-06    2013-10-09     26.08GB      13.04GB
         oricalJobName      17:41:03      06:47:43
1        HistoryJob1        2013-10-06    2013-10-08     26.08GB      13.04GB
                            17:57:43      00:31:03 switch>
```

20.7.4.13 show monitor hadoop cluster history jobs

The **show monitor hadoop cluster history jobs** command displays data about the specified job. Hadoop jobs are identified by job number and the cluster that ran the job.

Data that the command returns include job identifiers, JobTracker ID, start and stop times, and data consumption.

Command Mode

EXEC

Command Syntax

`show monitor hadoop cluster` ***c_name*** history jobs ***job_number***

Parameters

- ***c_name*** Cluster name.
- ***job_number*** Job number. Value ranges from ***0*** to ***2147483647***.

Example

This command displays information about ***job 1*** that ran on cluster ***Cluster0***.

```
switch> show monitor hadoop cluster Cluster0 history job 1

Job history data for job: HistoryJob1
  Cluster               : Cluster0
  Job Id                : 1
  JT Id                 : 201310110013
  User                  : HistoryUser1
  Job start time        : 2013-10-06 17:57:43
  Job end time          : 2013-10-08 00:31:03

Per Interface job counters:
Interface       TaskTracker        Bytes In       Bytes Out
--------------- ------------------ -------------- ---------
Ethernet7       TaskTracker2       26.08GB        13.04GB switch>
```

20.7.4.14 show monitor hadoop cluster jobs

The **`show monitor hadoop cluster jobs`** command displays a list of jobs that are running on the specified cluster.

Command Mode

EXEC

Command Syntax

`show monitor hadoop cluster` ***c_name*** jobs

Parameters

c_name Cluster name.

Example

This command displays the list of jobs running on cluster ***Cluster0***.

```
switch> show monitor hadoop cluster Cluster0 jobs

Last updated: 2013-10-06 18:14:23
Currently running jobs on cluster: Cluster0
JobId    Job Name           User        Maps    Reduces     Start Time
-------- ------------------ ----------- ------- ----------- -------------------
2        ShortName2         JobUser2    2       0           2013-10-06 17:37:43
1        ReallyAVeryLon\    JobUser1    2       0           2013-10-06 17:56:03
         gNameForAJob1 switch>
```

20.7.4.15 show monitor hadoop cluster jobs counter

The **`show monitor hadoop cluster jobs counter`** command displays data consumption and progress statistics for the specified job.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* [jobs *job_number*] counter

Parameters

- ***c_name*** Cluster name.
- ***job_number*** Job number. Value ranges from ***0*** to ***2147483647***.

Example

This command displays byte counters for the job named ***1*** that is running on the cluster named ***Cluster0***.

```
switch> show monitor hadoop cluster Cluster0 jobs 1 counters

Last updated: 2013-10-06 18:14:23
Cluster : Cluster0
Job Name : ReallyAVeryLongNameForAJob1
Job Id : 1
Interface         HDFS Bytes Read    HDFS Bytes Written    Reduce Shuffle Bytes
----------------- ------------------ --------------------- --------------------
Port-Channel8     4.14GB             8.48MB                12.72MB
Port-Channel9     6.21GB             12.72MB               19.07MB
Ethernet8         3.10GB             6.36MB                9.54MB
Ethernet9         5.17GB             10.60MB               15.89MB
Port-Channel7     2.07GB             4.24MB                6.36MB
Ethernet10        7.24GB             14.83MB               22.25MB
Port-Channel10    8.28GB             16.95MB               25.43MB
Ethernet7         1.03GB             2.12MB                3.18MB switch>
```

20.7.4.16 show monitor hadoop cluster jobs <job number>

The **show monitor hadoop cluster jobs*job number*** command displays information about the specified job. Hadoop jobs are identified by job ID and the cluster that is running the job.

Data that the command returns include time of update, job identifiers, start times, data consumption, and completion progress.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* [jobs *job_number*]

Parameters

- ***c_name*** Cluster name.
- ***job_number*** Job number. Value ranges from ***0*** to ***2147483647***.

Example

This command displays information about ***job 1*** that is running on cluster ***Cluster0***.

```
switch> show monitor hadoop cluster Cluster0 jobs 1

Last updated: 2013-10-06 18:14:23
Information for job: ReallyAVeryLongNameForAJob1 running on cluster: Cluster0
  Cluster : Cluster0
  Id                    : 1
  Name                  : ReallyAVeryLongNameForAJob1
  User                  : JobUser1
  Priority              : veryHigh
  Running state         : running
```

```
  Number of map tasks    : 2
  Number of reduce tasks : 0
  Start time             : 2013-10-06 17:56:03
  Bytes In               : 37.36GB
  Bytes Out              : 76.29MB
  Map Progress           : 12.34%
  Reduce Progress        : 13.45%
  Cleanup Progress       : 14.56%
  Setup Progress         : 15.67%

switch>
```

20.7.4.17 show monitor hadoop cluster status

The **show monitor hadoop cluster status** command displays configuration and connection information for the specified cluster.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster ***c_name*** status

Parameters

c_name Cluster name.

Example

This command displays configuration and connection data for the ***Cluster0*** cluster.

```
switch> show monitor hadoop cluster Cluster0 status

Last updated: 2013-10-06 18:14:23
Cluster status for cluster: Cluster0
  Admin status                 : Enabled
  JobTracker host              : host0
  JobTracker RPC port          : 9000
  JobTracker user              : user0
  JobTracker polling interval  : 100 seconds
  TaskTracker HTTP port        : 8800
  Operational status           : Enabled
  Active TaskTrackers          : 31
  Blacklisted TaskTrackers     : 1
  Decommissioned TaskTrackers  : 1
  Tracker expiry interval      : 20.0
  Map slots (used/total)       : 10/100
  Reduce slots (used/total)    : 11/110
  JobTracker heap size         : 1.04GB (max: 2.08GB)

switch>
```

20.7.4.18 show monitor hadoop cluster tasktracker

The **show monitor hadoop cluster tasktracker** command displays a list of TaskTrackers in the specified cluster. The IP address and access interface is included in the table.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* tasktracker

Parameters

c_name Cluster name.

Example

This command displays the TaskTrackers on the ***Cluster0*** cluster.

```
switch> show monitor hadoop cluster Cluster0 tasktracker

Last updated: 2013-10-06 18:14:23
Total 2 TaskTrackers on cluster Cluster0:
Node          IP Address   Interface      Maps   Reduces
------------- ------------ -------------- ------ -------
TaskTracker1   10.100.0.1   Ethernet7      4      0
TaskTracker2   10.100.0.2   Port-Channel7  4      0 switch>
```

20.7.4.19 show monitor hadoop counters

The **show monitor hadoop counters** command displays byte counter data for all jobs running on clusters for which MapReduce Tracer is configured.

Command Mode

EXEC

Command Syntax

show monitor hadoop counters

Example

This command displays byte counter data for all jobs running on clusters that the switch is accessing through MapReduce Tracer.

```
switch> show monitor hadoop counters

Last updated: 2013-10-06 18:14:23
Counters for running jobs:
JobId   Job Name          Cluster    Bytes In   Bytes Out   Start Time
------- ----------------- ---------- ---------- ----------- -------------------
510002  ShortName12       Cluster1   37.36GB    76.29MB     2013-10-06 17:37:43
510001  ReallyAVeryLon\   Cluster1   37.36GB    76.29MB     2013-10-06 17:56:03
        gNameForAJob11
2       ShortName2        Cluster0   37.36GB    76.29MB     2013-10-06 17:37:43
1       ReallyAVeryLon\   Cluster0   37.36GB    76.29MB     2013-10-06 17:56:03
        gNameForAJob1 switch>
```

20.7.4.20 show monitor hadoop history

The **show monitor hadoop history** command displays jobs that ran on clusters for which MapReduce Tracer is configured. The list includes all jobs that ran since the switch was reloaded, MapReduce Tracer was enabled, or the job history was cleared (**clear monitor hadoop job-history**).

Command Mode

EXEC

Command Syntax

show monitor hadoop history

Example

This command displays data that jobs that previously ran on connected Hadoop clusters.

```
switch> show monitor hadoop history

Job history for all clusters:
JobId  Job Name         Cluster   Start Time  End Time     Bytes In   Bytes Out
------ ---------------- --------- ----------- ------------ ---------- ---------
2      AReallyBigHist\  Cluster0  2013-10-06  2013-10-09   26.08GB    13.04GB
       oricalJobName              17:41:03    06:47:43
442    AReallyBigHist\  Cluster1  2013-10-06  2013-10-09   26.08GB    13.04GB
       oricalJobName              17:41:03    06:47:43
442    AReallyBigHist\  Cluster1  2013-10-06  2013-10-09   26.08GB    13.04GB
       oricalJobName              17:41:03    06:47:43
2      AReallyBigHist\  Cluster0  2013-10-06  2013-10-09   26.08GB    13.04GB
       oricalJobName              17:41:03    06:47:43
441    HistoryJob1      Cluster1  2013-10-06  2013-10-08   26.08GB    13.04GB
                                  17:57:43    00:31:03
1      HistoryJob1      Cluster0  2013-10-06  2013-10-08   26.08GB    13.04GB
                                  17:57:43    00:31:03 switch>
```

20.7.4.21 show monitor hadoop status

The **show monitor hadoop status** command displays system status for MapReduce Tracer.

Command Mode

EXEC

Command Syntax

show monitor hadoop status

Example

This command displays MapReduce Tracer status for all connected clusters and TaskTrackers.

```
switch> show monitor hadoop status
Last updated: 2013-10-06 18:14:23
Mapreduce Tracer status:
  Admin status                   : Enabled
  Operational status             : Enabled
  Number of clusters configured  : 3
  Number of local TaskTrackers   : 4
  Number of jobs running locally : 4 switch>
```

20.7.4.22 show monitor hadoop tasktracker all

The **show monitor hadoop tasktracker all** command displays a list of TaskTrackers that are on all monitored Hadoop clusters.

Command Mode

EXEC

Command Syntax

show monitor hadoop tasktracker all

Example

This command displays the TaskTrackers of all monitored clusters that are connected to the switch.

```
switch> show monitor hadoop tasktracker all

Last updated: 2013-10-06 18:14:23
All local TaskTrackers:
Node             Cluster      IP Address     Interface         Maps     Reduces
---------------- ------------ -------------- ----------------- -------- -------
TaskTracker1     Cluster0     10.100.0.1      Ethernet7         4        0
```

```
TaskTracker3     Cluster1     10.100.0.3      Ethernet8         4        0
TaskTracker2     Cluster0     10.100.0.2      Port-Channel7     4        0
TaskTracker4     Cluster1     10.100.0.4      Port-Channel8     4        0 switch>
```

20.7.4.23 show monitor hadoop tasktracker all counters

The **show monitor hadoop tasktracker all counters** command displays byte counters for the TaskTrackers of all monitored Hadoop clusters.

Command Mode

EXEC

Command Syntax

show monitor hadoop tasktracker all counters

Example

This command displays byte counter data for the TaskTrackers servicing all MapReduce Tracer Hadoop clusters.

