

8 VXLAN

Virtual eXtensible Local Area Network (VXLAN) is a technology for overlaying virtualized Layer 2 networks over Layer 3 networks.

8.1 VXLAN Overview

VXLAN is an important feature of network cloudification. It allows a virtual network to provide access services to a large number of tenants. In addition, tenants are able to plan their own virtual networks, not limited by the physical network IP addresses or broadcast domains. This greatly simplifies network management.

8.2 Basic VXLAN Concepts

Virtual eXtensible Local Area Network (VXLAN) is a Network Virtualization Overlays (NVO3) technology. VXLAN encapsulates data packets sent from VMs into UDP packets using the IP and MAC addresses of the physical devices as the outer headers, and then transmits the UDP packets over IP networks. Upon receiving the packets, the destination tunnel endpoint decapsulates them and sends them to the destination VM or physical server.

8.3 VXLAN Implementation Principles

VXLAN encapsulates packets in MAC-in-UDP mode so that Layer 2 packets can be transmitted on the Layer 3 networks.

8.4 VXLAN Deployment

A VXLAN is deployed on a layer-3 network to implement communication between virtual layer-2 devices.

8.5 Configuring VXLAN

This section describes how to configure VXLAN on the CLI.

8.6 Configuring EVPN-VXLAN

8.1 VXLAN Overview

VXLAN is an important feature of network cloudification. It allows a virtual network to provide access services to a large number of tenants. In addition, tenants are able to plan their own virtual networks, not limited by the physical network IP addresses or broadcast domains. This greatly simplifies network management.

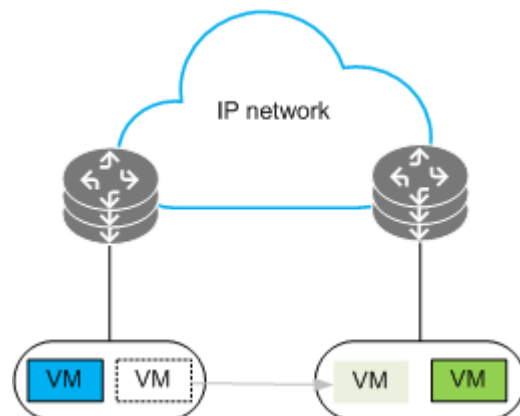
The rapid development of virtualization technologies such as network cloudification faces the following challenges:

- A single physical device is often virtualized into multiple virtual machines for networking. With the increase of application modules, the number of supported VLANs is also increasing. The 802.1Q standard supports a maximum of 4094 VLANs, which cannot meet the requirement of identifying a large number of user groups in a virtual network. This severely limits the number of virtual machines.
- A physical network may be shared by multiple tenants in cloud services, and the tenants may have varying network requirements and overlapping IP and MAC addresses, which cannot be addressed by traditional VLAN technologies.
- After a VM is started, it may need to be migrated to a new server due to resource issues on the original server. For example, the CPU usage is too high, or memory resources are inadequate. To ensure that services are not interrupted during VM migration, the IP address and MAC address of the VM must remain unchanged before and after the migration. This requires that the service network is a Layer 2 network.

The VXLAN technology effectively addresses the preceding problems.

- A 24-bit VXLAN network ID (VNI) is used to identify a network. A maximum of 16777215 (16M) VXLAN segments can be isolated, a lot more than 4096 virtual network identifiers of a VLAN and meeting the requirements of a large number of users.
- The MAC-in-UDP packet encapsulation allows for overlaying virtual Layer 2 networks over Layer 3 networks, decoupling physical networks from virtual networks. The VXLAN network devices are not aware of the VXLAN communication, and the network configurations are easier. Creating VXLANs allows multiple tenants to have independent networks and communicate with each other without IP address conflicts.
- After a large Layer 2 network is created using VXLAN, devices except edge devices in a VXLAN network do not need to identify the VM's MAC address, which reduces the number of MAC addresses learned and enhances network performance. As shown in [Figure 8-1](#), tenants can flexibly migrate their own VMs and plan virtual networks without being limited by the number of IP addresses or size of the broadcast domains on physical networks, significantly simplifying network management.

