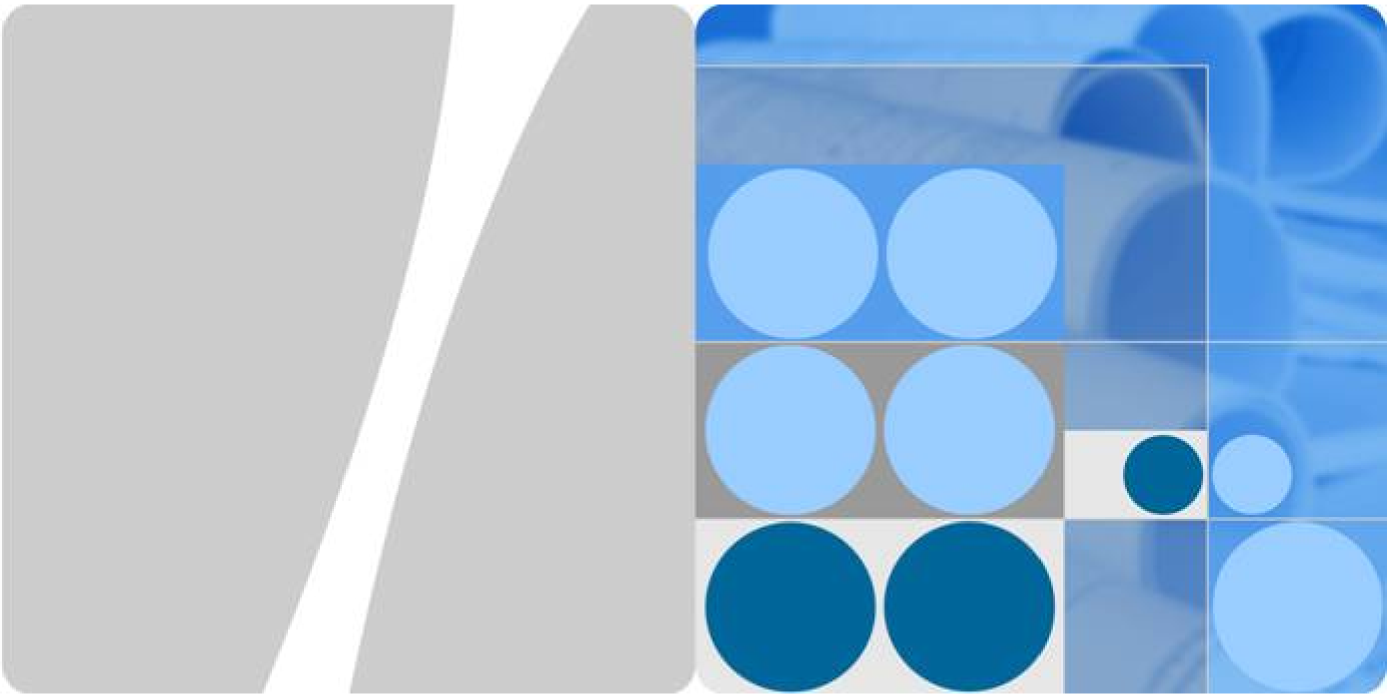


Huawei CloudEngine 12800 Series

VXLAN Technology White Paper

Issue	**06**
Date	**2016-07-28**

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.huawei.com

Email: support@huawei.com

Contents

1 VXLAN Overview.. 1

2 VXLAN Deployment Modes... 4

3 Principles..9

3.1 Basic Concepts.. 10

3.2 Gateway Classification.. 15

3.3 VXLAN Packet Format... 17

3.4 Tunnel Establishment (Static Mode)... 18

3.5 Tunnel Establishment (MP-BGP Dynamic Mode).. 18

3.6 Tunnel Establishment (BGP EVPN Dynamic Mode)..24

3.7 Data Packet Forwarding.. 30

3.8 ARP Broadcast Suppression...38

3.9 All-Active VXLAN Gateway...40

3.10 VXLAN Dual-Active Access.. 45

3.11 Application for Inter-Domain Active-Active VXLAN Gateways...51

3.12 VXLAN QoS.. 56

4 Applications..58

4.1 Application for Communication Between Terminal Users on a VXLAN....................................59

4.2 Application for Communication Between Terminal Users on a VXLAN and Legacy Network................................ 60

4.3 Application in VM Migration Scenarios.. 62

5 Configuration Notes.. 64

6 Configuring VXLAN (Through the Agile Controller-DCN)...70

7 Configuring VXLAN in Single-Node, Centralized Gateway, and Static Mode.............74

7.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function..................................79

7.2 Configuring a Service Access Point.. 80

7.3 Configuring a VXLAN Tunnel...84

7.4 Configuring a Layer 3 VXLAN Gateway.. 85

7.5 (Optional) Configuring Centralized All-Active Gateways for the VXLAN Network.. 86

7.6 (Optional) Configuring ARP Broadcast Suppression..89

7.7 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages...91

7.8 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting... 92

7.9 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection... 94

7.10 (Optional) Optimizing ACL Resource Usage on the VXLAN....94
7.11 (Optional) Optimizing Load Balancing on the VXLAN Network.... 96
7.12 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode.... 97
7.13 Checking the Configurations.... 98
8 Configuring VXLAN in Single-Node, Centralized Gateway, and BGP EVPN Mode.100
8.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function.... 105
8.2 Configuring a Service Access Point.... 106
8.3 Configuring a VXLAN Tunnel.... 110
8.4 Configuring a Layer 3 VXLAN Gateway....115
8.5 (Optional) Configuring Centralized All-Active Gateways for the VXLAN Network.... 116
8.6 (Optional) Configuring ARP Broadcast Suppression....119
8.7 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages.... 120
8.8 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting.... 121
8.9 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection.... 123
8.10 (Optional) Optimizing ACL Resource Usage on the VXLAN....124
8.11 (Optional) Optimizing Load Balancing on the VXLAN Network.... 125
8.12 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode.... 126
8.13 Checking the Configurations.... 127
9 Configuring VXLAN in Single-Node, Distributed Gateway, and MP-BGP Mode......129
9.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function.... 133
9.2 Configuring a VXLAN Service Access Point.... 134
9.3 Configuring a VXLAN Tunnel and a Layer 3 VXLAN Gateway....137
9.4 (Optional) Configuring ARP Broadcast Suppression....141
9.5 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages.... 143
9.6 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting.... 144
9.7 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection.... 146
9.8 (Optional) Optimizing ACL Resource Usage on the VXLAN....147
9.9 (Optional) Optimizing Load Balancing on the VXLAN Network.... 148
9.10 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode.... 149
9.11 Checking the Configurations.... 150
10 Configuring VXLAN in Single-Node, Distributed Gateway, and BGP EVPN Mode
....152
10.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function.... 156
10.2 Configuring a VXLAN Service Access Point.... 157
10.3 Configuring a VXLAN Tunnel and a Layer 3 VXLAN Gateway....160
10.4 (Optional) Configuring ARP Broadcast Suppression....171
10.5 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages.... 173
10.6 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting.... 174
10.7 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection.... 176
10.8 (Optional) Configuring IP Address Conflict Detection Parameters.... 176
10.9 (Optional) Optimizing ACL Resource Usage on the VXLAN....177

10.10 (Optional) Optimizing Load Balancing on the VXLAN Network....178
10.11 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode....179
10.12 Checking the Configurations....180

11 Maintaining VXLAN....182

11.1 Configuring the VXLAN Alarm Function....183
11.2 Collecting and Checking VXLAN Packet Statistics....183
11.3 Clearing VXLAN Packet Statistics....184
11.4 Checking Statistics about MAC Address Entries in a BD....185
11.5 Clearing Statistics about Dynamic MAC Address Entries in a BD....185
11.6 Configuring BGP EVPN Soft Reset....185
11.7 Resetting BGP EVPN Connections....186
11.8 Monitoring the VXLAN Operating Status....186

12 Configuration Examples (Single-Node Mode)....188

12.1 Example for Configuring VXLAN in Centralized Gateway Mode for Static Tunnel Establishment....189
12.2 Example for Configuring VXLAN in Centralized Gateway Mode Using BGP EVPN....194
12.3 Example for Configuring VXLAN in Distributed Gateway Mode Using MP-BGP....201
12.4 Example for Configuring VXLAN in Distributed Gateway Mode Using BGP EVPN....209
12.5 Example for Configuring All-Active VXLAN Gateways....216
12.6 Example for Configuring Dual-Active VXLAN Access....225

13 References....233

1 VXLAN Overview

This section describes the definition, purpose, and benefits of the Virtual eXtensible Local Area Network (VXLAN).

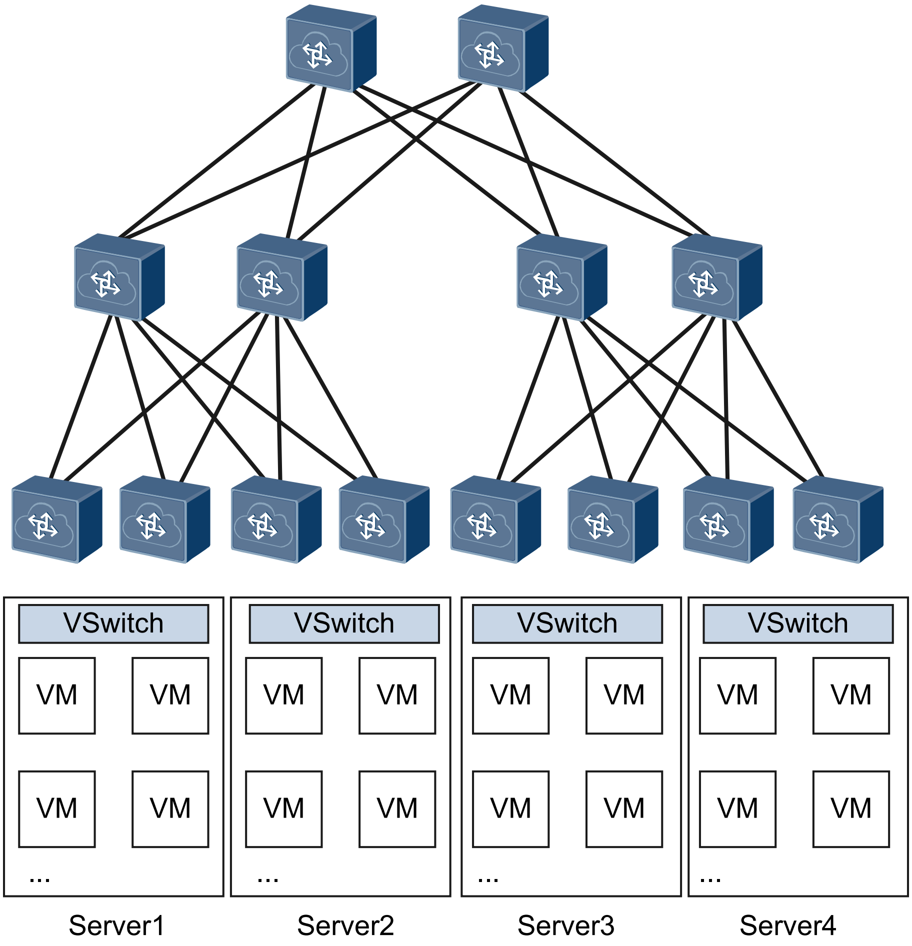
Definition

Virtual extensible local area network (VXLAN) defined in RFC 7348 is a Network Virtualization over Layer 3 (NVO3) technology that uses MAC-in-UDP encapsulation.

Purpose

As a widely deployed core cloud computing technology, server virtualization greatly reduces IT and O&M costs and improves service deployment flexibility.

Figure 1-1 Server virtualization



On the network shown in **Figure 1-1**, a server is virtualized into multiple virtual machines (VMs), each of which functions as a host. A great increase in the number of hosts causes the following problems:

- VM scale is limited by the network specification.

  On a legacy large Layer 2 network, data packets are forwarded at Layer 2 based on MAC entries. However, there is a limit on the MAC table capacity, which subsequently limits the number of VMs.

- Network isolation capabilities are limited.

  Most networks currently use VLANs to implement network isolation. However, the deployment of VLANs on large-scale virtualized networks has the following limitations:

  - The VLAN tag field defined in IEEE 802.1Q has only 12 bits and can support only a maximum of 4094 VLANs, which cannot meet user identification requirements of large Layer 2 networks.
  - VLANs on legacy Layer 2 networks cannot adapt to dynamic network adjustment.

- VM migration scope is limited by the network architecture.

  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, when the CPU usage is too high or memory resources are inadequate. To ensure uninterrupted services during VM migration, the IP and MAC addresses of the VM must remain unchanged. To carry this out, the service network must be a Layer 2 network and also provide multipathing redundancy backup and reliability.

VXLAN addresses the preceding problems on large Layer 2 networks.

- Eliminates VM scale limitations imposed by network specifications.

  VXLAN encapsulates data packets sent from VMs into UDP packets and encapsulates IP and MAC addresses used on the physical network into the outer headers. As a result, the network is aware of only the encapsulated parameters and not the inner data. This implementation greatly reduces the MAC address specification requirements of large Layer 2 networks.

- Provides greater network isolation capabilities.

  VXLAN uses a 24-bit network segment ID, called a VXLAN network identifier (VNI), to identify users. This VNI is similar to a VLAN ID, but supports a maximum of 16M VXLAN segments.

- Eliminates VM migration scope limitations imposed by network architecture.

  VXLAN uses MAC-in-UDP encapsulation to extend Layer 2 networks. It encapsulates Ethernet packets into IP packets for these Ethernet packets to be transmitted over routes, and does not need to be aware of VMs' MAC addresses. Because there is no limitation on Layer 3 network architecture, Layer 3 networks are scalable and have strong automatic fault rectification and load balancing capabilities. This allows for VM migration irrespective of the network architecture.

Benefits

As server virtualization is being rapidly deployed on data centers based on physical network infrastructure, VXLAN offers the following benefits:

- A maximum of 16M VXLAN segments are supported using 24-bit VNIs, which allows a data center to accommodate multiple tenants.
- Non-VXLAN network edge devices do not need to identify the VM's MAC address, which reduces the number of MAC addresses that have to be learned and enhances network performance.
- MAC-in-UDP encapsulation extends Layer 2 networks, decoupling between physical and virtual networks. 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.