```
switch> show monitor hadoop tasktracker all counters

Last updated: 2013-10-06 18:14:23
Counters for all TaskTrackers:
Node             IP Address     Interface         Bytes Read      Bytes Written
---------------- -------------- ----------------- --------------- -------------
TaskTracker1     10.100.0.1     Ethernet7         2.08GB          4.24MB
TaskTracker3     10.100.0.3     Ethernet8         6.23GB          12.72MB
TaskTracker2     10.100.0.2     Port-Channel7     4.15GB          8.48MB
TaskTracker4     10.100.0.4     Port-Channel8     8.30GB          16.95MB switch>
```

20.7.4.24 show monitor hadoop tasktracker counters

The **show monitor hadoop tasktracker counters** command displays a list of jobs running on the specified TaskTracker and output from byte counters associated with these jobs.

Command Mode

EXEC

Command Syntax

show monitor hadoop tasktracker NODES counters

Parameters

- **NODES** TaskTracker node access point. Options include:
  - **host *hostname*** Node name.
  - **interface ethernet *e_range*** Ethernet interfaces through which node connects.
  - **interface port-channel *p_range*** Port channel interfaces through which node connects.

Examples

- • This command displays the jobs running on the TaskTracker on the ***TaskTracker1*** node.

```
switch> show monitor hadoop tasktracker host TaskTracker1 counters

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId   Job Name          Cluster    Bytes In   Bytes Out   Start Time
------- ----------------- ---------- ---------- ----------- -------------------
2       ShortName2        Cluster0   37.36GB    76.29MB     2013-10-06 17:37:43
1       ReallyAVeryLon\   Cluster0   37.36GB    76.29MB     2013-10-06 17:56:03
        gNameForAJob1

Note: these counters are derived from Hadoop counters and represent approximate
```

```
network bandwidth utilization switch>
```

- This command displays jobs running on TaskTrackers accessed through Ethernet interfaces ***7*** and ***8***.

```
switch> show monitor hadoop tasktracker interface Ethernet 7,8 counters

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker3
JobId   Job Name          Cluster    Bytes In   Bytes Out   Start Time
------- ----------------- ---------- ---------- ----------- -------------------
510002  ShortName12       Cluster1   37.36GB    76.29MB     2013-10-06 17:37:43
510001  ReallyAVeryLon\   Cluster1   37.36GB    76.29MB     2013-10-06 17:56:03
        gNameForAJob11

Note: These counters are derived from Hadoop counters and represent approximate
network bandwidth utilization

Running job for TaskTracker: TaskTracker1
JobId   Job Name          Cluster    Bytes In   Bytes Out   Start Time
------- ----------------- ---------- ---------- ----------- -------------------
2       ShortName2        Cluster0   37.36GB    76.29MB     2013-10-06 17:37:43
1       ReallyAVeryLon\   Cluster0   37.36GB    76.29MB     2013-10-06 17:56:03
        gNameForAJob1

Note: these counters are derived from Hadoop counters and represent approximate
network bandwidth utilization switch>
```

- This command displays jobs running on TaskTrackers accessed through ***port channel interface 7***.

```
switch> show monitor hadoop tasktracker interface Port-Channel 7 counters

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker2
JobId   Job Name          Cluster    Bytes In   Bytes Out   Start Time
------- ----------------- ---------- ---------- ----------- -------------------
2       ShortName2        Cluster0   37.36GB    76.29MB     2013-10-06 17:37:43
1       ReallyAVeryLon\   Cluster0   37.36GB    76.29MB     2013-10-06 17:56:03
        gNameForAJob1

Note: these counters are derived from Hadoop counters and represent approximate
network bandwidth utilization switch>
```

20.7.4.25 show monitor hadoop tasktracker jobs

The **`show monitor hadoop tasktracker jobs`** command displays data about the jobs that are running on TaskTrackers located on the specified node or accessed through the listed interfaces.

Including a cluster parameter filters results to include data only from the cluster polled by the specified monitor.

Command Mode

EXEC

Command Syntax

`show monitor hadoop tasktracker` **NODES** jobs [**CLUSTERS**]

Parameters

- **NODES** TaskTracker node access point. Options include:
  - **host *hostname*** Node name.
  - **interface ethernet *e_range*** Ethernet interfaces through which node connects.
  - **interface port-channel *p_range*** Port channel interfaces through which node connects.
- **CLUSTERS** Hadoop cluster for which command displays data. Options include:
  - ***no parameter*** TaskTracker on specified **NODE** can be in any cluster.
  - **cluster *c_name*** TaskCluster on specified **NODE** must be in named cluster.

Examples

- This command displays data for jobs running on ***TaskTracker1***.

```
switch> show monitor hadoop tasktracker host TaskTracker1 jobs

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)  Reduces(#/%)   Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%       2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%   0/26.90%       2013-10-06 17:37:43 switch>
```

- This command displays data for jobs on TaskTrackers accessed through Ethernet interfaces ***7*** and ***8***.

```
switch> show monitor hadoop tasktracker interface Ethernet 7,8 jobs

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker3
JobId   Job Name         Cluster   Maps(#/%)  Reduces(#/%)  Start Time
------- ---------------- --------- ---------- ------------- -------------------
510001  ReallyAVeryLon\  Cluster1  2/12.34%   0/13.45%      2013-10-06 17:56:03
        gNameForAJob11
510002  ShortName12      Cluster1  2/24.68%   0/26.90%      2013-10-06 17:37:43

Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)  Reduces(#/%)   Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%       2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%   0/26.90%       2013-10-06 17:37:43 switch>
```

- This command displays data for jobs on TaskTrackers accessed through ***port channel interface 7***.

```
switch> show monitor hadoop tasktracker interface Port-Channel 7 jobs

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker2
JobId  Job Name         Cluster   Maps(#/%)  Reduces(#/%)   Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%       2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%   0/26.90%       2013-10-06 17:37:43 switch>
```

- This command displays data for jobs on TaskTrackers on the ***Cluster0*** cluster that are accessed through Ethernet interfaces ***7*** and ***8***.

```
switch> show monitor hadoop tasktracker interface Ethernet 7,8 jobs cluster Cluster0

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)  Reduces(#/%)   Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%       2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%   0/26.90%       2013-10-06 17:37:43 switch>
```

- This command displays data for jobs on TaskTracker named ***TaskTracker1*** on the ***Cluster0*** cluster.

```
switch> show monitor hadoop tasktracker host TaskTracker1 jobs cluster Cluster0

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)  Reduces(#/%)   Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%       2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%   0/26.90%       2013-10-06 17:37:43 switch>
```

- This command displays data for jobs on TaskTrackers on the ***Cluster0*** cluster that are accessed through ***port channel interface 7***.

```
switch> show monitor hadoop tasktracker interface Port-Channel 7 jobs cluster Cluster0

Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker2
```

```
JobId  Job Name         Cluster   Maps(#/%)  Reduces(#/%)   Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%       2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%   0/26.90%       2013-10-06 17:37:43 switch>
```

20.7.4.26 show monitor hadoop tasktracker running-tasks

The **show monitor hadoop tasktracker running-tasks** command displays progress and byte counts of tasks executed by TaskTrackers located on the specified node or accessed through the listed interfaces.

Including a cluster-ID parameter filters results to include data only from the specified cluster.

Command Mode

EXEC

Command Syntax

show monitor hadoop tasktracker **NODES** **running-tasks** [**CLUSTERS**][**JOBS**]

Parameters

- **NODES** TaskTracker node access point. Options include:
  - **host *hostname*** Node name.
  - **interface ethernet *e_range*** Ethernet interfaces through which node connects.
  - **interface port-channel *p_range*** Port channel interfaces through which node connects.
- **CLUSTERS** Hadoop cluster for which command displays data. Options include:
  - ***no parameter*** TaskTracker on specified **NODE** can be in any cluster.
  - **cluster *c_name*** TaskCluster on specified **NODE** must be in named cluster.
- **JOBS** Job list. Options include:
  - ***no parameter*** all jobs.
  - **job *0*** to ***2147483647*** Specifies number of single job.

Examples

- This command displays data for tasks running on TaskTracker named ***TaskTracker1***.

```
switch> show monitor hadoop tasktracker host TaskTracker1 running-tasks

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
2     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

- This command displays data for tasks running on the TaskTracker named ***TaskTracker1*** of the ***Cluster0*** cluster.

```
switch> show monitor hadoop tasktracker host TaskTracker1 running-tasks cluster
Cluster0

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
2     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

- This command displays data for tasks running for ***job 1*** on the TaskTracker named ***TaskTracker1*** of the ***Cluster0*** cluster.

```
switch> show monitor hadoop tasktracker host TaskTracker1 running-tasks cluster
Cluster0 job 1

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

- This command displays data for tasks running on TaskTrackers accessed through Ethernet interfaces ***7*** and ***8***.

```
switch> show monitor hadoop tasktracker interface Ethernet 7,8 running-tasks

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
2     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

Running tasks for TaskTracker: TaskTracker3 on interface Ethernet8
JobId  TaskId  Cluster  Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
------ ------ --------- ----- --------- -------- ---------- ----------- -------
510002 222    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510001 112    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510002 221    Cluster1  Map   50.00%    running  1.05MB     1.07MB      1.48MB
510001 111    Cluster1  Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

- This command displays data for tasks running on TaskTrackers of ***Cluster0*** cluster that are accessed through Ethernet interfaces ***7*** and ***8***.

```
switch> show monitor hadoop tasktracker interface Ethernet 7,8 running-tasks
cluster Cluster0

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2      Cluster0   Map   33.33%    running  2.10MB     2.14MB      2.96MB
2     2      Cluster0   Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1      Cluster0   Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1      Cluster0   Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

- This command displays data for tasks running for ***job 1*** on the TaskTrackers of ***Cluster0*** cluster that are accessed through Ethernet interfaces ***7*** and ***8***

```
switch> show monitor hadoop tasktracker interface Ethernet 7,8 running-tasks
cluster Cluster0 job 1

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

- This command displays data for tasks running on TaskTrackers accessed through ***port channe l*** interfaces ***7*** and ***8***.

```
switch> show monitor hadoop tasktracker interface Port-Channel 7-8 running-tasks

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker2 on interface Port-Channel7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
```

```
Running tasks for TaskTracker: TaskTracker4 on interface Port-Channel8
JobId  TaskId Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
------ ------ --------- ----- --------- -------- ---------- ----------- -------
510002 222    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510001 112    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510001 111    Cluster1  Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

- This command displays data for tasks running on TaskTrackers of ***Cluster0*** cluster accessed through ***port channel interface 7***.

```
switch> show monitor hadoop tasktracker interface Port-Channel 7 running-tasks
cluster Cluster0

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker2 on interface Port-Channel7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

- This command displays data for ***job 510001*** running on TaskTrackers of ***Cluster1*** cluster that are accessed through ***port channel interface 8***.