Figure 8-1 VM migration



8.2 Basic VXLAN Concepts

Virtual eXtensible Local Area Network (VXLAN) is a Network Virtualization Overlays (NVO3) technology. VXLAN encapsulates data packets sent from VMs into UDP packets using the IP and MAC addresses of the physical devices as the outer headers, and then transmits the UDP packets over IP networks. Upon receiving the packets, the destination tunnel endpoint decapsulates them and sends them to the destination VM or physical server.

Figure 8-2 shows the VXLAN gateway service model.

Figure 8-2 VXLAN service model

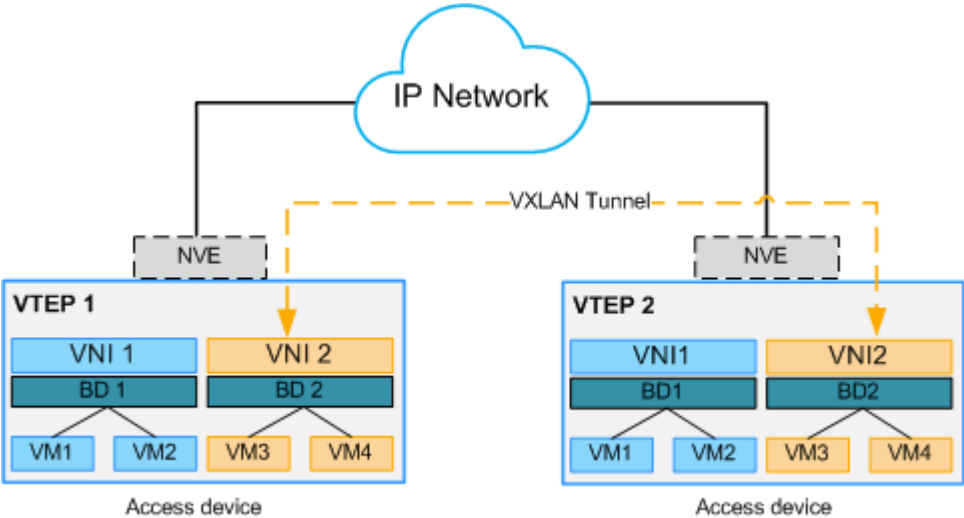


Table 8-1 Basic VXLAN Concepts

Concept	Description
VXLAN tunnel	It is used to forward VXLAN packets. It is identified by the local VTEP address + remote VTEP address.
NVE (Network virtualization edge)	It is the entity for implementing network virtualization. It is used for managing local VTEP addresses, VXLAN tunnels, and ingress replication lists.

Concept	Description
VTEP (VXLAN tunnel endpoint)	A device that supports VXLAN is a VTEP. It is encapsulated in NVE for encapsulating and decapsulating VXLAN packets. A VTEP is connected to a physical network and is assigned with a physical network IP address, which is irrelevant to the virtual network. For a VXLAN packet, the source IP address is the VTEP address of the local node and the destination IP address is the VTEP address of the peer node. These two VTEP addresses correspond to a VXLAN tunnel.
BD (Bridge Domain)	It is a Layer 2 broadcast domain for forwarding data packets on a VXLAN. On a VXLAN, VNIs are mapped into the BDs one on one. The BD is the entity for forwarding data packets on the VXLAN network.
VNI (VXLAN network identifier)	It is similar to VLAN ID and is used to identify a VXLAN. VMs in different VXLANs cannot communicate directly. A VNI (24 bits) indicates an ISP and a maximum of 16M ISPs can be supported.

8.3 VXLAN Implementation Principles

VXLAN encapsulates packets in MAC-in-UDP mode so that Layer 2 packets can be transmitted on the Layer 3 networks.