2 VXLAN Deployment Modes

Currently, you can deploy a VXLAN network in **Single-node mode** or **Controller mode**.

- **Single-node mode**: In the traditional network deployment node, you need to log in to each device to configure the devices according to the network plan. Collaboration with cloud platforms cannot be implemented in cloud computing data centers for automatic network deployment.
- **Controller mode**: To help control and deploy a large Layer 2 network, a controller can be used. A controller is a unified network control platform that orchestrates and manages network resources and cooperates with the cloud platform to implement automatic service and network provisioning.

Agile Controller-DCN Mode

- Introduction to Agile Controller-DCN Mode

  In Agile Controller-DCN mode, the Agile Controller-DCN dynamically establishes VXLAN tunnels. The Agile Controller-DCN uses NETCONF to establish VXLAN tunnels with devices and OpenFlow to control packet forwarding through the tunnels.

  As shown in **Figure 2-1**, the Agile Controller-DCN can directly manage the virtual network and obtain virtual network information from the Neutron. The Agile Controller-DCN dynamically calculates network configurations based on virtual network information and automatically maps the information to physical networks.

Figure 2-1 Networking of the Agile Controller-DCN + VXLAN solution

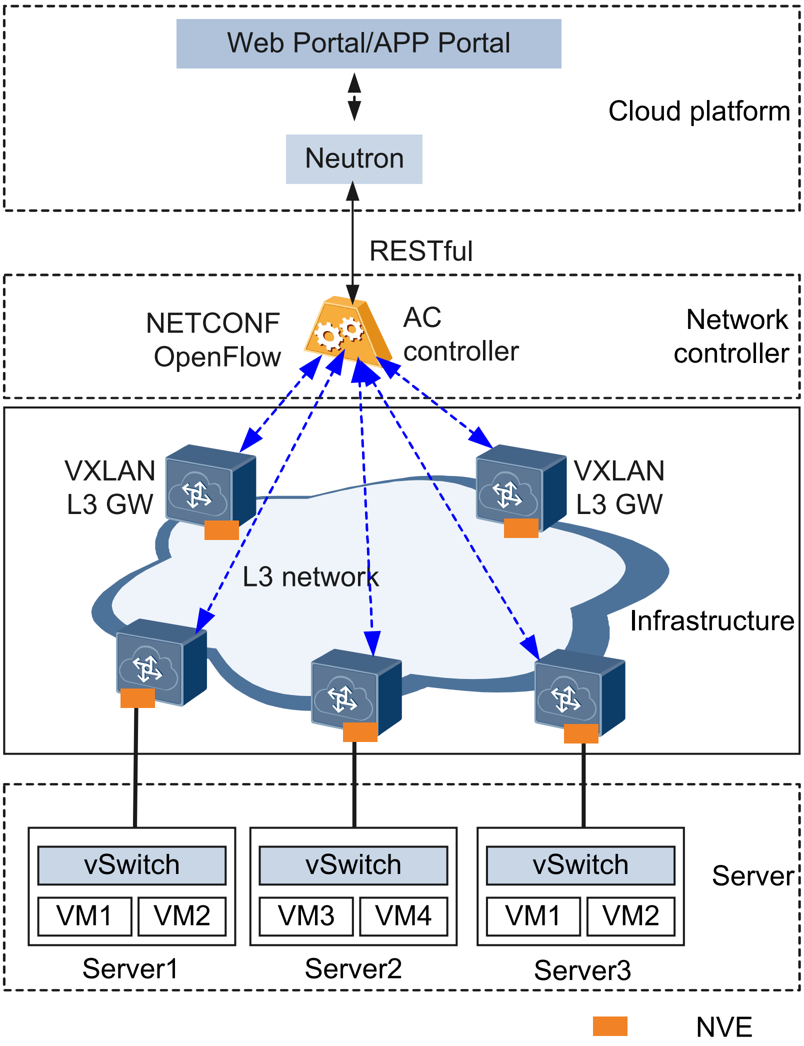


Table 2-1 describes the layers in the Agile Controller-DCN + VXLAN solution and functions of each layer.

Table 2-1 Layers in the Agile Controller-DCN + VXLAN solution

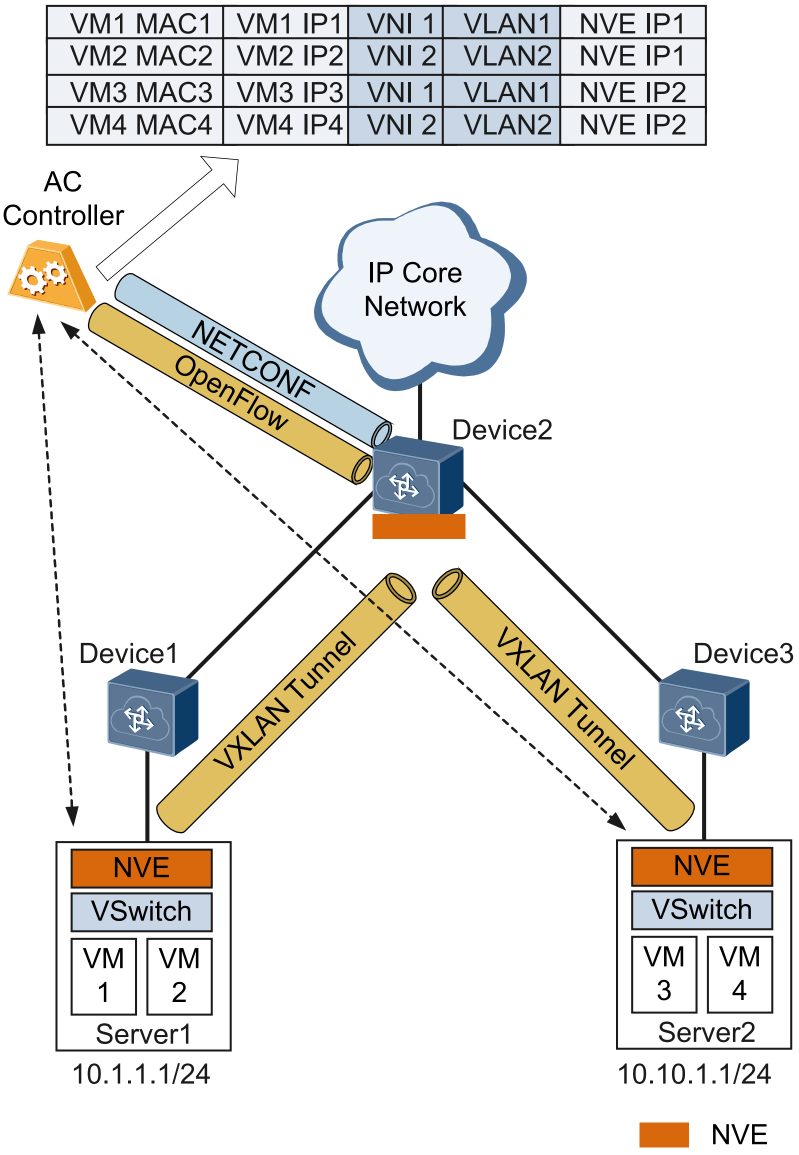
Layer	Description
Cloud platform	Schedules network, computing, and storage resources as required and provides service management and O&M interface. Neutron is the component of the cloud platform and is used to provide network services.
Network controller	Implements network modeling and instantiation. The Agile Controller-DCN uses RESTful interfaces to receive network modeling configuration from the cloud platform and convert the configurations into related commands, and uses the OpenFlow and NETCONF protocols to establish channels with network devices in the infrastructure layer and deliver commands to them.

Layer	Description
Infrastructure	Plans physical and virtual networks in a unified manner. ● Uses hardware-based VXLAN gateways to improve service performance. ● Is compatible with traditional VLANs.

- Channel Establishment and Maintenance Between the Agile Controller-DCN and Forwarder

  The Agile Controller-DCN can detect the tenant status in real time and obtain the virtual network information from the cloud platform. As shown in **Figure 2-2**, the Agile Controller-DCN dynamically calculates network configurations and flow table information based on virtual network information and automatically maps the information to physical networks after tenants go online.

Figure 2-2 Channel establishment and maintenance between the Agile Controller-DCN and device



VM1 MAC1	VM1 IP1	VNI 1	VLAN1	NVE IP1
VM2 MAC2	VM2 IP2	VNI 2	VLAN2	NVE IP1
VM3 MAC3	VM3 IP3	VNI 1	VLAN1	NVE IP2
VM4 MAC4	VM4 IP4	VNI 2	VLAN2	NVE IP2

As shown in **Figure 2-2**, network administrators must have completed the NETCONF and mandatory VXLAN configuration (for example, creating NVE interfaces and configuring VTEP IP addresses) on the device using CLI or Zero Touch Provisioning (ZTP). After the configurations are complete, the Agile Controller-DCN can manage the device using NETCONF.

- The Agile Controller-DCN automatically allocates a controller node to a network device based on the load of the controller cluster and establishes an OpenFlow channel between the network device and the controller node.
- The Agile Controller-DCN receives instructions from the cloud platform and converts the instructions to configurations of the network device to implement automatic service provisioning.
- Supporting the ARP protocol stack, the Agile Controller-DCN can learn and process ARP packets. The network device and Agile Controller-DCN use OpenFlow to transmit ARP packets.

- The Agile Controller-DCN delivers the ARP flow table to the network device through the OpenFlow channel to guide packet forwarding. If the OpenFlow channel between the Agile Controller-DCN and network device is disconnected, the ARP flow table will not be aged immediately; therefore, packet forwarding is not interrupted. After the OpenFlow channel is reconnected, the Agile Controller-DCN synchronizes the ARP flow table with the network device to ensure consistent entries.

3 Principles

About This Chapter

This section describes VXLAN implementation.

3.1 Basic Concepts

3.2 Gateway Classification

3.3 VXLAN Packet Format

3.4 Tunnel Establishment (Static Mode)

3.5 Tunnel Establishment (MP-BGP Dynamic Mode)

3.6 Tunnel Establishment (BGP EVPN Dynamic Mode)

3.7 Data Packet Forwarding

3.8 ARP Broadcast Suppression

3.9 All-Active VXLAN Gateway

3.10 VXLAN Dual-Active Access

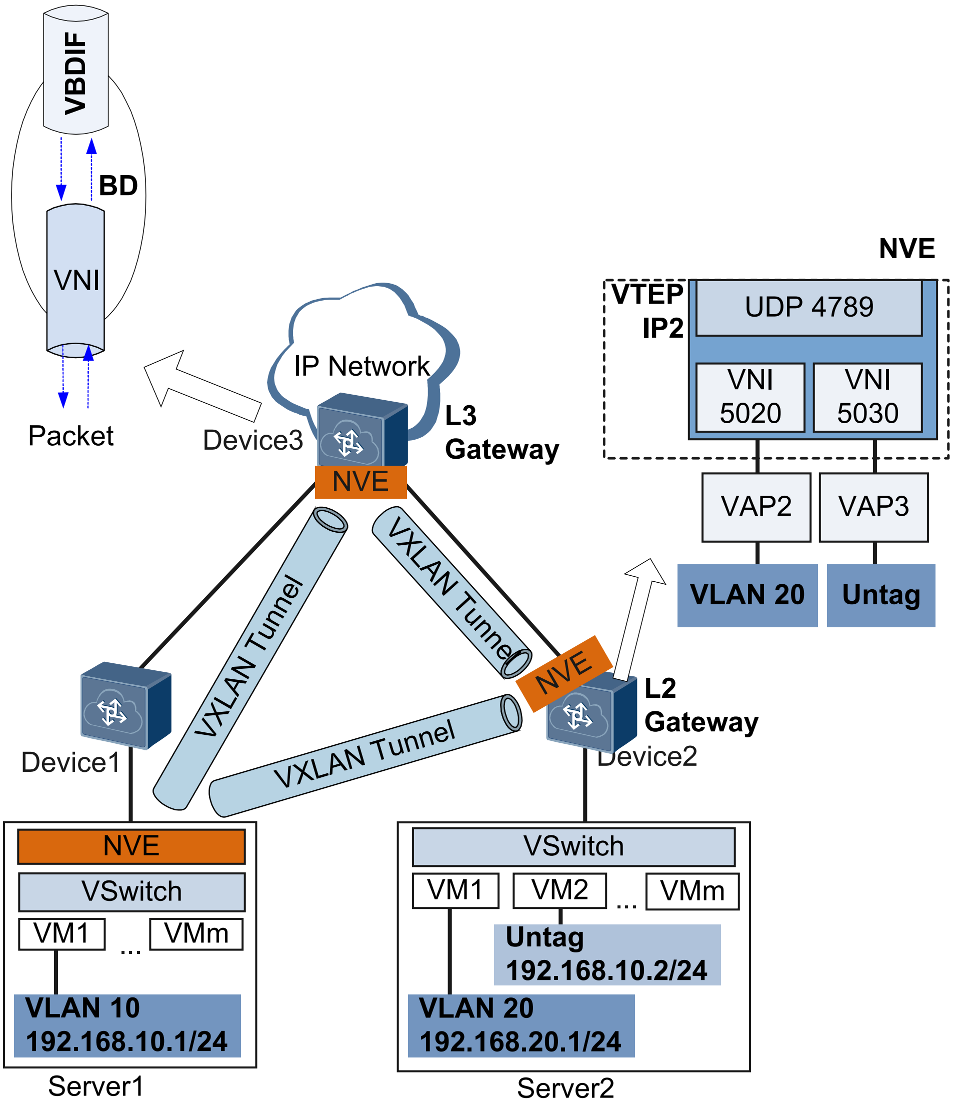
3.11 Application for Inter-Domain Active-Active VXLAN Gateways

3.12 VXLAN QoS

3.1 Basic Concepts

Virtual extensible local area network (VXLAN) is an NVO3 network virtualization technology that encapsulates data packets sent from virtual machines (VMs) into UDP packets and encapsulates IP and MAC addresses used on the physical network in outer headers before sending the packets over an IP network. The egress tunnel endpoint then decapsulates the packets and sends the packets to the destination VM.

Figure 3-1 VXLAN architecture



VXLAN 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. Table 3-1 describes VXLAN concepts.