```
switch> show monitor hadoop tasktracker interface Port-Channel 8 running-tasks
cluster Cluster1 job 510001

Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker4 on interface Port-Channel8
JobId  TaskId Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
------ ------ --------- ----- --------- -------- ---------- ----------- -------
510001 112    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510001 111    Cluster1  Map   50.00%    running  1.05MB     1.07MB      1.48MB switch>
```

20.7.4.27 show monitor hadoop tasktracker running-tasks cluster job task

The **`show monitor hadoop tasktracker running-tasks cluster job task`** command displays detailed data for the specified task.

Command Mode

EXEC

Command Syntax

`show monitor hadoop tasktracker` **NODE** [running-tasks cluster ***name***][job ***jnum***][task ***tnum***]

Parameters

- **NODE** TaskTracker node access point. Options include:
  - **host *hostname*** Node name.
  - **interface ethernet *e_range*** Ethernet interfaces through which node connects.
  - **interface port-channel *p_range*** Port channel interfaces through which node connects.
- ***name*** Cluster name.
- ***jnum*** Job number. Value ranges from ***0*** to ***2147483647***.
- ***tnum*** Task number. Value ranges from ***0*** to ***2147483647***.

Examples

- This command displays data for ***task 1*** of ***job 1*** on ***TaskTracker1*** of ***Cluster0***.

```
switch> show monitor hadoop tasktracker host TaskTracker1 running-tasks
 cluster Cluster0 job 1 task 1

Last updated: 2013-10-06 18:14:23
Task details for one task as given below:
  TaskTracker name     : TaskTracker1
```

```
  Interface            : 'Ethernet7'
  Cluster              : Cluster0
  Job Id               : 1
  Task Id              : 1
  Attempt Id           : 0
  Task type            : Map
  Status               : running
  State                : running
  Start time           : 2013-10-06 17:57:43
  Progress             : 50.00%
  HDFS bytes read      : 1.05MB
  HDFS bytes written   : 1.07MB
  Reduce shuffle bytes : 1.48MB switch>
```

- This command displays data for ***task 1*** of ***job 1*** on the ***Cluster0*** TaskTracker that is accessible through ***Ethernet interface 7***.

```
switch> show monitor hadoop tasktracker interface Ethernet 7 running-tasks
 cluster Cluster0 job 1 task 1

Last updated: 2013-10-06 18:14:23
Task details for one task as given below:
  TaskTracker name     : TaskTracker1
  Interface            : 'Ethernet7'
  Cluster              : Cluster0
  Job Id               : 1
  Task Id              : 1
  Attempt Id           : 0
  Task type            : Map
  Status               : running
  State                : running
  Start time           : 2013-10-06 17:57:43
  Progress             : 50.00%
  HDFS bytes read      : 1.05MB
  HDFS bytes written   : 1.07MB
  Reduce shuffle bytes : 1.48MB switch>
```

- This command displays data for ***task 111*** of ***job 510001*** on the ***Cluster0*** TaskTracker that is accessible through ***port channel interface 8***.

```
switch> show monitor hadoop tasktracker interface Port-Channel 8 running-
tasks cluster Cluster1 job 510001 task 111

Last updated: 2013-10-06 18:14:23
Task details for one task as given below:
  TaskTracker name     : TaskTracker4
  Interface            : 'Port-Channel8'
  Cluster              : Cluster1
  Job Id               : 510001
  Task Id              : 111
  Attempt Id           : 0
  Task type            : Map
  Status               : running
  State                : running
  Start time           : 2013-10-06 17:57:43
  Progress             : 50.00%
  HDFS bytes read      : 1.05MB
  HDFS bytes written   : 1.07MB
  Reduce shuffle bytes : 1.48MB switch>
```

20.7.4.28 show monitor hadoop tasktracker status

The **show monitor hadoop tasktracker status** command displays connection and activity information for the TaskTracker on the specified clusters or accessed through the specified interface. The following command formats display the listed TaskTracker information:

- **show monitor hadoop cluster *c_name* tasktracker status**: TaskTrackers on specified cluster.
- **show monitor hadoop tasktracker *node* status**: TaskTrackers on specified nodes or interfaces.
- **show monitor hadoop tasktracker all status**: all connected TaskTrackers.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* tasktracker **status**

show monitor hadoop tasktracker NODES status

show monitor hadoop tasktracker all **status**

Parameters

- *c_name* Cluster name.
- **NODES** TaskTracker node access point. Options include:
  - **host *hostname*** Node name.
  - **interface ethernet *e_range*** Ethernet interfaces through which node connects.
  - **interface port-channel *p_range*** Port channel interfaces through which node connects.

Examples

- This command displays connection and activity information for all TaskTrackers connected through Ethernet interfaces ***7*** and ***8***.

```
switch> show monitor hadoop tasktracker interface Ethernet7,8 status

Last updated: 2013-10-06 18:14:23
  TaskTracker            : TaskTracker1
  IP Address             : 10.100.0.1
  Interface              : Ethernet7
  State                  : active
  Running jobs           : 2
  Running tasks          : 4
  Map Tasks              : 4
  Reduce Tasks           : 0
  Total bytes read       : 2.08GB
  Total bytes written    : 4.24MB
  TaskTracker            : TaskTracker3
  IP Address             : 10.100.0.3
  Interface              : Ethernet8
  State                  : active
  Running jobs           : 2
  Running tasks          : 4
  Map Tasks              : 4
  Reduce Tasks           : 0
  Total bytes read       : 6.23GB
  Total bytes written    : 12.72MB switch>
```

- This command displays connection and activity information for all connected TaskTrackers.

```
switch> show monitor hadoop tasktracker all status

Last updated: 2013-10-06 18:14:23
```

```
All local TaskTrackers:
  TaskTracker            : TaskTracker4
  IP Address             : 10.100.0.4
  Interface              : Port-Channel8
  State                  : active
  Running jobs           : 2
  Running tasks          : 4
  Map Tasks              : 4
  Reduce Tasks           : 0
  Total bytes read       : 8.30GB
  Total bytes written    : 16.95MB

  TaskTracker            : TaskTracker3
  IP Address             : 10.100.0.3
  Interface              : Ethernet8
  State                  : active
  Running jobs           : 2
  Running tasks          : 4
  Map Tasks              : 4
  Reduce Tasks           : 0
  Total bytes read       : 6.23GB
  Total bytes written    : 12.72MB

  TaskTracker            : TaskTracker2
  IP Address             : 10.100.0.2
  Interface              : Port-Channel7
  State                  : active
  Running jobs           : 2
  Running tasks          : 4
  Map Tasks              : 4
  Reduce Tasks           : 0
  Total bytes read       : 4.15GB
  Total bytes written    : 8.48MB

  TaskTracker            : TaskTracker1
  IP Address             : 10.100.0.1
  Interface              : Ethernet7
  State                  : active
  Running jobs           : 2
  Running tasks          : 4
  Map Tasks              : 4
  Reduce Tasks           : 0
  Total bytes read       : 2.08GB
  Total bytes written    : 4.24MB switch>
```

- This command displays connection and activity data for TaskTracker on the ***TaskTracker1*** node.

```
switch> show monitor hadoop tasktracker host TaskTracker1 status

Last updated: 2013-10-06 18:14:23
  TaskTracker            : TaskTracker1
  IP Address             : 10.100.0.1
  Interface              : Ethernet7
  State                  : active
  Running jobs           : 2
  Running tasks          : 4
  Map Tasks              : 4
  Reduce Tasks           : 0
  Total bytes read       : 2.08GB
  Total bytes written    : 4.24MB switch>
```

- This command displays connection and activity data for all TaskTracker connected through ***Port Channel 7***.

```
switch> show monitor hadoop tasktracker interface Port-Channel 7 status

Last updated: 2013-10-06 18:14:23
  TaskTracker            : TaskTracker2
  IP Address             : 10.100.0.2
  Interface              : Port-Channel7
  State                  : active
  Running jobs           : 2
  Running tasks          : 4
  Map Tasks              : 4
  Reduce Tasks           : 0
  Total bytes read       : 4.15GB
  Total bytes written    : 8.48MB switch>
```

20.7.4.29 show monitor hadoop traffic burst

The **show monitor hadoop traffic burst** command displays the largest data bursts for specified Hadoop cluster jobs. A data burst is the data consumed during a polling interval. The command displays input and output burst:

- Input bursts include bytes written to the host.
- Output bursts include bytes written by the host.

Command Mode

EXEC

Command Syntax

show monitor hadoop [**CLUSTERS**] **traffic burst** [**NODE**]

Parameters

- **CLUSTERS** Hadoop clusters for which command displays data. Options include:
  - ***no parameter*** All clusters.
  - **cluster *c_name*** Cluster name.
- **NODES** TaskTracker node access point. Options include:
  - **host *hostname*** Node name.
  - **interface ethernet *e_range*** Ethernet interfaces through which node connects.
  - **interface port-channel *p_range*** Port channel interfaces through which node connects.

Examples

- This command displays traffic burst data for all running jobs.

```
switch> show monitor hadoop traffic burst
Last updated: 2013-10-06 18:14:23
Bursts on Interface: 'Ethernet7' in cluster: Cluster0
Top 2 input bursts:
JobId  Job Name          Burst    Time
------ ----------------- -------- -------------------
1      ShortName          3.07GB  2013-10-06 17:57:43
2      ReallyAVeryLon\    6.15GB  2013-10-06 17:41:03
       gNameForAJob
Top 2 output bursts:
JobId  Job Name          Burst    Time
------ ----------------- -------- -------------------
1       ShortName          4.10GB  2013-10-06 17:55:13
2       ReallyAVeryLon\    8.20GB  2013-10-06 17:36:03
```

```
        gNameForAJob
Bursts on Interface: 'Port-Channel7' in cluster: Cluster0
Top 2 input bursts:
JobId  Job Name         Burst    Time
------ ---------------- -------- -------------------
1       ShortName        3.07GB  2013-10-06 17:57:43
2       ReallyAVeryLon\  6.15GB  2013-10-06 17:41:03
        gNameForAJob
Top 2 output bursts:
JobId  Job Name         Burst    Time
------ ---------------- -------- -------------------
1       ShortName        4.10GB  2013-10-06 17:55:13
2       ReallyAVeryLon\  8.20GB  2013-10-06 17:36:03
        gNameForAJob
Bursts on Interface: 'Ethernet8' in cluster: Cluster1
Top 4 input bursts:
JobId   Job Name         Burst    Time
------- ---------------- -------- -------------------
510001   ShortName        3.07GB  2013-10-06 17:57:43
510002   ReallyAVeryLon\  6.15GB  2013-10-06 17:41:03
         gNameForAJob
510003   ShortName        9.22GB   2013-10-06 17:24:23
510004   ReallyAVeryLon\  12.29GB  2013-10-06 17:07:43
         gNameForAJob
Top 4 output bursts:
JobId  Job Name         Burst    Time
------ ---------------- -------- -------------------
510001  ShortName        4.10GB  2013-10-06 17:55:13
510002  ReallyAVeryLon\   8.20GB  2013-10-06 17:36:03
        gNameForAJob
510003  ShortName        12.29GB  2013-10-06 17:16:53
510004  ReallyAVeryLon\   16.39GB  2013-10-06 16:57:43
        gNameForAJob
Bursts on Interface: 'Port-Channel8' in cluster: Cluster1
Top 4 input bursts:
JobId  Job Name         Burst     Time
------ ---------------- --------- -------------------
3101
510001  ShortName        3.07GB    2013-10-06 17:57:43
510002  ReallyAVeryLon\  6.15GB    2013-10-06 17:41:03
        gNameForAJob
510003  ShortName         9.22GB    2013-10-06 17:24:23
510004  ReallyAVeryLon\   12.29GB   2013-10-06 17:07:43
        gNameForAJob
Top 4 output bursts:
JobId  Job Name         Burst    Time
------ ---------------- -------- -------------------
510001  ShortName        4.10GB  2013-10-06 17:55:13
510002  ReallyAVeryLon\  8.20GB  2013-10-06 17:36:03
        gNameForAJob
510003  ShortName        12.29GB  2013-10-06 17:16:53
510004  ReallyAVeryLon\  16.39GB  2013-10-06 16:57:43
        gNameForAJob switch>
```

- This command displays traffic burst for all jobs running on TaskTrackers that are accessible through Ethernet interfaces ***7*** and ***8***.