VXLAN packet encapsulation

The MAC-in-UDP packet encapsulation is implemented by adding a VXLAN header to Layer 2 packets and then encapsulating the packets in UDP packets. [Figure 8-3](#) shows the VXLAN packet encapsulation mode and packet format.

Figure 8-3 VXLAN packet format

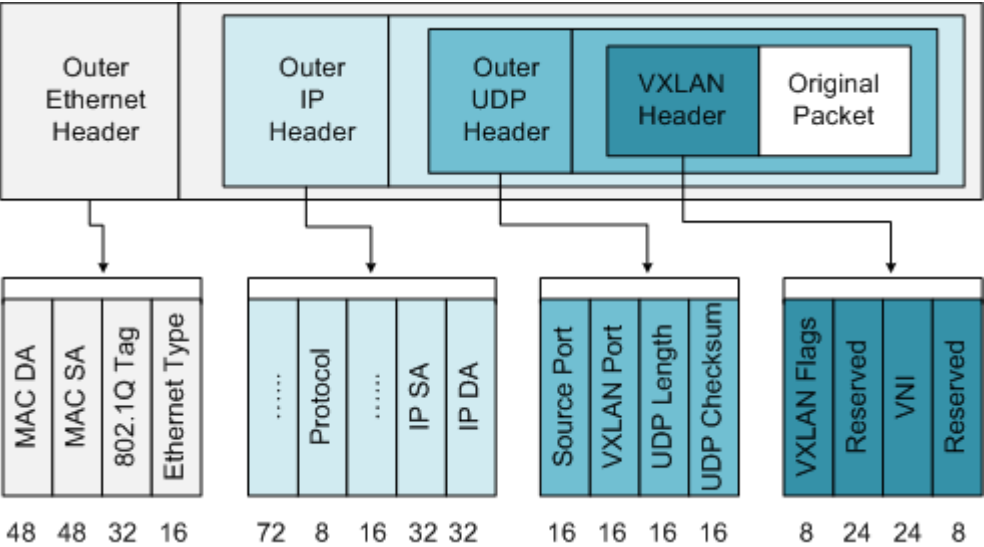


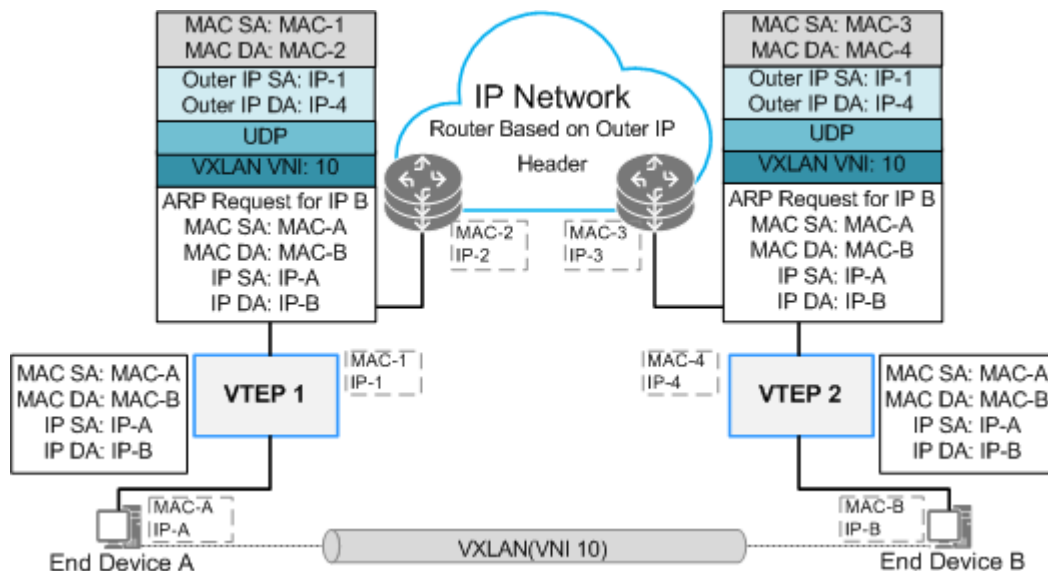
Table 8-2 VXLAN packet description

Packet Field	Description
VXLAN Header	VNI: indicates the 24-bit VXLAN identifier, which identifies a VXLAN.
Outer UDP Header	• DestPort: indicates a destination UDP port. • Source Port: indicates the source port.
Outer IP Header	• IP SA: indicates the source IP address, which is the IP address of the VTEP where the sender VM resides. • IP DA: indicates the destination IP address, which is the IP address of the VXLAN Tunnel EndPoint (VTEP) where the destination VM resides.
Outer Ethernet Header	• MAC DA: indicates the MAC address of the next-hop IP address in the routing table of the VTEP where the destination VM resides. • MAC SA: indicates the MAC address of the VTEP where the sender VM resides. • Ethernet Type: indicates the Ethernet packet type.

VXLAN packet forwarding process