Table 3-1 VXLAN concepts

Concept	Description
Network virtualization edge (NVE)	A network entity that is deployed at the network edge and implements network virtualization functions. **NOTE** vSwitches on devices and servers can function as NVEs. The following NVE deployment modes are available where NVEs are deployed. ● Hardware mode: All NVEs are deployed on NVE-capable devices, which perform VXLAN encapsulation and decapsulation. ● Software mode: All NVEs are deployed on vSwitches, which perform VXLAN encapsulation and decapsulation. ● Hybrid mode: Some NVEs are deployed on vSwitches, and others on NVE-capable devices. Both vSwitches and NVE-capable devices may perform VXLAN encapsulation and decapsulation.
VXLAN tunnel endpoint (VTEP)	A VXLAN tunnel endpoint that encapsulates and decapsulates VXLAN packets. It is represented by an NVE on the controller. A VTEP connects to a physical network and is assigned a physical network IP address. This IP address is irrelevant to virtual networks. In VXLAN packets, the source IP address is the local node's VTEP address, and the destination IP address is the remote node's VTEP address. This pair of VTEP addresses corresponds to a VXLAN tunnel.
VXLAN network identifier (VNI)	A VXLAN segment identifier similar to a VLAN ID. VMs on different VXLAN segments cannot communicate directly at Layer 2. A VNI identifies only one tenant. Even if multiple terminal users belong to the same VNI, they are considered one tenant. A VNI consists of 24 bits and supports a maximum of 16M tenants. In distributed VXLAN gateway scenarios, a VNI can be a Layer 2 or Layer 3 VNI. ● A Layer 2 VNI is mapped to a BD in 1:1 mode for intra-segment transmission of VXLAN packets. ● A Layer 3 VNI is bound to a VPN instance for inter-segment transmission of VXLAN packets.
Bridge domain (BD)	A Layer 2 broadcast domain through which VXLAN data packets are forwarded. VNIs identifying VNs must be mapped to BDs in 1:1 mode so that a BD can function as a VXLAN network entity to transmit VXLAN traffic.
VBDIF interface	A Layer 3 logical interface created for a BD. Configuring IP addresses for VBDIF interfaces allows communication between VXLANs on different network segments and between VXLANs and non-VXLANs and implements Layer 2 network access to a Layer 3 network.

Concept	Description
Virtual access point (VAP)	A VXLAN service access point. It can be a Layer 2 sub-interface or VLAN. ● If a Layer 2 sub-interface is used as a service access point, it can have different **encapsulation types** configured to transmit various types of data packets. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD. ● If a VLAN is used as a service access point, it can be bound to a BD for data packets in the VLAN to be transmitted through this BD.
Gateway	A device that ensures communication between VXLANs identified by different VNIs and between VXLANs and non-VXLANs. A VXLAN gateway can be a Layer 2 or Layer 3 gateway. ● Layer 2 gateway: allows tenants to access VXLANs and intra-segment communication on a VXLAN. ● Layer 3 gateway: allows inter-segment VXLAN communication and access to external networks.

Traffic Encapsulation Types

When a Layer 2 sub-interface is used as a service access point, different encapsulation types can be configured for the sub-interface to transmit various types of data packets. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD. **Table 3-2** describes the different encapsulation types.

Table 3-2 Traffic encapsulation types

Traffic Encapsulation Type	Description
dot1q	If a Dot1q sub-interface receives a single-tagged VLAN packet, the sub-interface forwards only the packet with a specifie VLAN ID. If a Dot1q sub-interface receives a double-tagged VLAN packet, the sub-interface forwards only the packet with a specified outer VLAN ID. ● When performing VXLAN encapsulation on packets, a Dot1q Layer 2 sub-interface removes the outer tags of the packets. ● When performing VXLAN decapsulation on packets, a Dot1q Layer 2 sub-interface replaces the VLAN tags with specified VLAN tags if the inner packets carry VLAN tags, or adds specified VLAN tags to the packets if the inner packets do not carry VXLAN tags. When setting the encapsulation type to **dot1q** for a Layer 2 sub-interface, note the following: ● The VLAN IDs specified for the Layer 2 sub-interface cannot be the same as either the VLAN IDs of packets allowed to pass through the corresponding Layer 2 interfaces or the MUX VLAN IDs. ● Layer 2 and Layer 3 sub-interfaces cannot have the same VLAN IDs specified.
untag	An untagged Layer 2 sub-interface receives only packets that do not carry VLAN tags. ● When performing VXLAN encapsulation on packets, an untagged Layer 2 sub-interface does not add any VLAN tag to the packets. ● When performing VXLAN decapsulation on packets, an untagged Layer 2 sub-interface removes the VLAN tags of single-tagged inner packets or the outer VLAN tags of double-tagged inner packets. When setting the encapsulation type to **untag** for a Layer 2 sub-interface, note the following: ● Ensure that the corresponding physical interface of the sub-interface does not have any configuration, and is removed from the default VLAN. ● Untagged Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● An interface can have only one untagged Layer 2 sub-interface configured.

Traffic Encapsulation Type	Description
qinq	A QinQ sub-interface receives only tagged packets with specified inner and outer VLAN tags. ● When performing VXLAN encapsulation on packets, a QinQ sub-interface removes two VLAN tags from packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintains the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. ● When performing VXLAN decapsulation on packets, a QinQ sub-interface adds two specific VLAN tags to packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintain the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. **NOTE** The traffic behavior for QinQ interfaces bound to the same BD must be the same. QinQ interfaces do not support DHCP Snooping or VBDIF and cannot be bound to the same BD as Dot1q sub-interfaces. A QinQ interface can have only one outer VLAN tag and one inner VLAN tag.
default	A default Layer 2 sub-interface receives all packets, irrespective of whether the packets carry VLAN tags. When performing VXLAN encapsulation and decapsulation on packets, a default Layer 2 sub-interface does not process VLAN tags of the packets. When setting the encapsulation type to **default** for a Layer 2 sub-interface, note the following: ● Ensure that the interface for the Layer 2 sub-interface is not added to any VLAN. ● Default Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● If a default Layer 2 sub-interface is created for an interface, the interface cannot have other types of Layer 2 sub-interfaces configured.

NOTE

When a sub-interface that is configured with dot1q and QinQ receives double-tagged VLAN packets, the QinQ sub-interface preferentially processes the packets. For example, if a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of 20, the QinQ sub-interface preferentially processes the packet. If a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of non-20, the dot1q sub-interface preferentially processes the packet.

3.2 Gateway Classification

A device that ensures communication between VXLANs identified by different VNIs and between VXLANs and non-VXLANs.

A VXLAN gateway can be a Layer 2 or Layer 3 gateway.

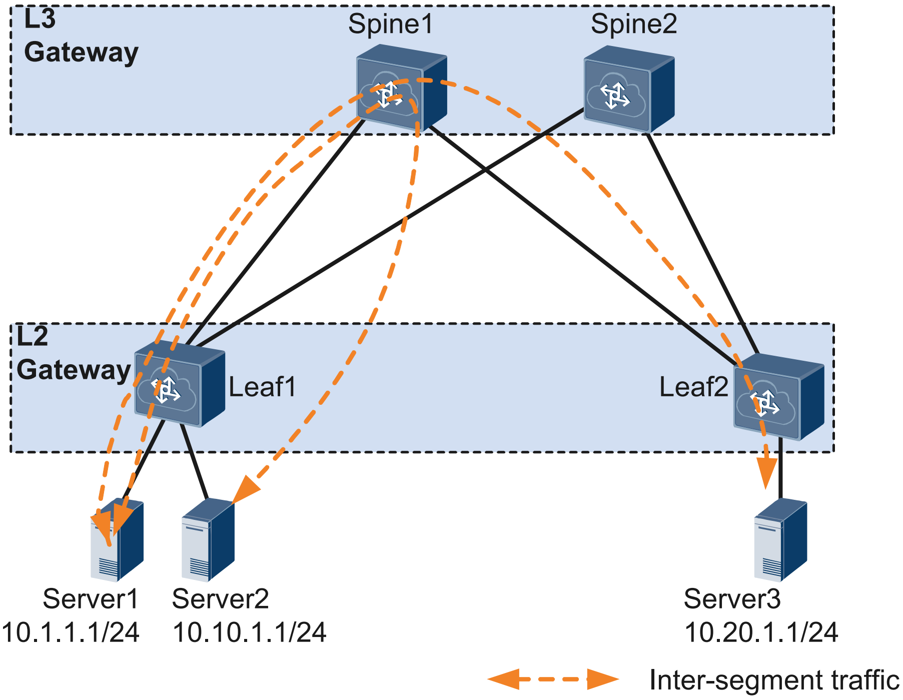
- Layer 2 gateway: allows tenants to access VXLANs and intra-segment communication on a VXLAN.
- Layer 3 gateway: allows inter-segment VXLAN communication and access to external networks.

VXLAN Layer 3 gateways can be deployed in centralized or distributed mode.

Centralized VXLAN Gateway Mode

In this mode, Layer 3 gateways are configured on one device. On the network shown in **Figure 3-2**, traffic across network segments is forwarded through Layer 3 gateways to implement centralized traffic management.

Figure 3-2 Centralized VXLAN gateway networking



Centralized VXLAN gateway deployment has its advantages and disadvantages.

- Advantage: Inter-segment traffic can be centrally managed, and gateway deployment and management is easy.
- Disadvantages:
  - Forwarding paths are not optimal. Inter-segment Layer 3 traffic of data centers connected to the same Layer 2 gateway must be transmitted to the centralized Layer 3 gateway for forwarding.

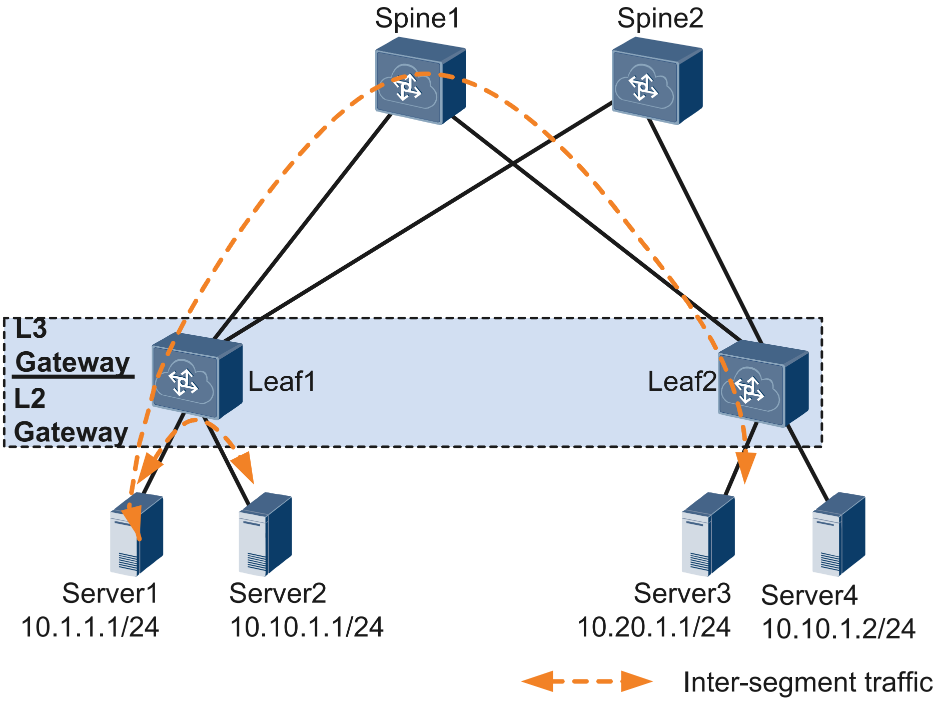
- The ARP entry specification is a bottleneck. ARP entries must be generated for tenants on the Layer 3 gateway. However, only a limited number of ARP entries are allowed by the Layer 3 gateway, impeding data center network expansion.

Distributed VXLAN Gateway Mode

- **Background**

  Deploying distributed VXLAN gateways addresses problems that occur in centralized VXLAN gateway networking. Distributed VXLAN gateways use the spine-leaf network. In this networking, leaf nodes, which can function as Layer 3 VXLAN gateways, are used as VTEPs to establish VXLAN tunnels. Spine nodes are unaware of the VXLAN tunnels and only forward VXLAN packets between different leaf nodes. On the network shown in **Figure 3-3**, Server 1 and Server 2 on different network segments both connect to Leaf 1. When Server 1 and Server 2 communicate, traffic is forwarded only through Leaf 1, not through any spine node.

  Figure 3-3 Distributed VXLAN gateway networking



  A spine node supports high-speed IP forwarding capabilities.

  A leaf node can:

  - Function as a Layer 2 VXLAN gateway to connect to physical servers or VMs and allow tenants to access VXLANs.
  - Function as a Layer 3 VXLAN gateway to perform VXLAN encapsulation and decapsulation to allow inter-segment VXLAN communication and access to external networks.

- **Characteristics of distributed VLAN gateways**

  Distributed VXLAN gateway networking has the following characteristics:

  - Flexible deployment. A leaf node can function as both Layer 2 and Layer 3 VXLAN gateways.

– Improved network expansion capabilities. A leaf node only needs to learn the ARP entries of servers attached to it. A centralized Layer 3 gateway in the same scenario, however, has to learn the ARP entries of all servers on the network. Therefore, the ARP entry specification is no longer a bottleneck on a distributed VXLAN gateway.

3.3 VXLAN Packet Format

VXLAN is a network virtualization technique that uses MAC-in-UDP encapsulation by adding a UDP header and a VXLAN header before an original Ethernet packet. **Figure 3-4** shows the VXLAN packet format.