```
switch> show monitor hadoop traffic burst interface Ethernet 7,8

Last updated: 2013-10-06 18:14:23
Bursts on Interface: 'Ethernet7' in cluster: Cluster0
Top 2 input bursts:
JobId  Job Name        Burst  Time
```

```
------ --------------- ------ -------------------
1      ShortName       3.07GB 2013-10-06 17:57:43
2      ReallyAVeryLon\ 6.15GB 2013-10-06 17:41:03
       gNameForAJob
Top 2 output bursts:
JobId  Job Name        Burst   Time
------ --------------- ------- -------------------
1      ShortName        4.10GB 2013-10-06 17:55:13
2      ReallyAVeryLon\  8.20GB 2013-10-06 17:36:03
       gNameForAJob
Bursts on Interface: 'Ethernet8' in cluster: Cluster1
Top 4 input bursts:
JobId  Job Name         Burst    Time
------ ---------------- -------- -------------------
510001  ShortName       3.07GB   2013-10-06 17:57:43
510002  ReallyAVeryLon\ 6.15GB   2013-10-06 17:41:03
        gNameForAJob
510003  ShortName        9.22GB   2013-10-06 17:24:23
510004  ReallyAVeryLon\  12.29GB  2013-10-06 17:07:43
        gNameForAJob
Top 4 output bursts:
JobId  Job Name        Burst    Time
------ --------------- -------- -------------------
510001 ShortName        4.10GB  2013-10-06 17:55:13
510002 ReallyAVeryLon\  8.20GB  2013-10-06 17:36:03
       gNameForAJob
510003 ShortName        12.29GB  2013-10-06 17:16:53
510004 ReallyAVeryLon\  16.39GB  2013-10-06 16:57:43
       gNameForAJob switch>
```

- This command displays traffic burst data for all running jobs that are accessible through ***port channel interface 7***.

```
switch> show monitor hadoop traffic burst interface Port-Channel 7

Last updated: 2013-10-06 18:14:23
Bursts on Interface: 'Port-Channel7' in cluster: Cluster0
Top 2 input bursts:
JobId  Job Name        Burst   Time
------ --------------- ------- -------------------
1      ShortName       3.07GB  2013-10-06 17:57:43
2      ReallyAVeryLon\ 6.15GB  2013-10-06 17:41:03
       gNameForAJob
Top 2 output bursts:
JobId  Job Name        Burst   Time
------ --------------- ------- -------------------
1      ShortName       4.10GB  2013-10-06 17:55:13
2      ReallyAVeryLon\ 8.20GB  2013-10-06 17:36:03
       gNameForAJob
```

20.7.4.30 show monitor system

The **show monitor system** command displays information about the memory usage status of the system including when low-memory situations have led to Out-Of-Memory (OOM) events.

Command Mode

EXEC

Command Syntax

show monitor system

Examples

- The following output is for a system where the feature is disabled and the system has never entered the low-memory mode.

```
Memory Exhaustion Feature enabled: False
System currently in Memory Exhaustion: False
Number of times system entered Memory Exhaustion: 0
```

- The following output is for a system where the feature is enabled and the system has never entered the low-memory mode.

```
Memory Exhaustion Feature enabled: True
System currently in Memory Exhaustion: False
Number of times system entered Memory Exhaustion: 0
```

- The following output is for a system where the feature is enabled and the system enters the low-memory mode.

```
Memory Exhaustion Feature enabled: True
System currently in Memory Exhaustion: True
Number of times system entered Memory Exhaustion: 1
Last time entered in Memory Exhaustion: 0:00:07 ago
```

- The following output is for a system where the feature is enabled and the system enters the low-memory mode once, and has exited the low-memory mode.

```
Memory Exhaustion Feature enabled: True
System currently in Memory Exhaustion: False
Last time entered in Memory Exhaustion: 0:01:00 ago
Last time exited from Memory Exhaustion: 0:00:05 ago
```

20.7.4.31 shutdown (Monitor-Hadoop)

The **shutdown** command globally disables MapReduce Tracer on the switch. Enabling MapReduce Tracer for an individual cluster requires the feature to be globally enabled through this command and enabled for the individual cluster thorough the **shutdown (Monitor Hadoop Cluster)** command. By default, MapReduce Tracer is globally disabled.

The **no shutdown** command globally enables MapReduce Tracer. The **shutdown** and **default shutdown** commands globally disable MapReduce Tracer by removing the corresponding **no shutdown** command from ***running-config***.

Command Mode

Monitor-hadoop Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Related Commands

The **monitor hadoop** command places the switch in the ***monitor-hadoop*** configuration mode.

Examples

- These commands globally enable MapReduce Tracer.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# no shutdown
switch(config-monitor-hadoop)# show active
```

```
 monitor hadoop
   no shutdown
switch(config-monitor-hadoop)#
```

- This command globally disables MapReduce Tracer.

```
switch(config-monitor-hadoop)# shutdown
switch(config-monitor-hadoop)# show active
switch(config-monitor-hadoop)#
```

20.7.4.32 shutdown (Monitor Hadoop Cluster)

The **shutdown** command disables MapReduce Tracer for the configuration mode cluster. Globally disabling MapReduce Tracer (**show monitor hadoop**) also disables the function on the individual cluster. Enabling MapReduce Tracer for the cluster requires the function to be enabled globally and for the individual cluster.

The **no shutdown** command configures the MapReduce Tracer setting as **enabled** for the configuration mode cluster. The **shutdown** and **default shutdown** commands disable MapReduce Tracer for the cluster by removing the corresponding **no shutdown** command from ***running-config***.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

shutdown

no shutdown

default shutdown

Related Commands

The **cluster (Monitor Hadoop)** command places the switch in the ***monitor-hadoop-cluster*** configuration mode.

Examples

- These commands globally enable MapReduce Tracer, then enables it for the ***CL2*** cluster.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# no shutdown
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# no shutdown
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
   cluster CL2
      no shutdown
switch(config-monitor-hadoop-CL2)# exit
switch(config-monitor-hadoop)# show active
 monitor hadoop
   no shutdown
   cluster CL2
      no shutdown
switch(config-monitor-hadoop)#
```

- These commands disable MapReduce Tracer for the ***CL2*** cluster. MapReduce Tracer remains globally enabled.

```
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# shutdown
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
```

```
   cluster CL2
switch(config-monitor-hadoop-CL2)# exit
switch(config-monitor-hadoop)# show active
 monitor hadoop
   no shutdown
   cluster CL2
switch(config-monitor-hadoop)#
```

20.7.4.33 tasktracker (Monitor Hadoop Cluster)

The **tasktracker** command specifies the HTTP port for accessing TaskTrackers of the Hadoop cluster monitored through configuration mode statements. The switch compiles a list of the cluster's TaskTracker addresses by periodically polling the cluster's JobTracker (**jobtracker (Monitor Hadoop Cluster)**). The default TaskTracker HTTP port is ***50060***.

The **no tasktracker** and **default tasktracker** commands restore the configuration mode TaskTracker HTTP port to ***50060*** by removing the corresponding **tasktracker** command from ***running-config***.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

tasktracker http-port ***port_number***

no tasktracker http-port

default tasktracker http-port

Parameters

port_num TaskTracker HTTP port number. Value ranges from ***1*** to ***65535***. Default value is ***50060***.

Related Command

The **cluster (Monitor Hadoop)** command places the switch in the ***monitor-hadoop-cluster*** configuration mode.

Examples

- These commands specify a TaskTracker HTTP port address of ***51000***.

```
switch(config)# monitor hadoop
switch(config-monitor-hadoop)# cluster CL2
switch(config-monitor-hadoop-CL2)# tasktracker http-port 51000
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
   cluster CL2
      tasktracker http-port 51000
switch(config-monitor-hadoop-CL2)#
```

- These commands restore the default TaskTracker HTTP port address of ***50060***.

```
switch(config-monitor-hadoop-CL2)# no tasktracker http-port
switch(config-monitor-hadoop-CL2)# show active
monitor hadoop
   cluster CL2
switch(config-monitor-hadoop-CL2)# show active all
monitor hadoop
   cluster CL2
      jobtracker rpc-port 8021
      tasktracker http-port 50060
      interval 10
```

```
      shutdown
switch(config-monitor-hadoop-CL2)#
```

20.8 Transceiver Performance Monitoring

- Transceiver Performance Monitoring Overview
- Configuring Transceiver Performance Monitoring
- Transceiver Performance Monitoring Limitations
- Transceiver Performance Monitoring Show Commands
- Transceiver Performance Monitoring Commands

20.8.1 Transceiver Performance Monitoring Overview

The Transceiver Performance Monitoring and Enhanced Diagnostics enables viewing of the Digital Optical Monitoring (DOM) parameters for the optics that support enhanced diagnostics using the CLI. The show commands are used to view the instantaneous values for various PAM4 parameters like Signal-To-Noise Ratio, Residual Inter Symbol Interference, PAM4 Level Transition Parameters, etc. that such optics support.

With Performance Monitoring feature EOS collects and maintains certain performance statistics over a user-defined time period. EOS stores data for two intervals, the current interval and the most recently completed interval. When the current interval completes, the data is saved as the previous interval so a new current interval may be started. The old previous interval data is discarded.

20.8.2 Configuring Transceiver Performance Monitoring

The Digital Optical Monitoring (DOM) is always enabled on all interfaces and can't be disabled via any configuration command.

The Performance Monitoring is disabled on all interfaces by default. The **`performance-monitoring transceiver default`** configuration command is used to enable Performance Monitoring on all interfaces across the system.

```
switch(config)# performance-monitoring transceiver default
```

The Performance Monitoring can also be enabled on a per interface basis using the **`performance-monitoring transceiver`** configuration command.

```
switch(config)# interface Ethernet8/1
switch(config-if-Et8/1)# performance-monitoring transceiver
```

The Performance Monitoring period can be configured using the **`performance-monitoring period <period>`** configuration command.

```
switch(config)# performance-monitoring period 60s
```

The above command configures the performance monitoring period to ***60*** seconds across the system. This command supports configuring the performance monitoring period in units of either seconds, minutes, hours or days. For example, use performance-monitoring period ***30*** minutes to configure the period as ***30*** minutes, performance-monitoring period ***2*** hours to configure the period as ***2*** hours and so on. The default performance-monitoring period is ***15*** minutes.

Use the **no** form of these commands to revert configuration.

20.8.3 Transceiver Performance Monitoring Limitations

100GBASE-FR, 100GBASE-LR and 100GBASE-DR do not have thresholds of enhanced Digital Optical Monitoring (DOM) parameters, so these parameters and their thresholds will not be displayed in the output of CLI command **`show interfaces transceiver dom thresholds`**.