VXLAN uses VTEPs, such as OLTs, to map user terminals to VXLANs and encapsulate and decapsulate packets to for VXLAN packet transmission. [Figure 8-4](#) shows the principles of the VXLAN.

Figure 8-4 VXLAN principles



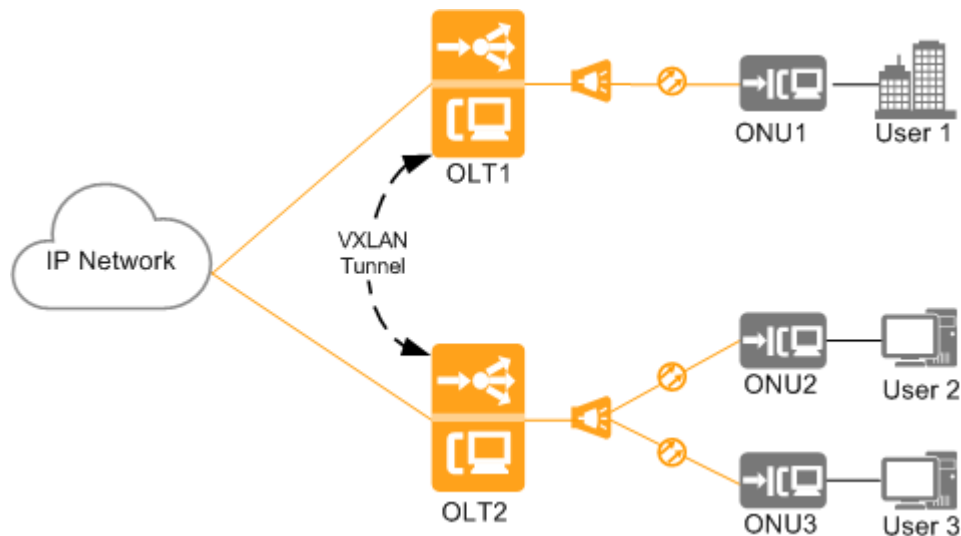
VMs A and B are on the same VXLAN. VM A sends packet M to VM B. The process is described as follows:

1. A sends packet M, which contains the virtual MAC addresses of the sender and receiver.
2. VTEP1 connected to A receives the packet, searches for the VTEP where the sender resides and the VTEP connected to the receiver, adds a VXLAN header, and a UDP header, and then an IP header to the packet, and sends the packet to the destination (VTEP2).
3. The packet is transmitted to the destination (VTEP2) through the physical network.
4. Upon receiving the packet, VTEP2 removes the outer IP header and UDP header of the packet, checks the VNI of the packet and the destination MAC address of the internal data frame, determines that receiver B is connected to the local VTEP, removes the VXLAN header, and sends the packet to B.
5. B receives the packet and the transmission is complete.

8.4 VXLAN Deployment

A VXLAN is deployed on a layer-3 network to implement communication between virtual layer-2 devices.