Figure 3-4 VXLAN packet format

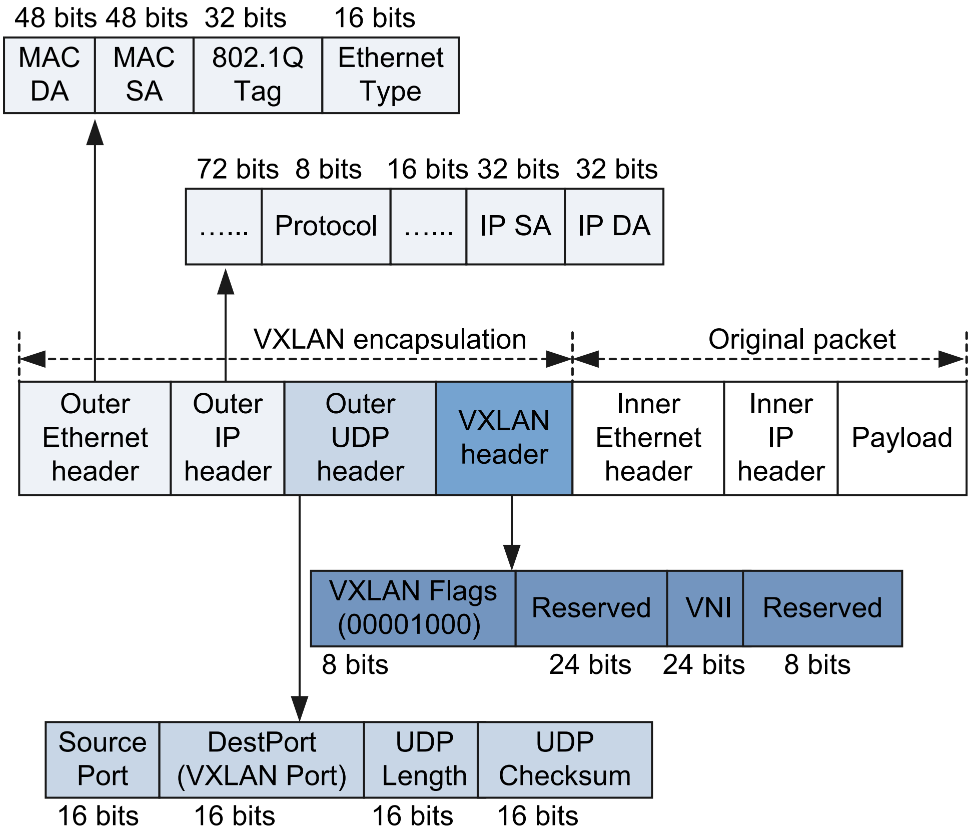


Table 3-3 Fields in the VXLAN packet format

Field	Description
VXLAN header	● VXLAN Flags (8 bits): The value is 00001000. ● VNI (24 bits): VXLAN Segment ID or VXLAN Network Identifier used to identify a VXLAN segment. ● Reserved fields (24 bits and 8 bits): must be set to 0.

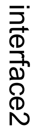
Field	Description
Outer UDP header	● DestPort: destination port number, which is 4789 for UDP. ● Source Port: source port number, which is calculated by performing the hash operation on the inner Ethernet frame headers.
Outer IP header	● IP SA: source IP address, which is the IP address of the local VTEP of a VXLAN tunnel. ● IP DA: destination IP address, which is the IP address of the remote VTEP of a VXLAN tunnel.
Outer Ethernet header	● MAC DA: destination MAC address, which is the MAC address mapped to the next-hop IP address based on the destination VTEP address in the routing table of the VTEP on which the VM that sends packets resides. ● MAC SA: source MAC address, which is the MAC address of the VTEP on which the VM that sends packet resides. ● 802.1Q Tag: VLAN tag carried in packets. This field is optional. ● Ethernet Type: Ethernet packet type. The value is 0x0800 in IP packets.

3.4 Tunnel Establishment (Static Mode)

A VXLAN tunnel is identified by a pair of VTEP IP addresses. A VXLAN tunnel is statically created after you configure local and remote VNIs, VTEP IP addresses, and an ingress replication list, and the tunnel goes Up when the pair of VTEPs are reachable at Layer 3.

On the network shown in **Figure 3-5**, Server 1 and Server 3 are deployed for Device 1, and Server 2 is deployed for Device 3. Server 1 and Server 2 reside on different network segments, whereas Server 2 and Server 3 reside on the same network segment. To allow VMs on Server 2 and Server 3 to communicate, VNIs and VTEP IP addresses must be configured for establishing a VXLAN tunnel between Device 1 and Device 3. To allow VMs on Server 1 and Server 2 to communicate, VNIs and VTEP IP addresses must be configured for establishing a VXLAN tunnel between Device 1 and Device 2 and between Device 2 and Device 3.

Figure 3-5 VXLAN networking



3.5 Tunnel Establishment (MP-BGP Dynamic Mode)

Multi-protocol Extensions for Border Gateway Protocol (MP-BGP) is applicable to the distributed gateway scenario to implement dynamic establishment and management of VXLAN tunnels and learning of host routes.

In a distributed VXLAN gateway scenario where MP-BGP runs on each leaf node, inter-segment communication requires leaf nodes to learn user-side ARP entries, generate host routes from these entries, and advertise these routes to other BGP peers. BGP peers then dynamically establish VXLAN tunnels based on their local VTEP IP addresses and the VTEP IP addresses carried in the remote-nexthop attributes of received BGP protocol packets. The MP-BGP control plane implementation process is as follows:

1. **VXLAN tunnel management**

   On the network shown in **Figure 3-6**, a leaf node assigns a Layer 2 VNI to each segment and a Layer 3 VNI to each tenant identified by a VPN instance. When Leaf 1 forwards traffic at Layer 3, Leaf 1 transmits tenants' VNI IDs to Leaf 2 through the VXLAN tunnel. Leaf 2 identifies VPNs based on tenants' VNI IDs to determine whether tenants belong to the same VPN.

   This example illustrates VXLAN tunnel management for distributed VXLAN gateways using Server 1 accessing other servers on a different segment.

   a. MP-BGP dynamically manages the VXLAN tunnel between Leaf 1 and Leaf 2. After Leaf 1 learns Server 1's host routes generated based on ARP entries, Leaf 1 uses BGP to advertise Server 1's host routes to Leaf 2 through the remote-nexthop attribute.

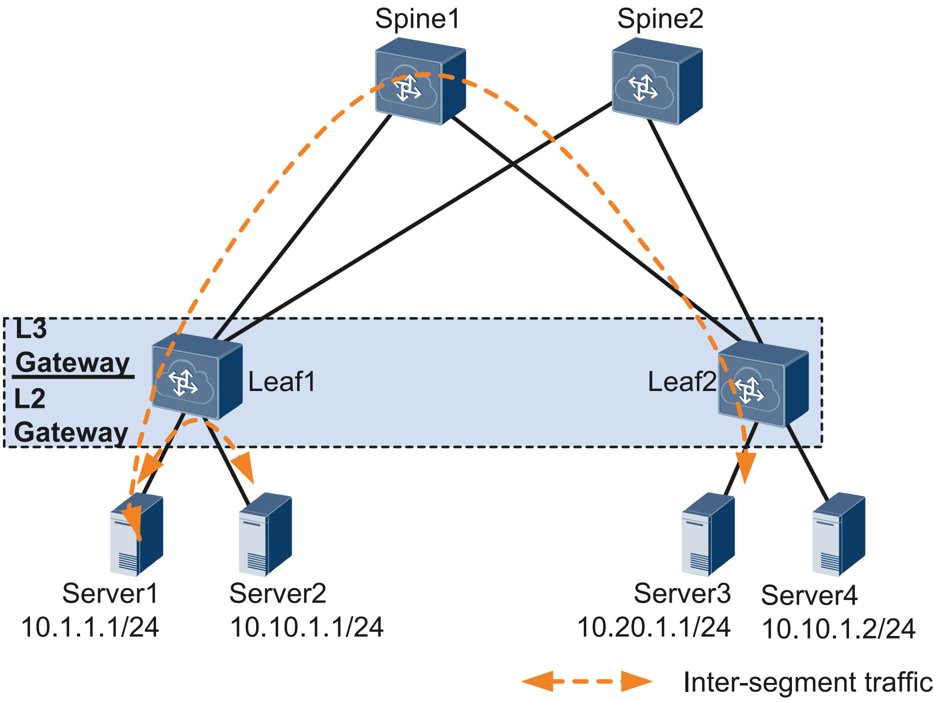
   b. Upon receipt of Server 1's host routes, Leaf 2 checks whether the next hop addresses of these routes are the same as Leaf 2's VTEP IP address. If they are the same, Leaf 2 dynamically establishes a VXLAN tunnel based on Leaf 1's VTEP IP address and Leaf 2's VTEP IP address.

   c. After Leaf 1 receives packets from Server 1 to Server 3, Leaf 1 searches the host routing table and finds the VXLAN tunnel for Server 3. Leaf 1 then encapsulates the packets from Server 1 into VXLAN packets and sends the VXLAN packets to Leaf 2 connected to Server 3. Leaf 2 then decapsulates the VXLAN packets and forwards the packets to Server 3.

   A VXLAN tunnel between two distributed gateways is identified by a pair of VTEP IP addresses. If multiple host routes carry the same source VTEP IP address and the same next-hop VTEP IP address, only one VXLAN tunnel can be established.

   When Leaf 1 no longer has any host attached, Leaf 1 will withdraw all host routes and notify its BGP peers of the withdrawal. When Leaf 2 finds no reachable host routes on Leaf 1, Leaf 2 will dynamically delete the VXLAN tunnel established with Leaf 1.

Figure 3-6 VXLAN tunnel management



2. **ARP learning**

   The following uses the network shown in **Figure 3-7** as an example to describe how Leaf 1 learns Server 1's ARP entry.

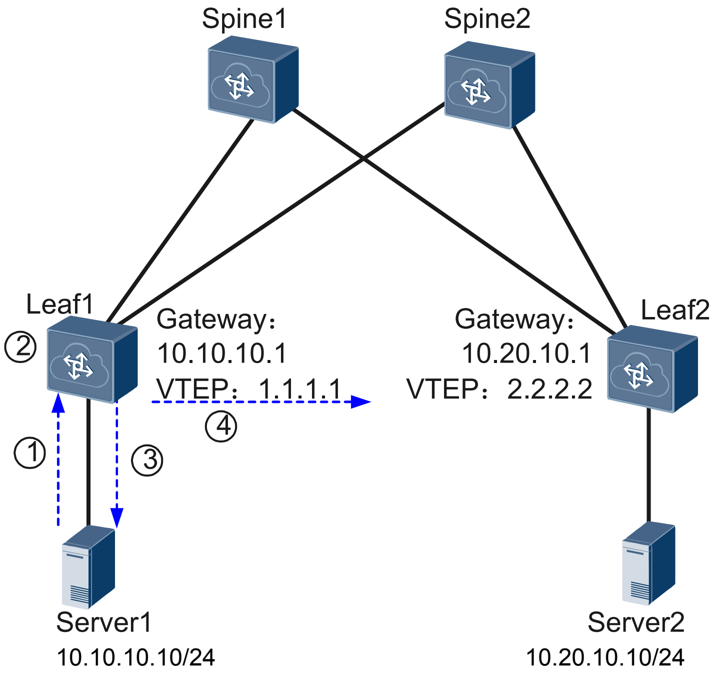
   a. Server 1 sends an ARP request.

   b. Upon receipt of the packet, Leaf 1 broadcasts the packet in the Layer 2 broadcast domain and learns Server 1's ARP entry.

   c. Leaf 1 responds to Server 1's ARP request.

   In a distributed VXLAN gateway scenario, leaf nodes learn only user-side ARP entries, not VXLAN tunnel-side ARP entries. In addition, only the leaf node connected to a host can advertise routes of the host, and the other leaf nodes cannot advertise the routes of this host. This is to prevent network traffic destined for a host from being imported to other leaf nodes.

Figure 3-7 ARP learning



① Server1 sends an ARP request message.

② Leaf1 receives the ARP request message and broadcasts it on the Layer 2 network, and learns the ARP entry of Server1.

③ Leaf1 responds to the ARP request message sent by Server1.

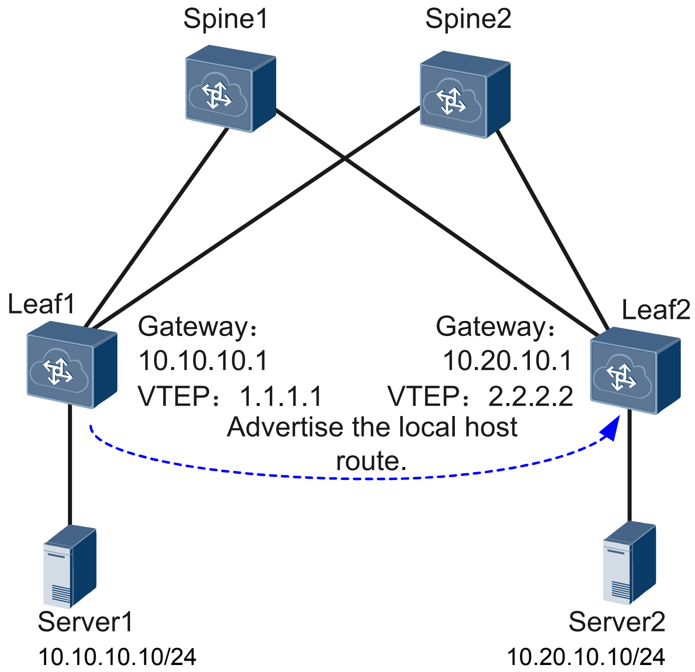
④ Leaf2 receives the ARP request message sent by Server1 from the VXLAN tunnel, but does not learn the ARP entry of Server1.

3. **Advertisement of local host routes generated based on ARP entries**

   On the network shown in **Figure 3-8**, the leaf nodes function as Layer 3 VXLAN gateways to learn tenants' ARP entries and generate host routes based on these ARP entries. The leaf nodes then use BGP to advertise the host routes to BGP peers. The next hops of these host routes are the VTEP IP addresses of peer leaf nodes' VTEP IP addresses.

   On a VXLAN, only edge nodes can learn host routes.

Figure 3-8 Advertisement of local host routes generated based on ARP entries



4. **Route priority control**

   In some scenarios, leaf nodes can use BGP to control the priority of host routes to be advertised to BGP peers for special purposes.