```
switch# show interfaces Ethernet8/1 transceiver dom thresholds
Ch: Channel, mA: milliamperes, dBm: decibels (milliwatts),
C: Celsius, V: Volts, NA or N/A: not applicable.

Ethernet8/1
Last update: 0:00:02 ago
                                      High Alarm  High Warn   Low Warn  Low Alarm
Parameter             Ch      Value  Threshold  Threshold  Threshold  Threshold
 Unit    Indicator
-------------------------------------------------------------------------------
-----------------------------
Temperature            -      58.87      72.00      65.00       5.00      -2.00
 C
Voltage                -       3.28       3.56       3.47       3.14       3.04
 V
TX bias current        1      83.90     131.00     120.00      35.00      30.00
 mA
                       2      85.30     131.00     120.00      35.00      30.00
 mA
Optical TX power       1     -10.51      -6.99      -8.00     -11.00     -12.00
 dBm
                       2      -9.70      -6.99      -8.00     -11.00     -12.00
 dBm
Optical RX power       1      -0.99       6.99       6.02      -2.00      -3.01
 dBm
```

20.8.4 Transceiver Performance Monitoring Show Commands

- The **`show interfaces transceiver dom`** command is used to display the Digital Optical Monitoring (DOM) parameters from the CLI. The displayed parameters are the most recently read, **instantaneous** values.

```
switch# show interfaces Ethernet8/1 transceiver dom
Ch: Channel, N/A: not applicable, TX: transmit, RX: receive
mA: milliamperes, dBm: decibels (milliwatts), C: Celsius, V: Volts

Ethernet8/1
Last update: 0:00:03 ago
Parameter             Ch     Value Unit
-----------------------------------------
Temperature            -     58.97 C
Voltage                -      3.28 V
TX bias current        1     83.61 mA
                       2     85.15 mA
Optical TX power       1    -10.60 dBm
                       2     -9.79 dBm
Optical RX power       1     -0.84 dBm
                       2     -2.28 dBm
SNR                    1     17.46 dB
                       2     15.89 dB
Residual ISI           1      0.00 ps/nm
                       2      0.00 ps/nm
Level transitions      1    829.00
                       2   3055.00
TEC current error      1   -527.60 mA
                       2   -365.70 mA
Frequency error        1     -0.48 GHz
```

```
                           2       0.00 GHz
Pre-FEC BER                -   1.53e-03
Uncorrected BER            -   0.00e+00
```

- The **`show interfaces transceiver dom thresholds`** command is used to display the Digital Optical Monitoring (DOM) parameters and their thresholds from the CLI. The displayed parameters are the most recently read, **instantaneous** values.

```
switch# show interfaces Ethernet8/1 transceiver dom thresholds
Ch: Channel, mA: milliamperes, dBm: decibels (milliwatts),
C: Celsius, V: Volts, NA or N/A: not applicable.

Ethernet8/1
Last update: 0:00:02 ago
                                 High Alarm  High Warn   Low Warn  Low Alarm
Parameter             Ch      Value  Threshold  Threshold  Threshold  Threshold Unit
 Indicator
-------------------------------------------------------------------------------------
---
Temperature           -       58.87      72.00      65.00       5.00      -2.00 C
Voltage               -        3.28       3.56       3.47       3.14       3.04 V
TX bias current       1       83.90     131.00     120.00      35.00      30.00 mA
                      2       85.30     131.00     120.00      35.00      30.00 mA
Optical TX power      1      -10.51      -6.99      -8.00     -11.00     -12.00 dBm
                      2       -9.70      -6.99      -8.00     -11.00     -12.00 dBm
Optical RX power      1       -0.99       6.99       6.02      -2.00      -3.01 dBm
                      2       -2.45       6.99       6.02      -2.00      -3.01 dBm
 WARN
SNR                   1       17.46      40.00      39.00      19.00      18.00 dB
 ALARM
                      2       15.89      40.00      39.00      19.00      18.00 dB
 ALARM
Residual ISI          1        0.00     409.60     128.00    -128.00    -409.60 ps/nm
                      2        0.00     409.60     128.00    -128.00    -409.60 ps/nm
Level transitions     1      812.00   57344.00   53248.00       0.00       0.00
                      2     3249.00   57344.00   53248.00       0.00       0.00
TEC current error     1     -504.70     409.60     128.00    -128.00    -409.60 mA
 ALARM
                      2     -329.50     409.60     128.00    -128.00    -409.60 mA
 WARN
Frequency error       1       -0.42       5.12       2.56      -2.56      -5.12 GHz
                      2       -0.06       5.12       2.56      -2.56      -5.12 GHz
Pre-FEC BER           -    1.47e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
Uncorrected BER       -    0.00e+00   0.00e+00   0.00e+00        N/A        N/A
```

The Indicator column in the output above gives an indication of where the value lies with respect to its thresholds. Nothing is shown when the parameter is in the expected range between the Low Warn and High Warn thresholds. If the value is greater than or equal to the High Alarm or less than or equal to the Low Alarm, **`ALARM`** is shown. If the value is greater than or equal to the High Warn (but less than the High Alarm) or less than or equal to the Low Warn (but greater than the Low Alarm), **`WARN`** is shown.

- The **`show interfaces transceiver performance-monitoring`** command is used to display the performance monitoring statistics from the CLI. This command will display data only if the feature has been enabled using the configuration command performance-monitoring transceiver as described in the Configuration section.

```
switch# show interfaces Ethernet8/1 transceiver performance-monitoring
Ch: Channel, N/A: not applicable

Performance monitoring period : 0:15:00
Ethernet8/1
Parameter                                        Ch         Value
-----------------------------------------------------------------
Current Interval 0
  Started                                         -   0:00:12 ago
  Last update                                     -   0:00:02 ago
  Pre-FEC BER avg                                 -      1.41e-03
  Pre-FEC BER high alarm exceeded                 -             0
  Pre-FEC BER high warn exceeded                  -             0
```

```
  Pre-FEC BER max                              -     1.54e-03
  Pre-FEC BER min                              -     5.93e-06
  Temperature alarm exceeded                   -            0
  Temperature warn exceeded                    -            0
  Uncorrected BER avg                          -     0.00e+00
  Uncorrected BER high alarm exceeded          -            0
  Uncorrected BER high warn exceeded           -            0
  Uncorrected BER max                          -     0.00e+00
  Uncorrected BER min                          -     0.00e+00
Interval 1
  Started                                      -  0:15:13 ago
  Last update                                  -  0:00:12 ago
  Pre-FEC BER avg                              -     1.53e-03
  Pre-FEC BER high alarm exceeded              -            0
  Pre-FEC BER high warn exceeded               -            0
  Pre-FEC BER max                              -     1.74e-03
  Pre-FEC BER min                              -     3.11e-05
  Temperature alarm exceeded                   -            0
  Temperature warn exceeded                    -            0
  Uncorrected BER avg                          -     0.00e+00
  Uncorrected BER high alarm exceeded          -            0
  Uncorrected BER high warn exceeded           -            0
  Uncorrected BER max                          -     0.00e+00
  Uncorrected BER min                          -     0.00e+00
```

Performance Monitoring Period: 0:15:00

This indicates the length of one Performance Monitoring period. ***0:15:00*** specifies a period of ***15*** minutes, which is the default value for this parameter. The Performance Monitoring period can be configured using the **`performance-monitoring period`** configuration command as described in the Configuration section.

Started: 0:00:12 ago

This indicates the time the host started the performance monitoring interval.

Last Update: 0:00:02 ago

This indicates the last time the host read the diagnostic parameters from the module. The host samples values every six seconds.

Pre-FEC BER avg, Uncorrected BER avg

This is the average value so far in the indicated performance monitoring interval.

Pre-FEC BER min, Uncorrected BER min

This is the minimum value so far in the indicated performance monitoring interval.

Pre-FEC BER max, Uncorrected BER max

This is the maximum value so far in the indicated performance monitoring interval.

Pre-FEC BER high warn exceeded, Uncorrected BER high warn exceeded, Pre-FEC BER high alarm exceeded, Uncorrected BER high alarm exceeded

This is the number of times the host read a value which was greater than the indicated threshold. The host reads values and checks exceeded thresholds every six seconds. Please note that this provides only an indication of the actual exceeded count.

Temperature warn exceeded, Temperature alarm exceeded

This is the number of times the host read a value which was outside the range of the indicated threshold. The host reads values and checks exceeded thresholds every six seconds. Please note that this provides only an indication of the actual exceeded count.

- The **`show interfaces transceiver performance-monitoring thresholds`** command is used to display the performance monitoring statistics and their thresholds from the CLI. This command will display data only if the feature has been enabled using the configuration command performance-monitoring transceiver as described in the Configuration section.

```
switch# show interfaces Ethernet8/1 transceiver performance-monitoring thresholds
Ch: Channel, N/A: not applicable

Performance monitoring period : 0:00:30

Ethernet8/1
                                                          High Alarm  High Warn   Low Warn  Low Alarm
Parameter                                  Ch        Value  Threshold  Threshold  Threshold  Threshold
 Indicator
-------------------------------------------------------------------------------------------------------
---------
Current Interval 0
  Started                                   -  0:00:26 ago
  Last update                               -  0:00:03 ago
  Pre-FEC BER avg                           -     1.41e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
  Pre-FEC BER high alarm exceeded           -            0        N/A        N/A        N/A        N/A
  Pre-FEC BER high warn exceeded            -            0        N/A        N/A        N/A        N/A
  Pre-FEC BER max                           -     1.55e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
  Pre-FEC BER min                           -     5.93e-06   1.00e-04   1.00e-05        N/A        N/A
  Temperature alarm exceeded                -            0        N/A        N/A        N/A        N/A
  Temperature warn exceeded                 -            0        N/A        N/A        N/A        N/A
  Uncorrected BER avg                       -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
  Uncorrected BER high alarm exceeded       -            0        N/A        N/A        N/A        N/A
  Uncorrected BER high warn exceeded        -            0        N/A        N/A        N/A        N/A
  Uncorrected BER max                       -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
  Uncorrected BER min                       -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
Interval 1
  Started                                   -  0:00:57 ago
  Last update                               -  0:00:26 ago
  Pre-FEC BER avg                           -     1.53e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
  Pre-FEC BER high alarm exceeded           -            0        N/A        N/A        N/A        N/A
  Pre-FEC BER high warn exceeded            -            0        N/A        N/A        N/A        N/A
  Pre-FEC BER max                           -     1.74e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
  Pre-FEC BER min                           -     3.11e-05   1.00e-04   1.00e-05        N/A        N/A
  WARN
  Temperature alarm exceeded                -            0        N/A        N/A        N/A        N/A
  Temperature warn exceeded                 -            0        N/A        N/A        N/A        N/A
  Uncorrected BER avg                       -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
  Uncorrected BER high alarm exceeded       -            0        N/A        N/A        N/A        N/A
  Uncorrected BER high warn exceeded        -            0        N/A        N/A        N/A        N/A
  Uncorrected BER max                       -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
  Uncorrected BER min                       -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
```

20.8.5 Transceiver Performance Monitoring Commands

Global Configuration Commands

- performance-monitoring period
- performance-monitoring transceiver default

Interface Configuration Command

- performance-monitoring transceiver

Show Commands

- show interfaces transceiver dom
- show interfaces transceiver dom thresholds
- show interfaces transceiver performance-monitoring
- show interfaces transceiver performance-monitoring thresholds

20.8.5.1 performance-monitoring period

The **`performance-monitoring period`** command configures Performance Monitoring Period across the system.

The **no performance-monitoring period** command removes the Performance Monitoring Period configuration from the ***running-config***.

Command Mode

Global Configuration Mode

Command Syntax

performance-monitoring period ***period***

no performance-monitoring period ***period***

Parameter

period performance monitoring period in units of either seconds, minutes, hours or days. The default performance-monitoring period is ***15*** minutes.

Example

This command configures the performance monitoring period to ***60*** seconds across the system.