The following figure shows the VXLAN networking. OLT1 and OLT2 are located on different layer 2 networks. You can configure VXLANs on the OLTs to implement virtual layer 2 communication between OLT1 and OLT2.

Figure 8-5 VXLAN networking

8.5 Configuring VXLAN

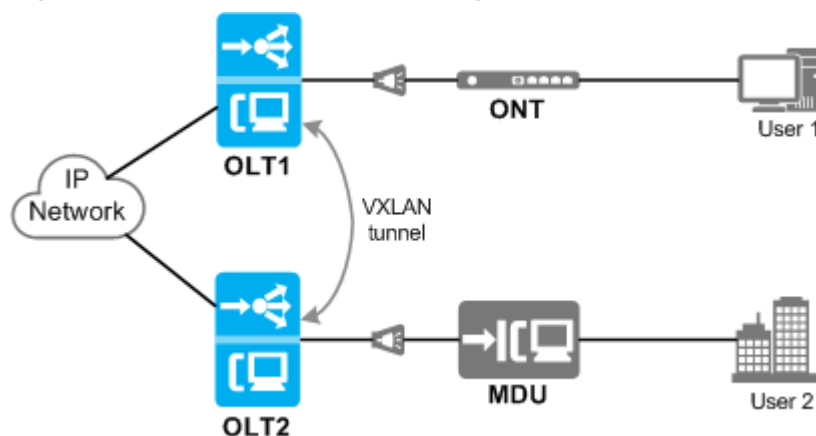
This section describes how to configure VXLAN on the CLI.

Prerequisites

1. Service configurations (such as VLAN and service flow configurations) have been completed, and user services are normal.
2. The loopback IP address has been configured.

Networking

As shown in [Figure 8-6](#), when two OLTs are located on different Layer 2 networks, you can configure a VXLAN for the two OLTs to establish a VXLAN tunnel. In this way, the OLTs can communicate with each other at Layer 2, and Layer 2 packets can be transmitted between User1 and User2.

Figure 8-6 VXLAN tunnel networking

The configuration procedures for OLT1 and OLT2 are the same. The following uses OLT1 as an example.

Procedure

- Step 1** Run the **interface nve** command to create an NVE interface and enter the NVE view.
 - Step 2** Run the **source ip-address** command to configure an IP address for the source VTEP.
 - Step 3** Run the **bridge-domain bd-id** command to create a bridge domain (BD) and enter the BD view.
 - Step 4** Run the **vlan vni vni-id** command to create a VXLAN network identifier (VNI).
 - Step 5** Run the **vni vni-id vni head-end peer-list ip-address** command to configure a VNI ingress replication list and specify the VNI and IP address of the peer VTEP.
 - Step 6** Run the **bd-binding vlan-id** command to bind the VLAN/service port to the BD.
- End

Example

To create vxlan vni 1, associate it with bridge-domain 1, and specify the VNI and the IP address of the peer VTEP of the tunnel as 10.10.11.11, do as follows:

```
huawei(config)#interface nve 1
huawei(config-Nve1)#source 10.10.10.11
huawei(config-Nve1)#vni 1 head-end peer-list 10.10.11.11
huawei(config-Nve1)#quit

huawei(config)#bridge-domain 1
huawei(config-bd-1)#vxlan vni 1
huawei(config-bd-1)#bd-binding vlan 100
huawei(config-bd-1)#quit
```

8.6 Configuring EVPN-VXLAN

EVPN Concepts

Ethernet VPN (EVPN) is a VPN technology based on Layer 2 network interconnection. Similar to BGP/MPLS IP VPN, EVPN uses the reachable information extended through the extended Border Gateway Protocol (BGP), to transfer MAC address learning and advertisement at Layer 2 network of different sites from the data plane to control plane.

EVPN VXLAN Configuration in Single Homing and One Service Flow Working

[Figure 8-7](#) shows the single homing and one service flow working.