   – Active-active gateway access

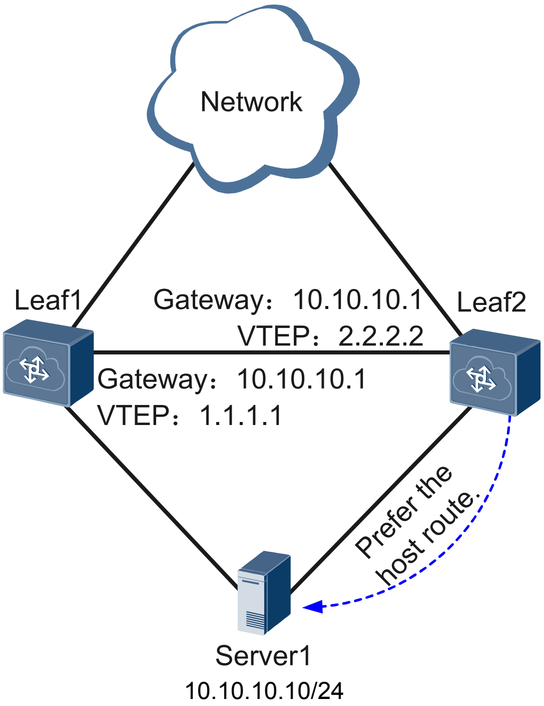
     On the network shown in **Figure 3-9**, Leaf 2's routing table will have two routes destined for Server 1.

     ■ One route is a direct host route.

     ■ The other route is a BGP route advertised to Leaf 2.

     Leaf 2 must be configured to prefer the direct route to prevent a traffic bypass.

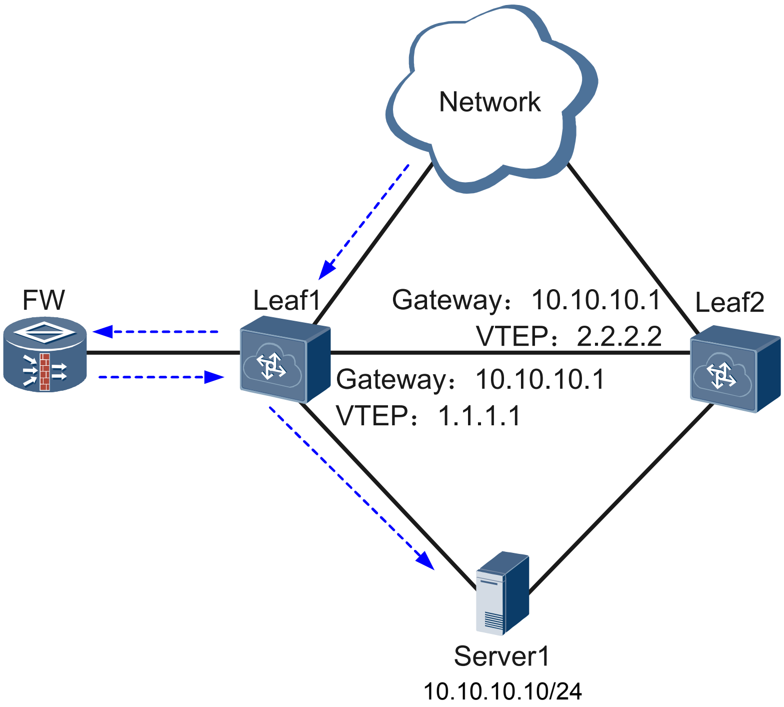
Figure 3-9 Host route priority control



– Traffic distribution on a service node

On the network shown in **Figure 3-10**, a service node (FW) is used for traffic distribution. External traffic destined for Server 1 will go along Leaf 1 -> FW -> Leaf 1 before reaching Server 1. In this scenario, the host route priority must be lower than the priority of the route obtained using PBR for traffic distribution.

Figure 3-10 Traffic distribution on a service node



3.6 Tunnel Establishment (BGP EVPN Dynamic Mode)

Ethernet Virtual Private Network Border Gateway Protocol (BGP EVPN) is applicable to the distributed or centralized gateway scenario to implement dynamic establishment and management of VXLAN tunnels and learning of host routes.

BGP EVPN for Centralized VXLAN Gateways

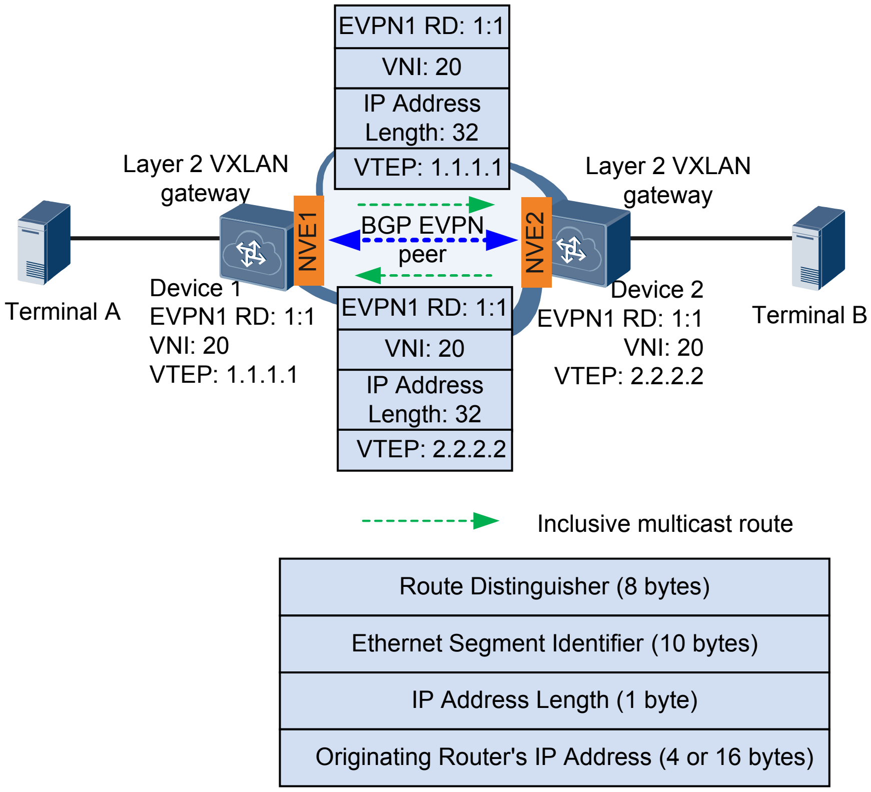
BGP EVPN extends BGP by defining a new type of network layer reachability information (NLRI) called EVPN NLRI. In a centralized VXLAN gateway scenario, EVPN can function as the VXLAN control plane by using inclusive multicast routes carried in the EVPN NLRI to exchange information between VXLAN gateways for tunnel establishment. A local gateway must obtain the VTEP IP address and VNI of a remote gateway before establishing a VXLAN tunnel with the remote gateway. On the network shown in **Figure 3-11**, Device 1 and Device 2 use EVPN inclusive multicast routes to exchange information for VXLAN tunnel establishment. The process is as follows:

1. Create EVPN instances on Device 1 and Device 2 and establish an BGP EVPN peer relationship between Device 1 and Device 2.
2. Device 1 and Device 2 use BGP EVPN to send EVPN routes comprised of inclusive multicast route prefixes and PMSI attributes. VTEP IP addresses are stored in the Originating Router's IP Address field in the inclusive multicast route prefix, and Layer 2 VNIs are stored in PMSI attributes.
3. Upon receipt of the EVPN routes, a gateway matches the export VPN target carried in the route against the import VPN target of its local EVPN instance. If the two VPN targets match, the gateway accepts the route and stores the VTEP IP address and VNI carried in the route for later packet transmission over the VXLAN tunnel. If the two VPN targets do not match, the gateway drops the route.

NOTE

In this example, the import VPN target of one EVPN instance must match the export VPN target of the other EVPN instance. Otherwise, the VXLAN tunnel cannot be established. If only one end can successfully accept the IRB or IP prefix route, this end can establish a VXLAN tunnel to the other end, but cannot exchange data packets with the other end. The other end drops packets after confirming that there is no VXLAN tunnel to the end that has sent these packets.

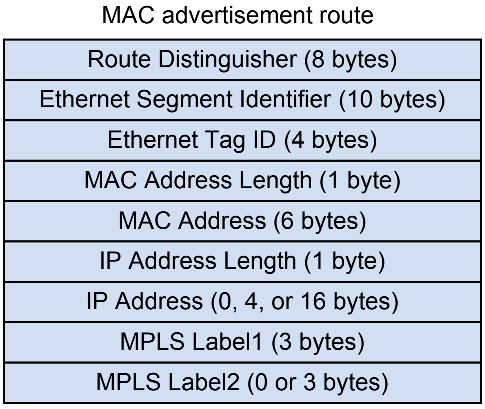
Figure 3-11 VXLAN tunnel establishment using EVPN in centralized gateway scenarios



Route Distinguisher (8 bytes)
Ethernet Segment Identifier (10 bytes)
IP Address Length (1 byte)
Originating Router's IP Address (4 or 16 bytes)

To implement ARP broadcast suppression in centralized VXLAN gateway scenarios, VXLAN gateways can be configured to advertise ARP routes. As shown in **Figure 3-12**, ARP routes are comprised of EVPN MAC advertisement route prefixes and extended community attributes, with MAC addresses of user terminals connecting to the gateways stored in the MAC Address and MAC Address Length fields, IP Address, and IP Address Length of packets. EVPN enables gateways to use ARP to learn the local MAC addresses and use ARP routes to learn remote MAC addresses and IP addresses corresponding to these MAC addresses, and store them locally. After ARP broadcast suppression is enabled and a gateway receives an ARP request message, the gateway first searches the locally stored MAC addresses. If a matching MAC address is found, the gateway responds with an ARP reply message without broadcasting the ARP request message to other gateways. This processing reduces network resource usage.

Figure 3-12 EVPN NLRI specific to the MAC advertisement route



MAC advertisement route
Route Distinguisher (8 bytes)
Ethernet Segment Identifier (10 bytes)
Ethernet Tag ID (4 bytes)
MAC Address Length (1 byte)
MAC Address (6 bytes)
IP Address Length (1 byte)
IP Address (0, 4, or 16 bytes)
MPLS Label1 (3 bytes)
MPLS Label2 (0 or 3 bytes)

BGP EVPN for Distributed VXLAN Gateways

BGP EVPN defines a new type of BGP network layer reachability information (NLRI), called EVPN NLRI. In a distributed VXLAN gateway scenario, EVPN serves as the VXLAN control plane. It uses MAC advertisement route prefixes and IP prefix route prefixes carried in EVPN NLRI as well as extended community attributes to transmit information required for VXLAN tunnel establishment. **Figure 3-13** illustrates the formats of a MAC advertisement route prefix, an IP prefix route prefix, and an extended community attribute. The MAC advertisement route prefix, IP prefix route prefix, and extended community attribute can form different types of routes. **Table 3-4** compares the two types of routes used in a distributed VXLAN gateway scenario.

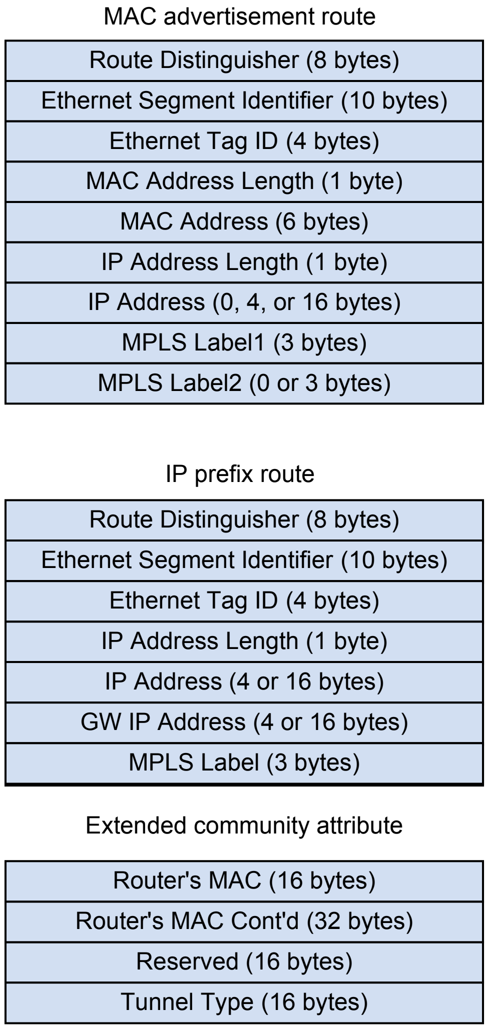
Table 3-4 Comparison of different types of routes

Route Type	Description	Characteristic
IRB route	An IRB route consists of a MAC advertisement roue prefix and an extended community attribute. ● The Layer 2 VNI and Layer 3 VNI are stored in MPLS Label 1 and MPLS Label 2, respectively. ● The host route is stored in the IP Address and IP Address Length fields of the MAC advertisement route prefix. ● The VTEP IP address is stored in the Next Hop attribute. ● The VTEP MAC address and tunnel type are stored in the Router's MAC and Tunnel Type fields of the extended community attribute, respectively.	VXLAN gateways advertise IRB routes to suppress ARP flooding and exchange host routes.

Route Type	Description	Characteristic
IP prefix route	Layer 3 VNI An IP prefix route consists of an IP prefix route prefix and an extended community attribute. ● An IP prefix route stores the Layer 3 VNI in MPLS Label 2, but does not carry the Layer 2 VNI. ● The host route is stored in the IP Address and IP Address Length fields of the IP prefix route prefix. ● The VTEP IP address is stored in the Next Hop attribute. ● The VTEP MAC address and tunnel type are stored in the Router's MAC and Tunnel Type fields of the extended community attribute, respectively.	VXLAN gateways advertise IP prefix routes to exchange host routes or host network segment routes. If many host routes belong to the same network segment, configure VXLAN gateways to advertise host network segment routes using IP prefix routes. Only one host network segment route is to be advertised for host routes belonging to the same network segment. **NOTE** A VXLAN gateway can advertise network segment routes only if the network segments attached to the gateway are unique network-wide.

A host route or host network segment route is stored in the IP Address and IP Address Length fields of a MAC advertisement route prefix or IP prefix route prefix.

Figure 3-13 Formats of the MAC advertisement route and IP prefix route



MAC advertisement route

Route Distinguisher (8 bytes)
Ethernet Segment Identifier (10 bytes)
Ethernet Tag ID (4 bytes)
MAC Address Length (1 byte)
MAC Address (6 bytes)
IP Address Length (1 byte)
IP Address (0, 4, or 16 bytes)
MPLS Label1 (3 bytes)
MPLS Label2 (0 or 3 bytes)

IP prefix route

Route Distinguisher (8 bytes)
Ethernet Segment Identifier (10 bytes)
Ethernet Tag ID (4 bytes)
IP Address Length (1 byte)
IP Address (4 or 16 bytes)
GW IP Address (4 or 16 bytes)
MPLS Label (3 bytes)

Extended community attribute

Router's MAC (16 bytes)
Router's MAC Cont'd (32 bytes)
Reserved (16 bytes)
Tunnel Type (16 bytes)

Figure 3-14 illustrates the process of automatically establishing a VXLAN tunnel between two distributed VXLAN gateways.