```
switch(config)# performance-monitoring period 60s
```

20.8.5.2 performance-monitoring transceiver

The **performance-monitoring transceiver** command enable Performance Monitoring on per interface basis.

The **no performance-monitoring transceiver** command removes the Performance Monitoring configuration from the ***running-config***.

Command Mode

Interface Configuration Mode

Command Syntax

performance-monitoring transceiver

no performance-monitoring transceiver

Example

This command enables the Performance Monitoring on a per interface basis.

```
switch(config)# interface Ethernet8/1
switch(config-if-Et8/1)# performance-monitoring transceiver
```

20.8.5.3 performance-monitoring transceiver default

The **performance-monitoring transceiver default** command enable Performance Monitoring on all interfaces across the system.

The **no performance-monitoring transceiver default** command removes the Performance Monitoring configuration from the ***running-config***.

Command Mode

Global Configuration Mode

Command Syntax

performance-monitoring transceiver default

no performance-monitoring transceiver default

Example

This command enables the Performance Monitoring on all interfaces across the system.

```
switch(config)# performance-monitoring transceiver default
```

20.8.5.4 show interfaces transceiver dom

The **show interfaces transceiver dom** command displays the Digital Optical Monitoring (DOM) parameters from the CLI. The displayed parameters are the most recently read, 'instantaneous' values.

Command Mode

Privileged EXEC

Command Syntax

show interfaces INTERFACE transceiver dom

Parameter

INTERFACE Ethernet interface name.

Example

This command displays the Digital Optical Monitoring (DOM) parameters.

```
switch# show interfaces Ethernet8/1 transceiver dom
Ch: Channel, N/A: not applicable, TX: transmit, RX: receive
mA: milliamperes, dBm: decibels (milliwatts), C: Celsius, V: Volts

Ethernet8/1
Last update: 0:00:03 ago
Parameter             Ch     Value Unit
-----------------------------------------
Temperature            -     58.97 C
Voltage                -      3.28 V
TX bias current        1     83.61 mA
                       2     85.15 mA
Optical TX power       1    -10.60 dBm
                       2     -9.79 dBm
Optical RX power       1     -0.84 dBm
                       2     -2.28 dBm
SNR                    1     17.46 dB
                       2     15.89 dB
Residual ISI           1      0.00 ps/nm
                       2      0.00 ps/nm
Level transitions      1    829.00
                       2   3055.00
TEC current error      1   -527.60 mA
                       2   -365.70 mA
Frequency error        1     -0.48 GHz
                       2      0.00 GHz
Pre-FEC BER            -  1.53e-03
Uncorrected BER        -  0.00e+00
```

20.8.5.5 show interfaces transceiver dom thresholds

The **show interfaces transceiver dom thresholds** command displays the Digital Optical Monitoring (DOM) parameters and their thresholds from the CLI.

Command Mode

Privileged EXEC

Command Syntax

`show interfaces` INTERFACE `transceiver dom thresholds`

Parameter

INTERFACE Ethernet interface name.

Example

This command displays the Digital Optical Monitoring (DOM) parameters and their thresholds.

```
switch# show interfaces Ethernet8/1 transceiver dom thresholds
Ch: Channel, mA: milliamperes, dBm: decibels (milliwatts),
C: Celsius, V: Volts, NA or N/A: not applicable.

Ethernet8/1
Last update: 0:00:02 ago
                                High Alarm  High Warn   Low Warn  Low Alarm
Parameter             Ch      Value  Threshold  Threshold  Threshold  Threshold Unit    Indicator
-------------------------------------------------------------------------------------------------
Temperature            -      58.87      72.00      65.00       5.00      -2.00 C
Voltage                -       3.28       3.56       3.47       3.14       3.04 V
TX bias current        1      83.90     131.00     120.00      35.00      30.00 mA
                       2      85.30     131.00     120.00      35.00      30.00 mA
Optical TX power       1     -10.51      -6.99      -8.00     -11.00     -12.00 dBm
                       2      -9.70      -6.99      -8.00     -11.00     -12.00 dBm
Optical RX power       1      -0.99       6.99       6.02      -2.00      -3.01 dBm
                       2      -2.45       6.99       6.02      -2.00      -3.01 dBm          WARN
SNR                    1      17.46      40.00      39.00      19.00      18.00 dB           ALARM
                       2      15.89      40.00      39.00      19.00      18.00 dB           ALARM
Residual ISI           1       0.00     409.60     128.00    -128.00    -409.60 ps/nm
                       2       0.00     409.60     128.00    -128.00    -409.60 ps/nm
Level transitions      1     812.00   57344.00   53248.00       0.00       0.00
                       2    3249.00   57344.00   53248.00       0.00       0.00
TEC current error      1    -504.70     409.60     128.00    -128.00    -409.60 mA           ALARM
                       2    -329.50     409.60     128.00    -128.00    -409.60 mA            WARN
Frequency error        1      -0.42       5.12       2.56      -2.56      -5.12 GHz
                       2      -0.06       5.12       2.56      -2.56      -5.12 GHz
Pre-FEC BER            -   1.47e-03   1.00e-04   1.00e-05        N/A        N/A              ALARM
Uncorrected BER        -   0.00e+00   0.00e+00   0.00e+00        N/A        N/A
```

Command Output

The Indicator column in the output above gives an indication of where the value lies with respect to its thresholds. Nothing is shown when the parameter is in the expected range between the Low Warn and High Warn thresholds. If the value is greater than or equal to the High Alarm or less than or equal to the Low Alarm, **`ALARM`** is shown. If the value is greater than or equal to the High Warn (but less than the High Alarm) or less than or equal to the Low Warn (but greater than the Low Alarm), **`WARN`** is shown.

20.8.5.6 show interfaces transceiver performance-monitoring

The **`show interfaces transceiver performance-monitoring`** command displays the performance monitoring statistics from the CLI. This command will display data only if the Performance Monitoring has been enabled using the configuration command **`performance-monitoring transceiver`**.

Command Mode

Privileged EXEC

Command Syntax

`show interfaces` INTERFACE `transceiver performance-monitoring`

Parameter

INTERFACE Ethernet interface name.

Example

This command displays the performance monitoring statistics from the CLI.

```
switch# show interfaces Ethernet8/1 transceiver performance-monitoring
Ch: Channel, N/A: not applicable
```

```
Performance monitoring period : 0:15:00
Ethernet8/1
Parameter                                          Ch        Value
-------------------------------------------------------------------
Current Interval 0
  Started                                           -  0:00:12 ago
  Last update                                       -  0:00:02 ago
  Pre-FEC BER avg                                   -     1.41e-03
  Pre-FEC BER high alarm exceeded                   -            0
  Pre-FEC BER high warn exceeded                    -            0
  Pre-FEC BER max                                   -     1.54e-03
  Pre-FEC BER min                                   -     5.93e-06
  Temperature alarm exceeded                        -            0
  Temperature warn exceeded                         -            0
  Uncorrected BER avg                               -     0.00e+00
  Uncorrected BER high alarm exceeded               -            0
  Uncorrected BER high warn exceeded                -            0
  Uncorrected BER max                               -     0.00e+00
  Uncorrected BER min                               -     0.00e+00
Interval 1
  Started                                           -  0:15:13 ago
  Last update                                       -  0:00:12 ago
  Pre-FEC BER avg                                   -     1.53e-03
  Pre-FEC BER high alarm exceeded                   -            0
  Pre-FEC BER high warn exceeded                    -            0
  Pre-FEC BER max                                   -     1.74e-03
  Pre-FEC BER min                                   -     3.11e-05
  Temperature alarm exceeded                        -            0
  Temperature warn exceeded                         -            0
  Uncorrected BER avg                               -     0.00e+00
  Uncorrected BER high alarm exceeded               -            0
  Uncorrected BER high warn exceeded                -            0
  Uncorrected BER max                               -     0.00e+00
  Uncorrected BER min                               -     0.00e+00
```

Command Output

Performance Monitoring Period: 0:15:00

This indicates the length of one Performance Monitoring period. ***0:15:00*** specifies a period of ***15*** minutes, which is the default value for this parameter. The Performance Monitoring period can be configured using the **`performance-monitoring period configuration`** command as described in the Configuration section.

Started: 0:00:12 ago

This indicates the time the host started the performance monitoring interval.

Last Update: 0:00:02 ago

This indicates the last time the host read the diagnostic parameters from the module. The host samples values every six seconds.

Pre-FEC BER avg, Uncorrected BER avg

This is the average value so far in the indicated performance monitoring interval.

Pre-FEC BER min, Uncorrected BER min

This is the minimum value so far in the indicated performance monitoring interval.

Pre-FEC BER max, Uncorrected BER max

This is the maximum value so far in the indicated performance monitoring interval.

Pre-FEC BER high warn exceeded, Uncorrected BER high warn exceeded, Pre-FEC BER high alarm exceeded, Uncorrected BER high alarm exceeded

This is the number of times the host read a value which was greater than the indicated threshold. The host reads values and checks exceeded thresholds every ***6*** seconds. Please note that this provides only an indication of the actual exceeded count.

Temperature warn exceeded, Temperature alarm exceeded

This is the number of times the host read a value which was outside the range of the indicated threshold. The host reads values and checks exceeded thresholds every six seconds. Note that this provides only an indication of the actual exceeded count.

20.8.5.7 show interfaces transceiver performance-monitoring thresholds

The **`show interfaces transceiver performance-monitoring thresholds`** command displays the performance monitoring statistics and their thresholds from the CLI.

Command Mode

Privileged EXEC

Command Syntax

`show interfaces` **INTERFACE** **`transceiver performance-monitoring thresholds`**

Parameter

INTERFACE Ethernet interface name.

Example

This command displays the performance monitoring statistics and their thresholds from the CLI.