Figure 8-7 Single homing and one service flow working

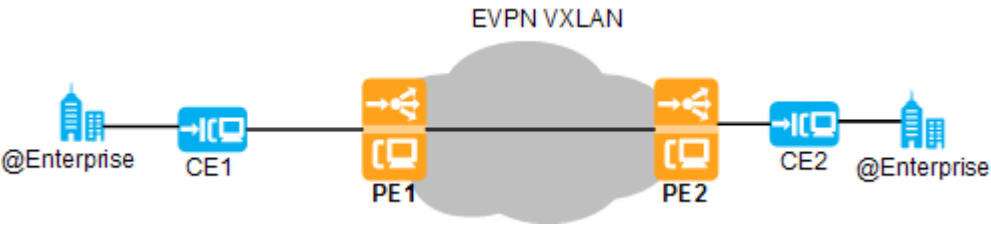


Table 8-3 Device interface parameters

Device	Parameter Configuration
Provider Edge (PE1)	• loopback0 IP: 10.1.1.1 32 • Stand VLAN: 1001, Port 0/9/3 • Smart VLAN: 12, Port 0/9/0 • evpl Route Distinguisher (RD): 101:1 • evpl RT: 200:1
PE2	• loopback0 IP: 10.1.1.2 32 • Stand VLAN: 1001, Port 0/9/3 • Smart VLAN: 12, Port 0/9/0 • Smart VLAN: 32, Port 0/9/1 • evpl RD: 101:1 • evpl RT: 200:1

The following uses PE1 as an example to describe the detailed configuration:

1.

In Loopback 0 mode, set the IP address to 10.1.1.1 32.
huawei(config)#interface loopback 0
huawei(config-if-loopback0)#ip address 10.1.1.1 32
huawei(config-if-loopback0)#quit
2.

Add smart VLAN 12 in batches for the Internet access service, and add upstream port 0/9/0 to the VLAN. Add standard-type management VLAN 1001, and add upstream port 0/9/3 to the VLAN.
huawei(config)#vlan 12 smart
huawei(config)#vlan 1001 standard
huawei(config)#port vlan 12 0/9 0
huawei(config)#port vlan 1001 0/9 3
3.

Create a VLANIF interface with IP address 10.1.1.12 24 for VLAN 12.
huawei(config)#interface vlanif 12
huawei(config-if-vlanif12)#ip address 10.1.1.12 24
huawei(config-if-vlanif12)#quit
4.

Run OSPF. Set ID of the area where the interface locates to 0.0.0.0, and specify main IP address of the interface running OSPF to be in network segment 10.1.1.1.
huawei(config)#ospf router-id 10.1.1.1
huawei(config-ospf-1)#area 0.0.0.0
huawei(config-ospf-1-area-0.0.0.0)#network 10.1.1.1 0.0.0.0
huawei(config-ospf-1-area-0.0.0.0)#network 10.1.1.13 0.0.0.255
huawei(config-ospf-1-area-0.0.0.0)#quit
huawei(config-ospf-1)#quit