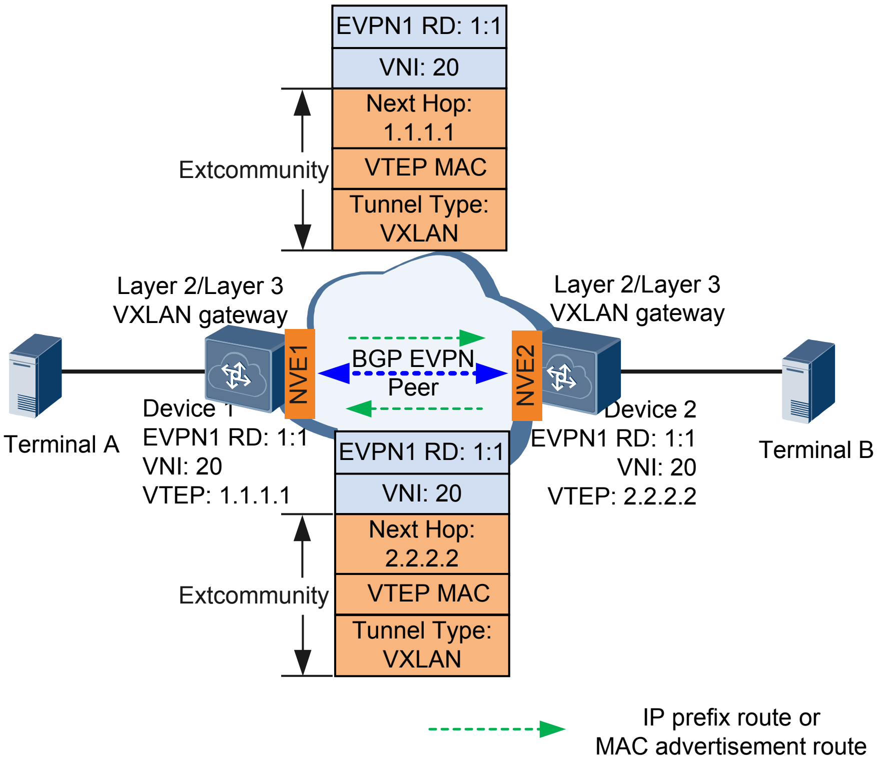
1. Create an EVPN instance and a VPN instance on each VXLAN gateway (Device 1 and Device 2 in this example) and establish an BGP EVPN peer relationship between the two devices.
2. Device 1 and Device 2 use BGP EVPN to exchange IRB or IP prefix routes.
   - Upon receipt of an ARP request from a terminal, a gateway obtains the host's ARP entry and generates a MAC advertisement route from this entry. Then, the gateway advertises this route as an IRB route to the other gateway.
   - A gateway imports the routes destined for the host address or host network segment address to its local VPN instance. Then, the gateway imports these routes from the local VPN instance to the local EVPN instance and advertises these routes to the other gateway using an IP prefix route.
3. Upon receipt of the IRB or IP prefix route, a gateway matches the export VPN target carried in the route against the import VPN target of its local EVPN instance. If the two VPN targets match, the gateway accepts the route and stores the VTEP IP address and

VNI carried in the route for later packet transmission over the VXLAN tunnel. If the two VPN targets do not match, Device 2 drops the route.

NOTE

In this example, the import VPN target of one EVPN instance must match the export VPN target of the other EVPN instance. Otherwise, the VXLAN tunnel cannot be established. If only one end can successfully accept the IRB or IP prefix route, this end can establish a VXLAN tunnel to the other end, but cannot exchange data packets with the other end. The other end drops packets after confirming that there is no VXLAN tunnel to the end that has sent these packets.

Figure 3-14 Establishing a VXLAN tunnel between two distributed VXLAN gateways using the EVPN control plane



3.7 Data Packet Forwarding

Layer 2 packets can be transmitted over legacy Layer 3 networks after being encapsulated into VXLAN packets. This implementation allows you to construct a logic large Layer 2 network over a Layer 3 network.

Intra-Segment Packet Forwarding

The intra-segment packet forwarding process is the same in both centralized and distributed VXLAN gateway scenarios, covering BUM packet forwarding and known unicast forwarding.

- **BUM packet forwarding process**

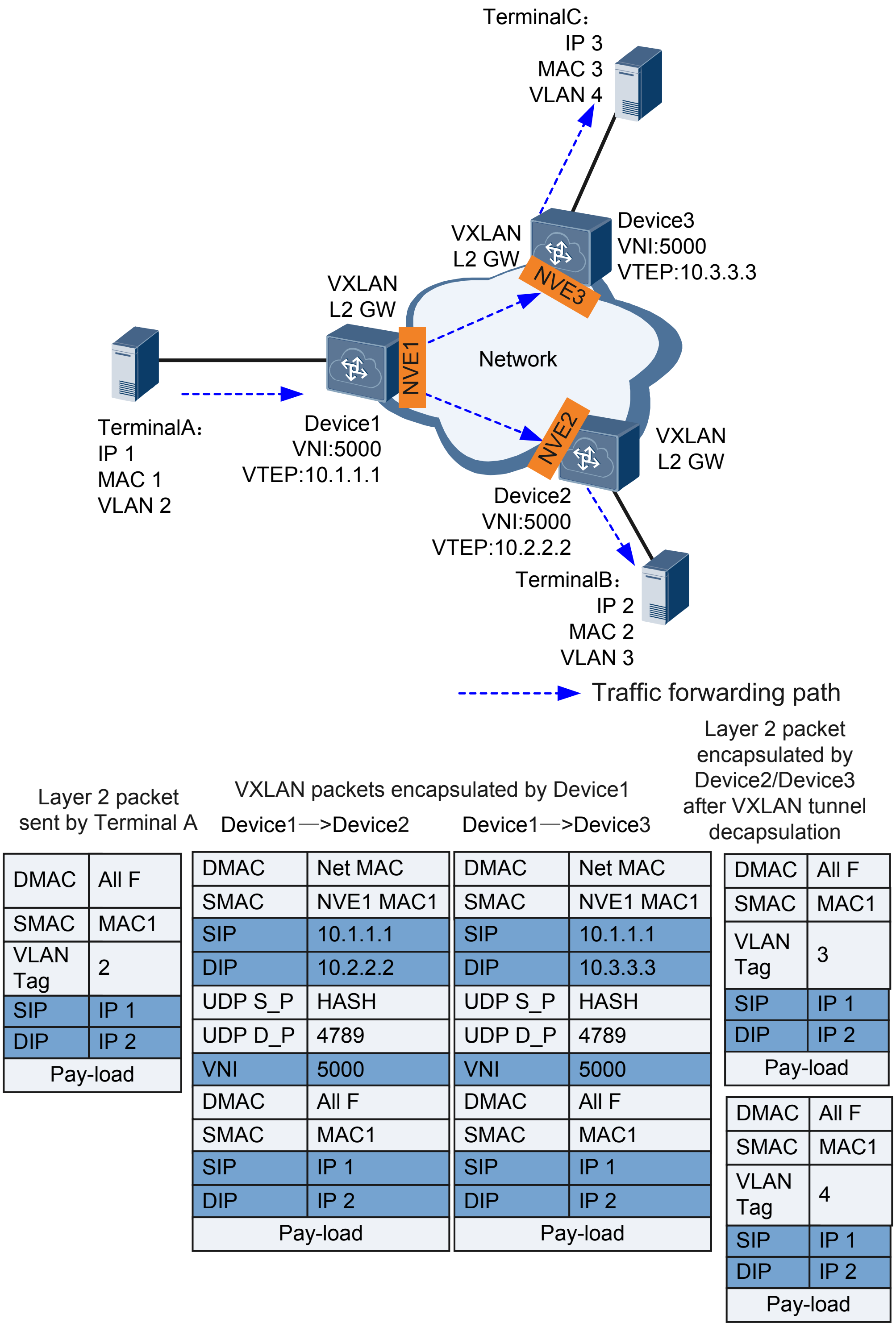
  When a BUM packet enters a VXLAN tunnel, the ingress VTEP uses ingress replication to perform VXLAN tunnel encapsulation. When the BUM packet leaves the VXLAN tunnel, the egress VTEP decapsulates the BUM packet. **Figure 3-15** shows the BUM packet forwarding process.

NOTE

Ingress replication: After an NVE receives broadcast, unknown unicast, and multicast (BUM) packets, the local VTEP obtains a list of VTEPs on the same VXLAN segment as itself through the control plane and sends a copy of the BUM packets to every VTEP in the list.

Ingress replication allows BUM packets to be transmitted in broadcast mode, independent of multicast routing protocols.

Figure 3-15 BUM packet forwarding process



Layer 2 packet sent by Terminal A

DMAC	All F
SMAC	MAC1
VLAN Tag	2
SIP	IP 1
DIP	IP 2
Pay-load	

VXLAN packets encapsulated by Device1

Device1—>Device2

DMAC	Net MAC
SMAC	NVE1 MAC1
SIP	10.1.1.1
DIP	10.2.2.2
UDP S_P	HASH
UDP D_P	4789
VNI	5000
DMAC	All F
SMAC	MAC1
SIP	IP 1
DIP	IP 2
Pay-load	

Device1—>Device3

DMAC	Net MAC
SMAC	NVE1 MAC1
SIP	10.1.1.1
DIP	10.3.3.3
UDP S_P	HASH
UDP D_P	4789
VNI	5000
DMAC	All F
SMAC	MAC1
SIP	IP 1
DIP	IP 2
Pay-load	

Layer 2 packet encapsulated by Device2/Device3 after VXLAN tunnel decapsulation

DMAC	All F
SMAC	MAC1
VLAN Tag	3
SIP	IP 1
DIP	IP 2
Pay-load	

DMAC	All F
SMAC	MAC1
VLAN Tag	4
SIP	IP 1
DIP	IP 2
Pay-load	

a. After Device 1 receives packets from Terminal A, Device 1 determines the Layer 2 broadcast domain of the packets based on the access interface and VLAN information carried in the packets and checks whether the destination MAC address is a BUM address.

- If the destination MAC address is a BUM address, Device 1 broadcasts the packets in the Layer 2 broadcast domain and goes to **2**.

  - ■ If the destination MAC address is not a BUM address, Device 1 follows the **unicast packet forwarding process**.

b. Device 1's VTEP obtains the ingress replication list for the VNI, replicates packets based on the list, and performs VXLAN tunnel encapsulation by adding outer headers. Device 1 then forwards the packets through the outbound interface.

c. Upon receipt of the VXLAN packets, the VTEP on Device 2 or Device 3 verifies the VXLAN packets based on the UDP destination port numbers, source and destination IP addresses, and VNI. The VTEP obtains the Layer 2 broadcast domain based on the VNI and removes the outer headers to obtain the inner Layer 2 packets. It then determines whether the destination MAC address is a BUM address.

  - ■ If the destination MAC address is a BUM address, the VTEP broadcasts the packets to the user side in the Layer 2 broadcast domain.
  - ■ If the destination MAC address is not a BUM address, the VTEP further checks whether it is a local MAC address.
    - ○ If it is a local MAC address, the VTEP sends the packets to the device.
    - ○ If it is not a local MAC address, the VTEP searches for the outbound interface and encapsulation information in the Layer 2 broadcast domain and goes to **4**.

d. Device 2 or Device 3 adds VLAN tags to the packets based on the outbound interface and encapsulation information and forwards the packets to Terminal B or Terminal C.

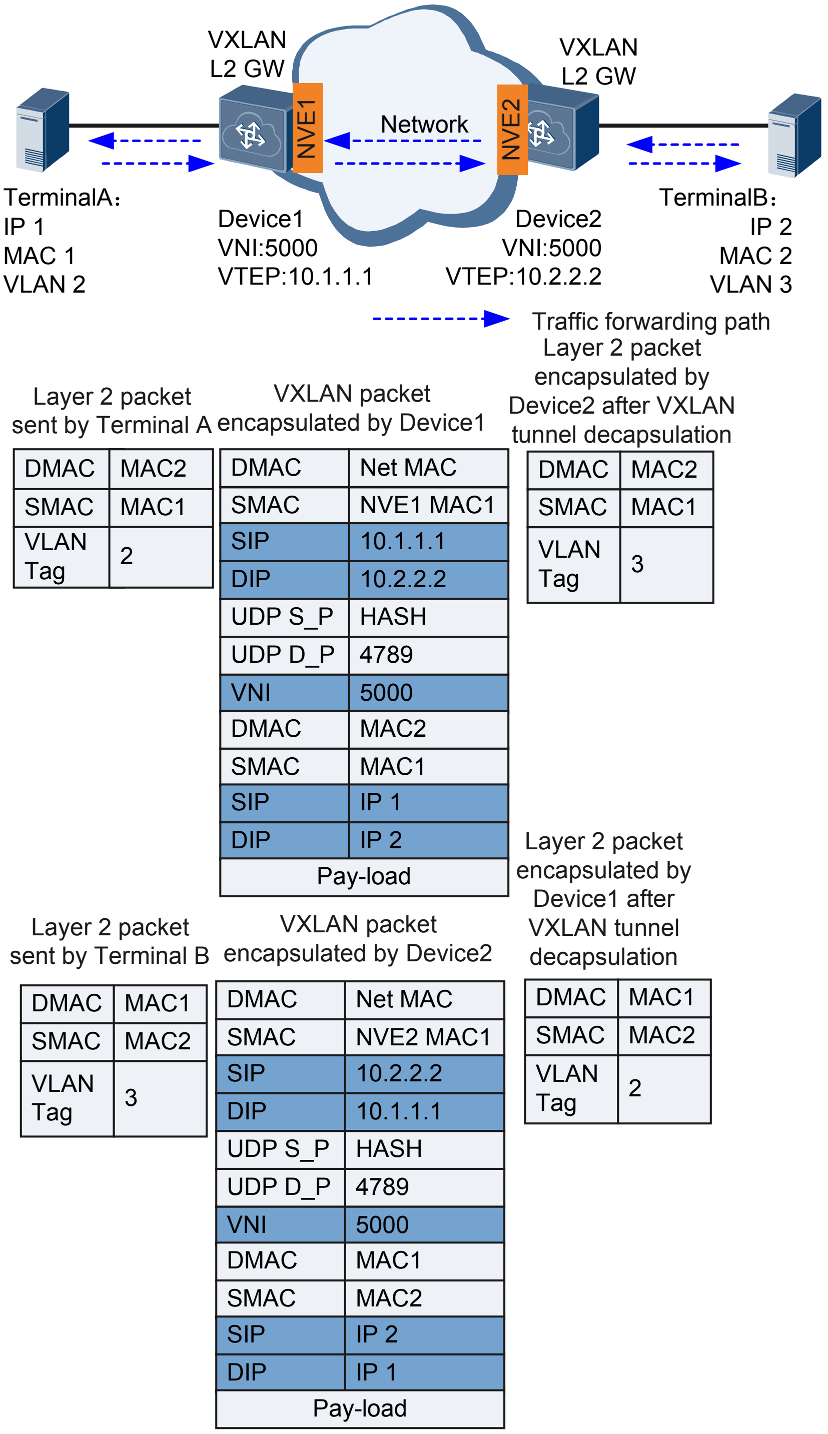
NOTE

Terminal B or Terminal C responds to Terminal A following the **unicast packet forwarding process**.