```
switch# show interfaces Ethernet8/1 transceiver performance-monitoring thresholds
Ch: Channel, N/A: not applicable

Performance monitoring period : 0:00:30

Ethernet8/1
                                                          High Alarm  High Warn   Low Warn  Low Alarm
Parameter                                    Ch        Value  Threshold  Threshold  Threshold  Threshold
 Indicator
-------------------------------------------------------------------------------------------------------
---------
Current Interval 0
  Started                                     -  0:00:26 ago
  Last update                                 -  0:00:03 ago
  Pre-FEC BER avg                             -     1.41e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
  Pre-FEC BER high alarm exceeded             -            0        N/A        N/A        N/A        N/A
  Pre-FEC BER high warn exceeded              -            0        N/A        N/A        N/A        N/A
  Pre-FEC BER max                             -     1.55e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
  Pre-FEC BER min                             -     5.93e-06   1.00e-04   1.00e-05        N/A        N/A
  Temperature alarm exceeded                  -            0        N/A        N/A        N/A        N/A
  Temperature warn exceeded                   -            0        N/A        N/A        N/A        N/A
  Uncorrected BER avg                         -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
  Uncorrected BER high alarm exceeded         -            0        N/A        N/A        N/A        N/A
  Uncorrected BER high warn exceeded          -            0        N/A        N/A        N/A        N/A
  Uncorrected BER max                         -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
  Uncorrected BER min                         -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
Interval 1
  Started                                     -  0:00:57 ago
  Last update                                 -  0:00:26 ago
  Pre-FEC BER avg                             -     1.53e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
  Pre-FEC BER high alarm exceeded             -            0        N/A        N/A        N/A        N/A
  Pre-FEC BER high warn exceeded              -            0        N/A        N/A        N/A        N/A
  Pre-FEC BER max                             -     1.74e-03   1.00e-04   1.00e-05        N/A        N/A
 ALARM
  Pre-FEC BER min                             -     3.11e-05   1.00e-04   1.00e-05        N/A        N/A
 WARN
  Temperature alarm exceeded                  -            0        N/A        N/A        N/A        N/A
  Temperature warn exceeded                   -            0        N/A        N/A        N/A        N/A
  Uncorrected BER avg                         -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
  Uncorrected BER high alarm exceeded         -            0        N/A        N/A        N/A        N/A
  Uncorrected BER high warn exceeded          -            0        N/A        N/A        N/A        N/A
  Uncorrected BER max                         -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
  Uncorrected BER min                         -     0.00e+00   0.00e+00   0.00e+00        N/A        N/A
```

20.9 AVA Sensor

AVA switch sensor also known as “monitor security awake” feature provides deep network analysis by doing deep packet inspection of some or all packets of traffic that's forwarded by the switch. It continuously monitors enterprise devices, users, and applications wherever they are, even as IP addresses change while maintaining a forensic record of past activities. This functionality can be enabled or disabled on the fly without impacting regular packet forwarding functionality.

AVA Switch Sensor Extension

AVA switch sensor extends EOS telemetry for network threat detection. The FlowWatcher agent RPMs along with other RPMs that enable AVA switch sensor functionality are released as part of the **`NDRSensor.swix`** extension. The **`NDRSensor.swix`** extension needs to be installed on the system running supported EOS version. After the extension is installed, it can be enabled using **`monitor security awake`** command as described in the Configuration section.

When **`monitor security awake`** is enabled, the FlowWatcher agent enables hardware flow tracking with filter based packet sampling, IP/IPv6 access lists and mirroring features to receive the first few packets of every new flow and flow volume using IPFIX. It connects to the AVA Nucleus using SSL. The EOS FlowWatcher agent processes the most interesting packets of all the flows that are forwarded by the switch. It does deep packet inspection (DPI) of the packets, computes flow volume using IPFIX data records and generates activity records. The generated activity records are then sent to AVA Nucleus using Kafka over a TLS session. The AVA Nucleus uses a combination of detection models to uncover malicious intent to provide NDR functionality.

Platform Compatibility

- CCS-720XP-96ZC2
- CCS-720XP-48Y6
- CCS-720XP-48ZC2
- CCS-720XP-24ZY4
- CCS-720XP-24Y6
- CCS-720DP-48ZS

- NDR Sensor Extension Installation
- Configuration
- Upgrade EOS and/or NDRSensor.swix Extension
- Show Commands
- Limitations
- AVA Sensor Commands

20.9.1 NDR Sensor Extension Installation

If the switch already has the **`NDRSensor.swix`** installed, in order to upgrade EOS image or extension follow the steps to uninstall extension before proceeding to install new extension. After the switch is up with supported EOS.swi, copy and install the **`NDRSensor.swix`** extension.

```
switch# copy <source>/NDRSensor.swix extension:
switch# extension NDRSensor.swix
```

To show the status of extensions use the following command.

```
switch# show extensions
Name                               Version/Release      Status      Extension
---------------------------------- -------------------- ----------- ---------
NDRSensor.swix                     4.30.1F/316201\      A, NI, B    8
                                   29.4301F
```

Optionally, copy **NDRSensor.swix** to boot-extensions.

```
switch# copy installed-extensions boot-extensions
Copy completed successfully.
```

Refer to Managing EOS Extensions for additional details on managing extensions on EOS.

20.9.2 Configuration

20.9.2.1 SSL Profile Configuration

To configure an SSL profile for communication with the AVA Nucleus, use the following CLIs:

```
switch(config)# management security
switch(config-mgmt-security)# ssl profile <profile-name>
switch(config-mgmt-sec-ssl-profile-awake-nucleus1)# certificate <certificate-
name> key <key-name>
switch(config-mgmt-sec-ssl-profile-awake-nucleus1)# trust certificate system
```

The AVA Nucleus certificate can either be self-signed or signed by a third-party. For the self-signed case, the certificate needs to be copied onto the switch **certificate:** directory and configured using **trust certificate <certificate>** CLI under the SSL **profile** configuration. In case it's signed by a third-party, then the CA certificate of the third-party needs to be copied onto the switch **certificate:** directory and configured using **trust certificate <certificate>**. If the third-party is a common CA and present in the system-supplied list of trusted CAs (most cases), then the **trust certificate system** can be used, and no certificate needs to be copied over to the switch.

Refer to **SSL certificate and key management** for additional details on all SSL profile configuration parameters. For instructions on how to manage certificates on the switch using the CLI refer to: **Working with certificates**.

Generating SSL Key and Self-signed Certificate

Use these steps to setup senor and Nucleus SSL connection with self signed certificates. For complete details on EOS TLS security refer to **EOS user manual control plane security section**.

- To create an SSL key on the switch:

  ```
  switch# security pki key generate rsa 2048 <key-name>
  ```

- To show the SSL key generated:

  ```
  switch# show management security ssl key <key-name>
  ```

- To generate a self signed certificate on the switch use the following CLI.

  ```
  switch# security pki certificate generate self-signed <certificate-name> key
   <key-name> generate rsa 2048 [parameters … ]
  ```

  Enter the certificate parameters interactively or use **parameters** option to specify them inline. Once SSL key and certificate are generated, they can be used in the **ssl profile**.

- The generated certificate needs to be copied to AVA Nucleus using either **more** or use **copy** command on the switch CLI.

```
switch# more certificate:<certificate-name>  --> Will emit certificate on
 the switch console
```

```
Arista# copy certificate:<certificate-name> <destination> --> To copy
 certificate to external location
```

20.9.2.2 Monitor Security Awake Configuration

- To configure the feature, enter **monitor security awake** mode:

```
switch(config)# [no|default] monitor security awake
```

- To enable Monitor security awake:

```
switch(config-monitor-security-awake)# [no|default] disabled
```

- To configure the maximum flow table size:

```
switch(config-monitor-security-awake)# [no|default] flow table size <SIZE>
 entries
! The flow table size configuration change will cause the FlowWatcher agent
 restart and all active flows to be lost.
Do you wish to proceed with this command? [y/N]
```

The default flow table size is 16k entries. Flow table size configuration change will restart the FlowWatcher agent.

- To configure Kafka topic name:

```
switch(config-monitor-security-awake)# [no|default] topic <name>
```

The default topic name is: **packet-analysis-sessions**

- To configure monitor-point identifier (MPID) to identify campus/site:

```
switch(config-monitor-security-awake)# [no|default] monitor-point identifier
 <id>
```

Every AVA sensor needs to be configured with a unique monitor-point identifier and provisioned in the Nucleus. The default monitor point id used is: 0.

20.9.2.3 Nucleus Configuration

- To configure the Nucleus enter nucleus mode:

```
switch(config-monitor-security-awake)# [no|default] nucleus <name>
```

- To configure Nucleus IP and port:

```
switch(config-monitor-security-awake-nucleus)# [no|default] destination
 <ipv4 address>/<ipv6 address>/<dns> [port <port>]
```

The default port is: 9094

- To configure the local interface for source IP, VRF information:

```
switch(config-monitor-security-awake-nucleus-<name>)# [no|default] local
 interface <local-interface>
```

- To configure SSL profile:

```
switch(config-monitor-security-awake-nucleus-<name>)# [no|default] ssl
 profile <profile-name>
```

20.9.2.4 Loopback Interface Configuration

If at least one Nucleus doesn't have a local interface configured, then ***Loopback0*** interfaces need to be configured with an IP address. ***Loopback0*** interface will be used by Hardware flow tracking as a local interface for sending IPFIX records to CPU.

```
switch(config)# interface Loopback0
switch(config-if-Lo0)# ip address <ip>[/n] [ip subnet mask]
```

20.9.2.5 Configuration Examples

Example configuration sequence:

```
switch# Generate SSL client key
switch# security pki key generate rsa 2048 client-key
switch# show management security ssl key client-key
…
switch# Generate SSL client certificate
switch# security pki certificate generate self-signed client-cert key client-
key generate rsa 2048 parameters common-name SwitchName country US state CA
 locality "Santa Clara" organization Arista organization-unit IT
certificate:client-cert generated
switch# show management security ssl certificate client-cert
…
switch# Copy SSL client certificate to AVA nucleus
switch# copy certificate:client-cert <destination>

switch# Copy AVA nucleus certificate onto Switch
switch# copy terminal: certificate:ca-cert
enter input line by line; when done enter one or more control-d
….
Copy completed successfully.
switch# show management security ssl certificate ca-cert
…

# Create SSL profile with client-key, client-cert and ca-cert
switch(config)# management security
switch(config-mgmt-security)# ssl profile awake-nucleus
switch(config-mgmt-sec-ssl-profile-awake-nucleus)# certificate client-cert key
 client-key
switch(config-mgmt-sec-ssl-profile-awake-nucleus)# trust certificate ca-cert
switch(config-mgmt-sec-ssl-profile-awake-nucleus)# end

# Enable IP routing
switch# conf
switch(config)# ip routing

# Configure monitor security awake and enable
switch(config)# monitor security awake
```

```
switch(config-monitor-security-awake)# monitor-point identifier 1
switch(config-monitor-security-awake)# nucleus awake-nucleus
switch(config-monitor-security-awake-nucleus-awake-nucleus)# local interface
 Management 1
switch(config-monitor-security-awake-nucleus-awake-nucleus)# destination
 nucleus1.foo.com
switch(config-monitor-security-awake-nucleus-awake-nucleus)# ssl profile awake-
nucleus
switch(config-monitor-security-awake-nucleus-awake-nucleus)# exit
switch(config-monitor-security-awake)# no disabled
switch(config-monitor-security-awake)# end
```

Running configuration.