5. Create EVPN instance evpn1, and set RD to 101:1 and RT to 200:1.

```
huawei(config)#evpn vpn-instance evpn1 vpws
huawei(config-vpws-evpn-instance-evpn1)#route-distinguisher 101:1
huawei(config-vpws-evpn-instance-evpn1)#vpn-target 200:1 both
huawei(config-vpn-instance-evpn1)#quit
```
6. Create an NVE interface, configure IP address for source VTEP, and create a dynamic P2P VXLAN tunnel.

```
huawei(config)#interface nve 1
huawei(config-Nve1)#source 10.1.1.1
huawei(config-Nve1)#vni 1 evpl protocol bgp
huawei(config-Nve1)#quit
```
7. Create an EVPL instance with ID 1 in VXLAN mode, bind it to VNI 1, bind the EVPL instance to EVPN instance, and enable local and remote service ID carried in the EVPL instance.

```
huawei(config)#evpl instance 1 vxlan-mode
huawei(config-evpl-vxlan1)#local-vni 1
huawei(config-evpl-vxlan1)#evpn binding vpn-instance evpn1
huawei(config-evpl-vxlan1)#local-service-id 1 remote-service-id 2
huawei(config-evpl-vxlan1)#evpl-binding vlan 1001
huawei(config-evpl-vxlan1)#quit
```
8. Set the router ID in BGP mode, set the peer AS number of the IPv4 peer 10.1.1.1 to 100, and specify the source interface and source address of BGP packets.

```
huawei(config)#bgp 100
huawei(config-bgp)#router-id 10.1.1.1
huawei(config-bgp)#peer 10.1.1.1 as-number 100
huawei(config-bgp)#peer 10.1.1.1 connect-interface loopback 0
huawei(config-bgp)#peer 10.1.1.1 enable
huawei(config-bgp)#l2vpn-family evpn
huawei(config-bgp-af-evpn)#peer 10.1.1.1 enable
huawei(config-bgp-af-evpn)#quit
huawei(config-bgp)#quit
```

EVPN VXLAN Configuration in Dual Homing and One Service Flow Working

Figure 8-8 shows the dual homing and one service flow working, in which, PE3 is added and CE1 is dual homed to PE1 and PE3 based on single homing and one service flow working.

- Only one of the ACs on PE1 and PE3 can work at a time.
- The ESI of the user-side interface on PE3 must be the same as that on PE1.
- The local and remote service IDs of instance evpl on PE3 must be the same as that on PE1.

Figure 8-8 Dual homing and one service flow working

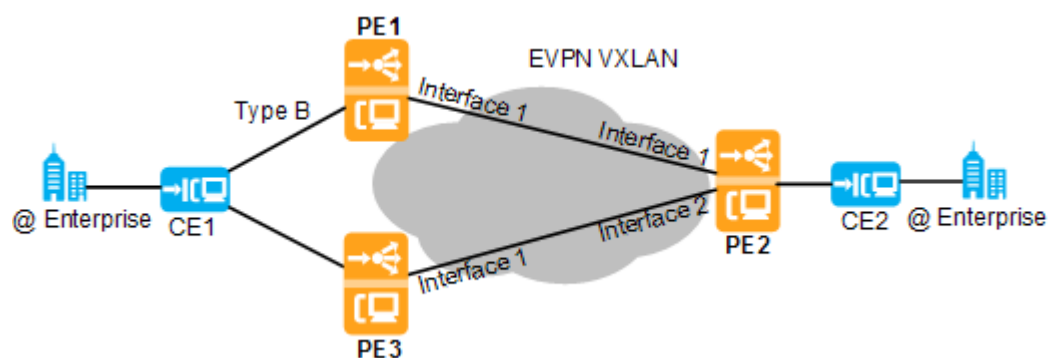


Table 8-4 Device interface parameters

Device	Parameter Configuration
PE1	• loopback0 IP: 10.1.1.1 32 • StandVLAN: 1001, Port 0/9/3 • SmartVLAN: 12, Port 0/9/0 • evpl RD: 101:1 • evpl RT: 200:1
PE2	• loopback0 IP: 10.1.1.2 32 • StandVLAN: 1001, Port 0/9/3 • SmartVLAN: 12, Port 0/9/0 • SmartVLAN: 32, Port 0/9/1 • evpl RD: 101:1 • evpl RT: 200:1
PE3	• loopback0 IP: 10.1.1.3 32 • StandVLAN: 1001, Port 0/9/3 • SmartVLAN: 32, Port 0/9/0 • evpl RD: 101:1 • evpl RT: 200:1

In dual homing and one service flow working, configurations of PE1 and PE3 are the same as those in single homing and one service flow working. The following describes how to configure PE2:

1. In Loopback 0 mode, set the IP address to 10.1.1.2 32.

```
huawei(config)#interface loopback 0
huawei(config-if-loopback0)#ip address 10.1.1.2 32
huawei(config-if-loopback0)#quit
```
2. Add smart VLAN 12 in batches for the Internet access service, and add upstream port 0/9/0 to the VLAN. Add standard-type management VLAN 1001, and add upstream port 0/9/3 to the VLAN.

```
huawei(config)#vlan 12 smart
huawei(config)#vlan 32 smart
huawei(config)#vlan 1001 standard
huawei(config)#port vlan 12 0/9 0
huawei(config)#port vlan 32 0/9 1
huawei(config)#port vlan 1001 0/9 3
```
3. Create a VLANIF interface with IP address 10.1.1.12 24 for VLAN 12.

```
huawei(config)#interface vlanif 12
huawei(config-if-vlanif12)#ip address 10.1.1.12 24
huawei(config-if-vlanif12)#quit
```
4. Run OSPF. Set ID of the area where the interface locates to 0.0.0.0, and specify main IP address of the interface running OSPF to be in network segment 10.1.1.1.

```
huawei(config)#ospf router-id 10.1.1.1
huawei(config-ospf-1)#area 0.0.0.0
huawei(config-ospf-1-area-0.0.0.0)#network 10.1.1.1 0.0.0.0
huawei(config-ospf-1-area-0.0.0.0)#network 10.1.1.13 0.0.0.255
huawei(config-ospf-1-area-0.0.0.0)#network 10.1.1.32 0.0.0.255
```

```
huawei(config-ospf-1-area-0.0.0.0)#quit
huawei(config-ospf-1)#quit
```

5. Create EVPN instance evpn1, and set RD to 101:1 and RT to 200:1.

```
huawei(config)#evpn vpn-instance evpn1 vpws
huawei(config-vpws-evpn-instance-evpn1)#route-distinguisher 102:1
huawei(config-vpws-evpn-instance-evpn1)#vpn-target 200:1 both
huawei(config-vpws-evpn-instance-evpn1)#remote frr enable
huawei(config-vpn-instance-evpn1)#quit
```

6. Create an NVE interface, configure IP address for source VTEP, and create a dynamic P2P VXLAN tunnel.

```
huawei(config)#interface nve 1
huawei(config-Nve1)#source 10.1.1.1
huawei(config-Nve1)#vni 2 evpl protocol bgp
huawei(config-Nve1)#quit
```

7. Create an EVPL instance with ID 1 in VXLAN mode, bind it to VNI 2, bind the EVPL instance to EVPN instance evpn1, and enable local and remote service ID carried in the EVPL instance.

```
huawei(config)#evpl instance 1 vxlan-mode
huawei(config-evpl-vxlan1)#local-vni 2
huawei(config-evpl-vxlan1)#evpn binding vpn-instance evpn1
huawei(config-evpl-vxlan1)#local-service-id 2 remote-service-id 2
huawei(config-evpl-vxlan1)#evpl-binding vlan 1001
huawei(config-evpl-vxlan1)#quit
```

8. Set the router ID in BGP mode, set the peer AS number of the IPv4 peer 10.1.1.1 to 100, and specify the source interface and source address of BGP packets.

```
huawei(config)#bgp 100
huawei(config-bgp)#router-id 10.1.1.1
huawei(config-bgp)#peer 10.1.1.1 as-number 100
huawei(config-bgp)#peer 10.1.1.1 connect-interface loopback 0
huawei(config-bgp)#peer 10.1.1.1 enable
huawei(config-bgp)#peer 10.1.1.3 as-number 100
huawei(config-bgp)#peer 10.1.1.3 connect-interface loopback 0
huawei(config-bgp)#peer 10.1.1.3 enable
huawei(config-bgp)#l2vpn-family evpn
huawei(config-bgp-af-evpn)#peer 10.1.1.1 enable
huawei(config-bgp-af-evpn)#peer 10.1.1.3 enable
huawei(config-bgp-af-evpn)#quit
huawei(config-bgp)#quit
```

Reference Standard

Reference standards: RFC7432 and RFC4360.