```
…
management security
   ssl profile awake-nucleus
      certificate client-cert key client-key
      trust certificate ca-cert
…
ip routing
…
monitor security awake
   nucleus awake-nucleus
      local interface Management1
      destination nucleus1.foo.com
      ssl profile awake-nucleus
   no disabled
```

Sample configuration - with multiple profiles and Nucleus:

```
management security
   ssl profile awake-nucleus1
      certificate client-cert key client-key1
      trust certificate system
   ssl profile awake-nucleus2
      certificate client-cert key client-key2
      trust certificate root-cert2

monitor security awake
   topic packet-analysis-sessions
   monitor-point identifier 10
   flow table size 153600 entries
   !
   nucleus nucleus1
      local interface Loopback10
      destination nucleus1.foo.com
      ssl profile awake-nucleus1
   nucleus nucleus2
      local interface Management1
      destination nucleus2.foo.com
      ssl profile awake-nucleus2
   no disabled
```

20.9.3 Upgrade EOS and/or NDRSensor.swix Extension

1. Disable AVA sensor by configuring **`disabled`** under **`monitor security awake`**. This will stop the AVA switch sensor running on the switch.
2. Uninstall **`NDRSensor.swix`** extension.

```
switch# show installed-extensions
```

```
NDRSensor.swix

switch# no extension NDRSensor.swix
switch# show extensions
Name                                Version/Release      Status      Extension
----------------------------------- -------------------- ----------- ---------
NDRSensor.swix                      4.30.1F/316201\      A, NI, B    8
                                    29.4301F
```

3. Remove **`NDRSensor.swix`** from boot extensions.

```
switch# show boot-extensions
NDRSensor.swix # Old extension needs to be removed from boot-extension

switch# copy installed-extensions boot-extensions
Copy completed successfully.

switch# show boot
# boot-extensions empty.
```

4. Upgrade EOS image on the switch - Follow the steps to upgrade the EOS image on the box as described in the EOS configuration guide. Once the switch is back up and running.
5. Install new **`NDRSensor.swix`** extension corresponding to the new EOS image and copy it to boot-extensions. This step is same as NDR Sensor Extension Installation
6. Verify the AVA switch sensor is running by using the **`show monitor security awake`** command.

20.9.4 Show Commands

- Following CLI command shows the status of the Monitor security Awake feature:

```
switch# show monitor security awake [nucleus <nucleus>]
```

Example

```
switch# show monitor security awake
Monitor security awake status: active
Topic identifier: packet-analysis-sessions
Monitor point identifier: 0
Flow table size: 16384 entries
Flow table inactive timeout: 40.0 seconds
Active interfaces: Et37,39,49

Nucleus: nucleus1
Status: connected
VRF: default
Local interface: Management1 (172.28.134.144)
Destination: 10.243.93.139 port 9094
SSL profile: awake-nucleus1
Last established: 2 days, 19:58:23 ago
```

- Following CLI command shows the various counters of the Monitor security awake feature:

```
switch# show monitor security awake counters [flows|ipfix|nucleus
 [<nucleus>]]
```

Example

```
switch# show monitor security awake counters
Active flows: 269, RX packets: 8.943M (8943077)
Flows created: 558.655K (558655), expired: 558.386K (558386)
```

```
IPv4 flows:
Application       Flows Active       Flows Created           Flows Expired
----------------- ------------------ ----------------------- -------------
----
DHCP              9                  13.072K (13072)         13.063K (13063)
DNS               53                 279.857K (279857)       279.804K
 (279804)
HTTP              5                  34.759K (34759)         34.754K (34754)
LDAP              0                  15                      15
NetBios           0                  18                      18
SMB               0                  18                      18
TLS/SSL           45                 143.986K (143986)       143.941K
 (143941)
Other             157                86.93K (86930)          86.773K (86773)

IPv6 flows:
Application       Flows Active       Flows Created       Flows Expired
----------------- ------------------ ------------------- -------------

Nucleus: nucleus1
Activity records sent: 578.306K (578306), last sent 0:00:00 ago
Progress records sent: 172.057K (172057), last sent 0:00:00 ago
Last successful connection: 10:11:22 ago
Successful connections: 11
Last connection failure: 10:11:51 ago
Connection failures: 46
Activity records in queue: 0
Progress records in queue: 1

IPFIX counters:
Exporter: 172.22.197.142 Source port: 36582 Observation domain ID: 1
Messages received: 379.455K (379455)
Template records received: 208
Options template records received: 0
Data records received: 7.308M (7308225)
Options data records received: 0
Unknown template ID errors: 0
Invalid IPFIX messages received: 0
Flow record queue full: 0
```

- Following CLI command shows the flow table snapshot of all the flows currently active in the FlowWatcher agent:

```
switch# switch# show monitor security awake flow-table [detail]
```

This is debugging CLI and can take a while to dump the snapshot of the current flow table.

Example 1:

```
switch# show monitor security awake flow-table
IPv4 flows: 5
Lower IP address     Higher IP address     Protocol                  Start
 Time  Packets      Bytes
-------------------- --------------------- --------- -------------
-------------- -------- ---------
10.8.3.102:58119     87.98.179.150:6893    UDP       2022-03-09
 21:54:00.447260        1         71
10.3.1.31:58554      10.5.10.2:1999        UDP       2022-03-09
 21:54:00.493157       11       1358
10.3.1.61:34703      10.5.7.2:1999         UDP       2022-03-09
 21:54:11.586934       29      11460
10.3.1.63:52498      10.5.6.2:1999         UDP       2022-03-09
 21:54:18.397906        9        637
```

```
10.3.1.81:47766      10.5.13.2:1999        UDP       2022-03-09
 21:54:00.411780       25      12606

IPv6 flows: 2
Lower IP address                  Higher IP address Protocol
 Start Time Packets Bytes
-------------------------------- ----------------- -------- -------------
------------ ------- -----
[fe80::10fd:3ded:b992:b0fe]:5353 [ff02::fb]:5353   UDP      2022-03-09
 22:15:55.765183      14  3020
[fe80::186d:bd78:4904:679f]:5353 [ff02::fb]:5353   UDP      2022-03-09
 22:15:55.765136      11  2647
```

Example 2:

```
switch# show monitor security awake flow-table detail
Flow table detail codes: L2H - Lower to higher IP address, H2L - Higher to
 lower IP address

IPv4 Flows: 2
Flow: UDP 10.8.4.103:61591 - 87.98.179.248:6893
Start time: 2022-03-14 17:58:03.390968, Last packet time: 2022-03-14
 17:58:10.874020
Packets L2H: 0, Bytes L2H: 0, Packets H2L: 1, Bytes H2L: 64

Flow: UDP 10.3.1.63:60835 - 10.5.12.2:1999
Start time: 2022-03-14 17:58:39.406626, Last packet time: 2022-03-14
 17:58:40.207204
Packets L2H: 0, Bytes L2H: 0, Packets H2L: 35, Bytes H2L: 13692

IPv6 Flows: 2
Flow: UDP [fe80::7854:510f:c685:ff22]:57938 - [ff02::c]:3702
Start time: 2022-03-14 17:58:15.818269, Last packet time: 2022-03-14
 17:58:15.818269
Packets L2H: 0, Bytes L2H: 0, Packets H2L: 1, Bytes H2L: 722

Flow: UDP [fe80::d065:deb9:d239:bed3]:61403 - [ff02::1:3]:5355
Start time: 2022-03-14 17:58:21.011325, Last packet time: 2022-03-14
 17:58:21.011325
Packets L2H: 0, Bytes L2H: 0, Packets H2L: 2, Bytes H2L: 180
```

20.9.5 Limitations

- **`NDRSensor.swix`** is supported on the 32-bit EOS version only.
- In an MLAG setup, AVA switch sensor doesn't correlate bidirectional flows that are hashed to different MLAG peers.
- AVA switch sensor doesn't correlate bidirectional flows that are NATed.
- When AVA switch sensor is enabled, the TerminAttr agent will also process IPFIX packets sent to the CPU, and the Traffic flows feature on the CloudVision portal is automatically enabled and can cause additional CPU usage on the switch. If "Traffic flows" feature is not required on ClouldVision, disable TerminAttr IPFIX processing by adding **`-ipfix=false`** argument to **`exec /usr/bin/TerminAttr`** under **`daemon TerminAttr`** configuration.

20.9.6 AVA Sensor Commands

- monitor security awake
- show monitor security awake
- show monitor security awake counters

20.9.6.1 monitor security awake

The **monitor security awake** command when enabled the FlowWatcher agent enables Hardware flow tracking with filter based packet sampling, IP/IPv6 access lists and Mirroring features to receive the first few packets of every new flow and flow volume using IPFIX.

The **no** form of the **monitor security awake** command disables the monitor security awake feature from the **running config**.



Note: Before enabling **monitor security awake** command the **NDRSensor.swix** extension needs to be installed on the system running supported EOS version.

Command Mode

Global Configuration

Command Syntax

monitor security awake

no monitor security awake

default monitor security awake

Example

These commands create and enable a non-persistent DirectFlow flow.

```
switch(config)# monitor security awake
switch(config-monitor-security-awake)#
```

20.9.6.2 show monitor security awake

The **show monitor security awake** command displays the status of the Monitor security Awake feature on the switch.

Command Mode

EXEC

Command Syntax

show monitor security awake [nucleus ***nucleus***]

Example

This command displays the status of the Monitor security Awake feature on the switch..

```
switch# show monitor security awake
Monitor security awake status: active
Topic identifier: packet-analysis-sessions
Monitor point identifier: 0
Flow table size: 16384 entries
Flow table inactive timeout: 40.0 seconds
Active interfaces: Et37,39,49

Nucleus: nucleus1
Status: connected
VRF: default
Local interface: Management1 (172.28.134.144)
Destination: 10.243.93.139 port 9094
SSL profile: awake-nucleus1
Last established: 2 days, 19:58:23 ago
```

20.9.6.3 show monitor security awake counters

The **`show monitor security awake counters`** command displays the various counters of the Monitor security awake feature on the switch.

Command Mode

EXEC

Command Syntax

`show monitor security awake counters` [flows | ipfix | nucleus [***nucleus***<>]]

Example

This command displays the various counters of the Monitor security awake feature on the switch.

```
switch# show monitor security awake counters
Active flows: 269, RX packets: 8.943M (8943077)
Flows created: 558.655K (558655), expired: 558.386K (558386)

IPv4 flows:
Application       Flows Active       Flows Created           Flows Expired
----------------- ------------------ ----------------------- -----------------
DHCP              9                  13.072K (13072)         13.063K (13063)
DNS               53                 279.857K (279857)       279.804K (279804)
HTTP              5                  34.759K (34759)         34.754K (34754)
LDAP              0                  15                      15
NetBios           0                  18                      18
SMB               0                  18                      18
TLS/SSL           45                 143.986K (143986)       143.941K (143941)
Other             157                86.93K (86930)          86.773K (86773)

IPv6 flows:
Application       Flows Active       Flows Created       Flows Expired
----------------- ------------------ ------------------- -------------

Nucleus: nucleus1
Activity records sent: 578.306K (578306), last sent 0:00:00 ago
Progress records sent: 172.057K (172057), last sent 0:00:00 ago
Last successful connection: 10:11:22 ago
Successful connections: 11
Last connection failure: 10:11:51 ago
Connection failures: 46
Activity records in queue: 0
Progress records in queue: 1

IPFIX counters:
Exporter: 172.22.197.142 Source port: 36582 Observation domain ID: 1
Messages received: 379.455K (379455)
Template records received: 208
Options template records received: 0
Data records received: 7.308M (7308225)
Options data records received: 0
Unknown template ID errors: 0
Invalid IPFIX messages received: 0
Flow record queue full: 0
```