- **Unicast Packet Forwarding Process**

  Figure 3-16 shows the unicast packet forwarding process.

Figure 3-16 Unicast packet forwarding process



Layer 2 packet sent by Terminal A

DMAC	MAC2
SMAC	MAC1
VLAN Tag	2

VXLAN packet encapsulated by Device1

DMAC	Net MAC
SMAC	NVE1 MAC1
SIP	10.1.1.1
DIP	10.2.2.2
UDP S_P	HASH
UDP D_P	4789
VNI	5000
DMAC	MAC2
SMAC	MAC1
SIP	IP 1
DIP	IP 2
Pay-load	

Layer 2 packet encapsulated by Device2 after VXLAN tunnel decapsulation

DMAC	MAC2
SMAC	MAC1
VLAN Tag	3

Layer 2 packet sent by Terminal B

DMAC	MAC1
SMAC	MAC2
VLAN Tag	3

VXLAN packet encapsulated by Device2

DMAC	Net MAC
SMAC	NVE2 MAC1
SIP	10.2.2.2
DIP	10.1.1.1
UDP S_P	HASH
UDP D_P	4789
VNI	5000
DMAC	MAC1
SMAC	MAC2
SIP	IP 2
DIP	IP 1
Pay-load	

Layer 2 packet encapsulated by Device1 after VXLAN tunnel decapsulation

DMAC	MAC1
SMAC	MAC2
VLAN Tag	2

a. After Device 1 receives packets from Terminal A, Device 1 determines the Layer 2 broadcast domain of the packets based on the access interface and VLAN information carried in the packets and checks whether the destination MAC address is a unicast address.

- If the destination MAC address is a unicast address, Device 1 further checks whether it is a local MAC address.
  - If it is a local MAC address, Device 1 processes the packets.

        - ○ If it is not a local MAC address, Device 1 searches for the outbound interface and encapsulation information in the Layer 2 broadcast domain and goes to **2**.
    - ■ If the destination MAC address is not a unicast address, the VTEP follows the **2**.

b. Device 1's VTEP performs VXLAN tunnel encapsulation based on the outbound interface and encapsulation information and forwards the packets.

c. Upon receipt of the VXLAN packets, Device 2's VTEP verifies the VXLAN packets based on the UDP destination port numbers, source and destination IP addresses, and VNI. Device 2 obtains the Layer 2 broadcast domain based on the VNI and performs VXLAN tunnel decapsulation to obtain the inner Layer 2 packets. It then determines whether the destination MAC address is a unicast address.
    - ■ If the destination MAC address is a unicast address, the VTEP searches for the outbound interface and encapsulation information in the Layer 2 broadcast domain and goes to **4**.
    - ■ If the destination MAC address is not a unicast address, the VTEP further checks whether it is a local MAC address.
        - ○ If it is a local MAC address, the VTEP sends the packets to Device 2.
        - ○ If it is not a local MAC address, the VTEP follows the **BUM packet forwarding process**.

d. Device 2 adds VLAN tags to the packets based on the outbound interface and encapsulation information and forwards the packets to Terminal B.

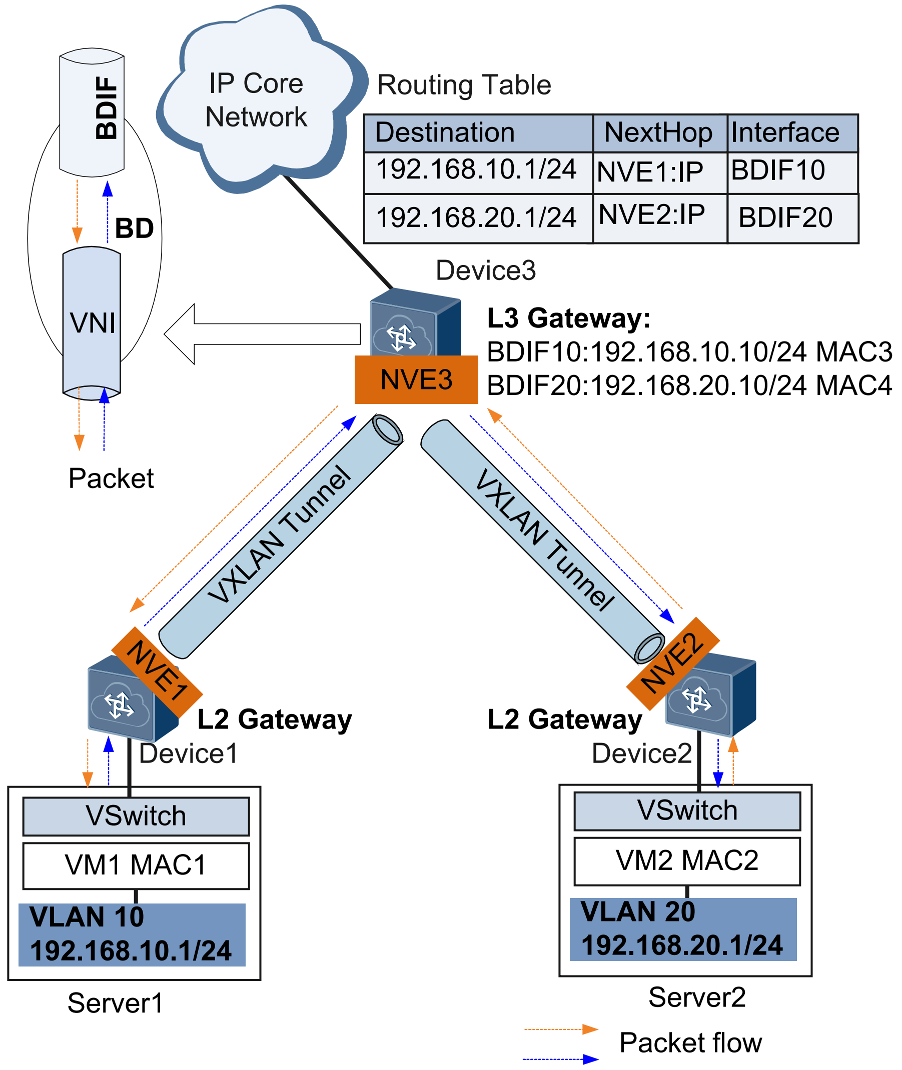
Inter-Segment Packet Forwarding

The inter-segment packet forwarding process varies in centralized and distributed VXLAN gateway scenarios.

- **Centralized VXLAN gateway**

  VNIs can be mapped to BDs in 1:1 mode so that a BD can function as a VXLAN network entity to transmit VXLAN data packets. Layer 3 VBDIF interfaces can be configured for BDs. VBDIF interfaces have similar functions as VLANIF interfaces. Configuring IP addresses for VBDIF interfaces allows communication between VXLANs on different network segments and between VXLANs and non-VXLANs, implementing VXLAN Layer 3 gateway functionality.

Figure 3-17 Inter-segment packet forwarding in a centralized VXLAN gateway scenario



Routing Table

Destination	NextHop	Interface
192.168.10.1/24	NVE1:IP	BDIF10
192.168.20.1/24	NVE2:IP	BDIF20

In a centralized VXLAN gateway scenario shown in **Figure 3-17**, the inter-segment packet forwarding process is as follows:

a. After Device 3 receives VXLAN packets, it decapsulates the packets and checks whether the destination MAC address in the inner packets is the MAC address of the Layer 3 gateway interface VBDIF10.

  - If the destination MAC address is a local MAC address, Device 3 forwards the packets to the Layer 3 gateway on the destination network segment and goes to **2**.
  - If the destination MAC address is not a local MAC address, Device 3 searches for the outbound interface and encapsulation information in the Layer 2 broadcast domain.

b. Device 3 remove the Ethernet headers of the inner packets and parse the destination IP address. Device 3 searches the routing table for the next-hop IP address base on the destination and the ARP entries based on the next-hop IP address. Device 3 uses the ARP entries to identify the destination MAC address, VXLAN tunnel's outbound interface, and VNI.

- If the VXLAN tunnel's outbound interface and VNI cannot be found, Device 3 performs Layer 3 forwarding.
- If the VXLAN tunnel's outbound interface and VNI can be found, Device 3 follows 3.

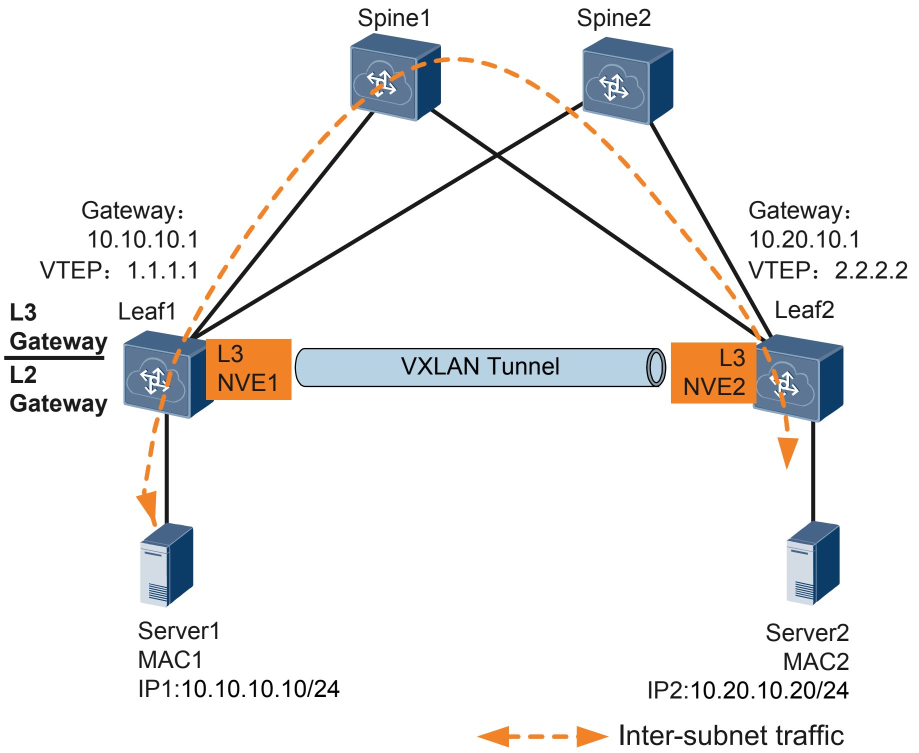
c. Device 3 encapsulate VXLAN packets again, with the source MAC address in the Ethernet header of the inner packets as the MAC address of the Layer 3 gateway interface VBDIF20.

NOTE

For details on communication between Device 3 and other devices, see Layer 2 gateway principles.

- **Distributed VXLAN gateway**

Figure 3-18 Inter-segment packet forwarding in a distributed VXLAN gateway scenario



Inter-subnet IP packet sent by Server1

DMAC	GW MAC
SMAC	MAC1
SIP	IP 1
DIP	IP 2
Pay-load	

VXLAN packet encapsulated by Leaf1

DMAC	Net MAC
SMAC	NVE1 MAC1
SIP	1.1.1.1
DIP	2.2.2.2
UDP S_P	HASH
UDP D_P	4789
VNI	VPN VNI ID
DMAC	Leaf2 MAC
SMAC	Leaf1 MAC
SIP	IP 1
DIP	IP 2
Pay-load	

VXLAN packet decapsulated by Leaf2

DMAC	MAC2
SMAC	Leaf2 MAC
SIP	IP 1
DIP	IP 2
Pay-load	

In a distributed VXLAN gateway scenario shown in **Figure 3-18**, the inter-segment packet forwarding process is as follows:

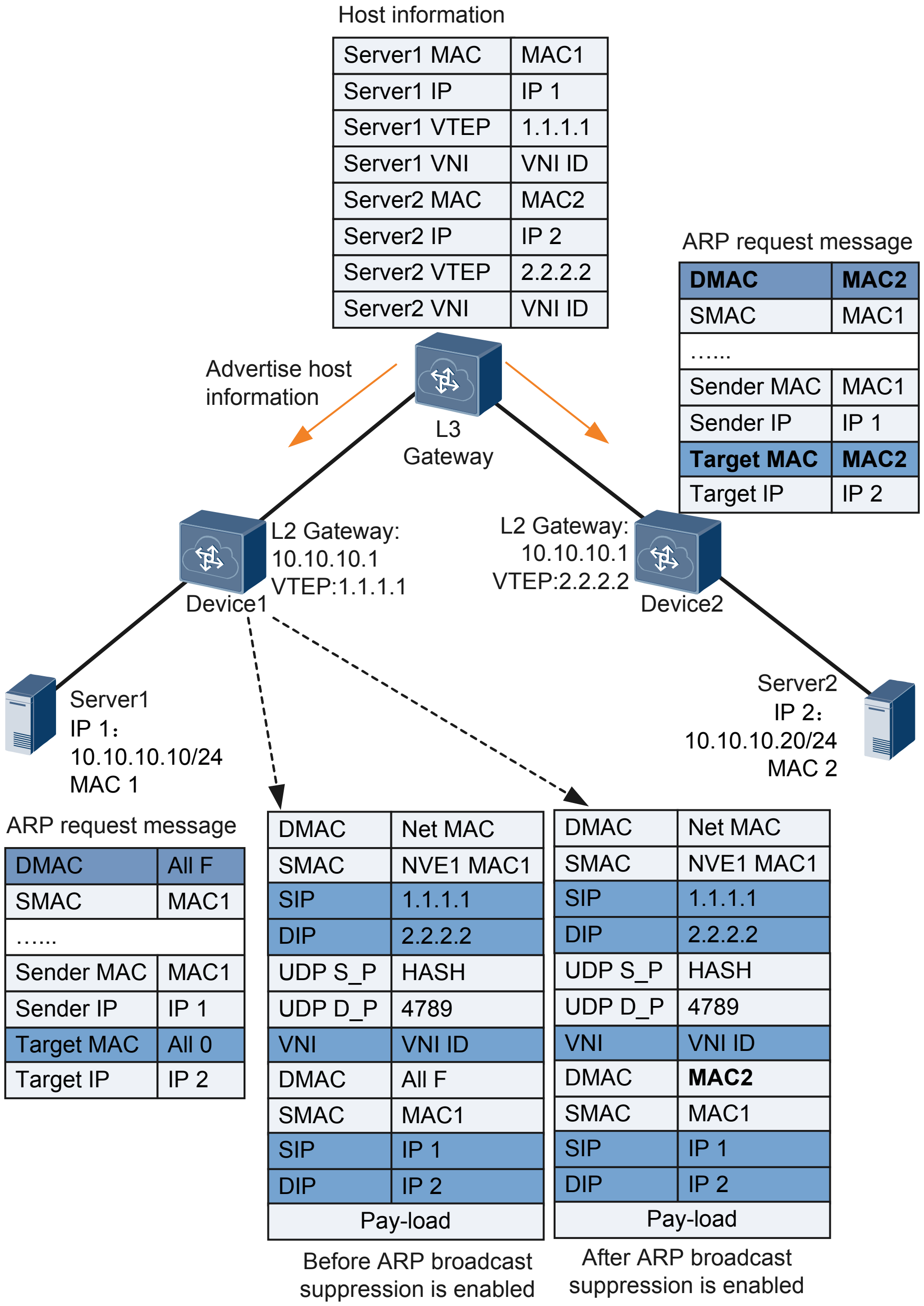
a. After Leaf1 receives IP packets from Server 1 to Server 2 on a different subnet, Leaf1 performs VXLAN encapsulation. Specifically, Leaf1 maps Server 1's subnet VNI to its VPN instance's VNI and finds the gateway's outbound interface of Server 1's IP packets. Leaf1 then forwards the packets at Layer 3 based on the VPN instance bound to the gateway's outbound interface, with the host route's next hop address being Leaf2's VTEP address.

b. After Leaf2 receives the VXLAN packets, it decapsulates the VXLAN packets. Specifically, Leaf2 maps the VNI to a VPN instance, searches a host route in the VPN, and replaces the Ethernet header. Leaf2 then sends the packets to the destination host Server 2.

3.8 ARP Broadcast Suppression

When tenants access each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks, which may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on Layer 2 VXLAN gateways.

Figure 3-19 ARP broadcast suppression networking


Advertise host information
L3 Gateway
L2 Gateway: 10.10.10.1 VTEP:1.1.1.1
Device1
L2 Gateway: 10.10.10.1 VTEP:2.2.2.2
Device2
Server1
IP 1: 10.10.10.10/24
MAC 1
Server2
IP 2: 10.10.10.20/24
MAC 2

Host information

Server1 MAC	MAC1
Server1 IP	IP 1
Server1 VTEP	1.1.1.1
Server1 VNI	VNI ID
Server2 MAC	MAC2
Server2 IP	IP 2
Server2 VTEP	2.2.2.2
Server2 VNI	VNI ID

ARP request message

DMAC	**MAC2**
SMAC	MAC1
……	
Sender MAC	MAC1
Sender IP	IP 1
Target MAC	**MAC2**
Target IP	IP 2

ARP request message

DMAC	All F
SMAC	MAC1
……	
Sender MAC	MAC1
Sender IP	IP 1
Target MAC	All 0
Target IP	IP 2

DMAC	Net MAC
SMAC	NVE1 MAC1
SIP	1.1.1.1
DIP	2.2.2.2
UDP S_P	HASH
UDP D_P	4789
VNI	VNI ID
DMAC	All F
SMAC	MAC1
SIP	IP 1
DIP	IP 2
Pay-load	

Before ARP broadcast suppression is enabled

DMAC	Net MAC
SMAC	NVE1 MAC1
SIP	1.1.1.1
DIP	2.2.2.2
UDP S_P	HASH
UDP D_P	4789
VNI	VNI ID
DMAC	**MAC2**
SMAC	MAC1
SIP	IP 1
DIP	IP 2
Pay-load	

After ARP broadcast suppression is enabled

On the network shown in **Figure 3-19**, the Layer 3 VXLAN gateway dynamically learns ARP entries of tenants and generates host information (host IP address, MAC address, VTEP address, and VNI ID) based on the ARP entries. The Layer 3 VXLAN gateway then uses MP-BGP or BGP EVPN to advertise the host information to MP-BGP or BGP EVPN peers. Layer 2 VXLAN gateways that are MP-BGP or BGP EVPN peers then use the learned host information for ARP broadcast suppression.

When Server1 accesses Server2 for the first time, Server1 broadcasts an ARP request message for Server2's MAC address. The process is as follows:

1. Server1 broadcasts an ARP request message for Server2's MAC address.
2. Device1, a Layer 2 VXLAN gateway, receives the ARP request message and searches host information.
   - If the host information on Device1 contains Server2 information, Device1 replaces the destination MAC address and target MAC in the broadcast ARP request message with Server2's MAC address, performs VXLAN encapsulation, and forwards the VXLAN packet.
   - If the host information on Device1 does not contain Server2 information, Device1 retains the destination MAC address in the broadcast ARP request message, performs VXLAN encapsulation, and forwards the VXLAN packet.
3. After Device2, a Layer 2 VXLAN gateway, receives the VXLAN packet, it performs VXLAN decapsulation and obtains the ARP request message. Device2 then determines whether the destination MAC address in the ARP request message is a broadcast MAC address.
   - If the destination MAC address is a broadcast address, Device2 broadcasts the ARP request message to the user side in the Layer 2 broadcast domain.
   - If the destination MAC address is not a broadcast address, Device2 forwards the ARP request message to the destination host Server2.
4. Server2 receives the ARP request message and responds with an ARP reply message.
5. Server1 receives the ARP reply message and generates an ARP entry in the ARP cache. By now, Server1 can communicate with Server2.

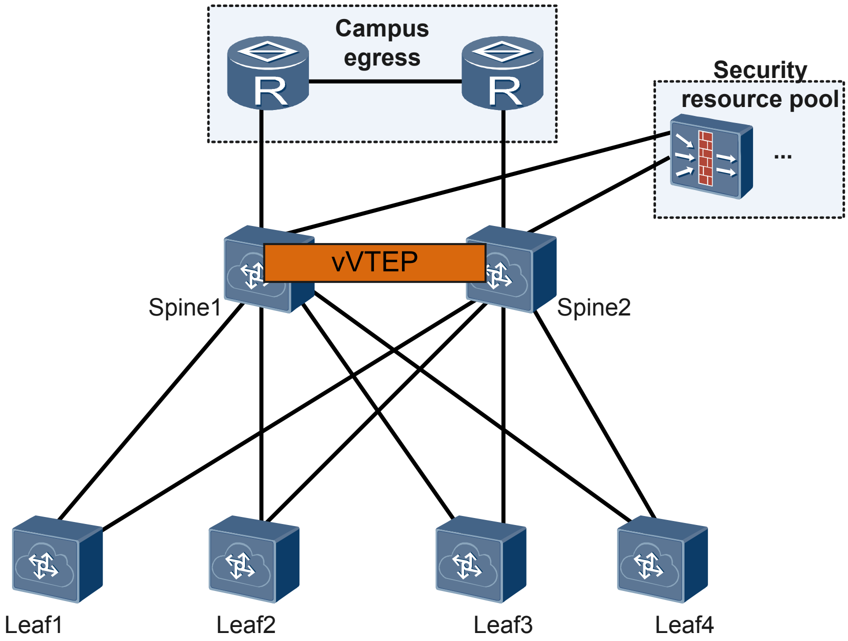
3.9 All-Active VXLAN Gateway

Background

Multiple gateways are often deployed on a VXLAN network to improve reliability. When one gateway fails, traffic can be rapidly switched to another gateway. This prevents service interruptions.

VRRP can be used to improve the reliability. In VRRP networking, only the active gateway can forward traffic and provide the gateway service. The standby gateway provides the gateway service only after the active gateway fails. The switchover mechanism reduces gateway usage and slows down convergence. It is required that reliability be guaranteed and multiple gateways be used to forward traffic to make full use of gateway resources.

Figure 3-20 Centralized all-active VXLAN gateway



Centralized all-active VXLAN gateway can be deployed to meet the preceding requirements. In typical networking composed of spine and leaf switches, the same VTEP address is configured for spine switches to simulate them into one VTEP, and then all spine switches are configured with Layer 3 gateway. Regardless of the spine switch to which traffic is sent, the spine switch can provide the gateway service and correctly forward packets to the next-hop device. Similarly, regardless of the spine switch to which external traffic is sent, traffic can be correctly forwarded to hosts. As shown in **Figure 3-20**, Spine1 and Spine2 are configured with centralized all-active VXLAN gateway so that Spine1 and Spine2 can forward traffic simultaneously. This function improves the device resource usage and convergence performance.

When centralized all-active VXLAN gateway is deployed, the spine switch functions as the Layer 3 gateway. Entries of tenants whose packets are forwarded at Layer 3 are generated on the spine switch, whereas the space of the spine switch is limited. When more and more VMs or servers are deployed, the spine switch may become the bottleneck.

Concepts

In **Figure 3-20**, the concepts relevant to centralized all-active VXLAN gateway are described as follows:

- **Spine**

  Layer 3 gateway on a VXLAN network. The spine switch decapsulates VXLAN packets and forwards them again, and allows servers or VMs between subnets to communicate and physical servers and VMs to communicate with the external network.

- **Leaf**

  Layer 2 access device on a VXLAN network. The leaf switch connects to a physical server or VM, encapsulates packets from physical servers and VMs into VXLAN packets, and transmits them on the VXLAN network.

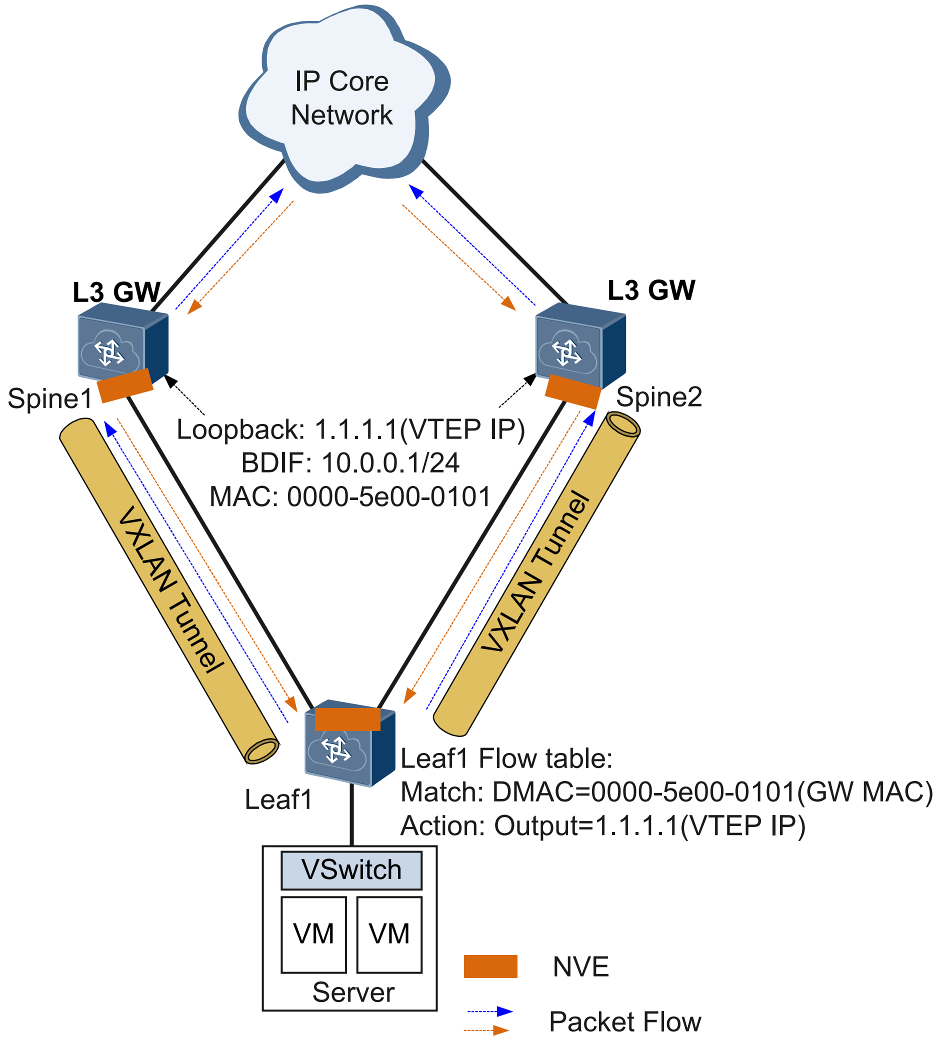
- **vVTEP**

  In all-active VXLAN gateway networking, when the same VTEP address is configured for multiple gateways, the gateways form an all-active gateway group and function as a vVTEP.

All-active gateway packet forwarding

- When the network communication between devices is normal:
  - Leaf1 learns two gateway routes through a routing protocol, and selects the optimum path according to route selection rules. When the costs of the two paths are the same, equal-cost routes are available to implement link backup and traffic load balancing.
  - Spine1 and Spine2 advertise VXLAN Layer 3 gateway routes to the IP core network, and the IP core network advertises routes from other networks to Spine1 and Spine2.

Figure 3-21 Links are normal



- If the link between Spine1 and Leaf1 fails or Spine1 fails:

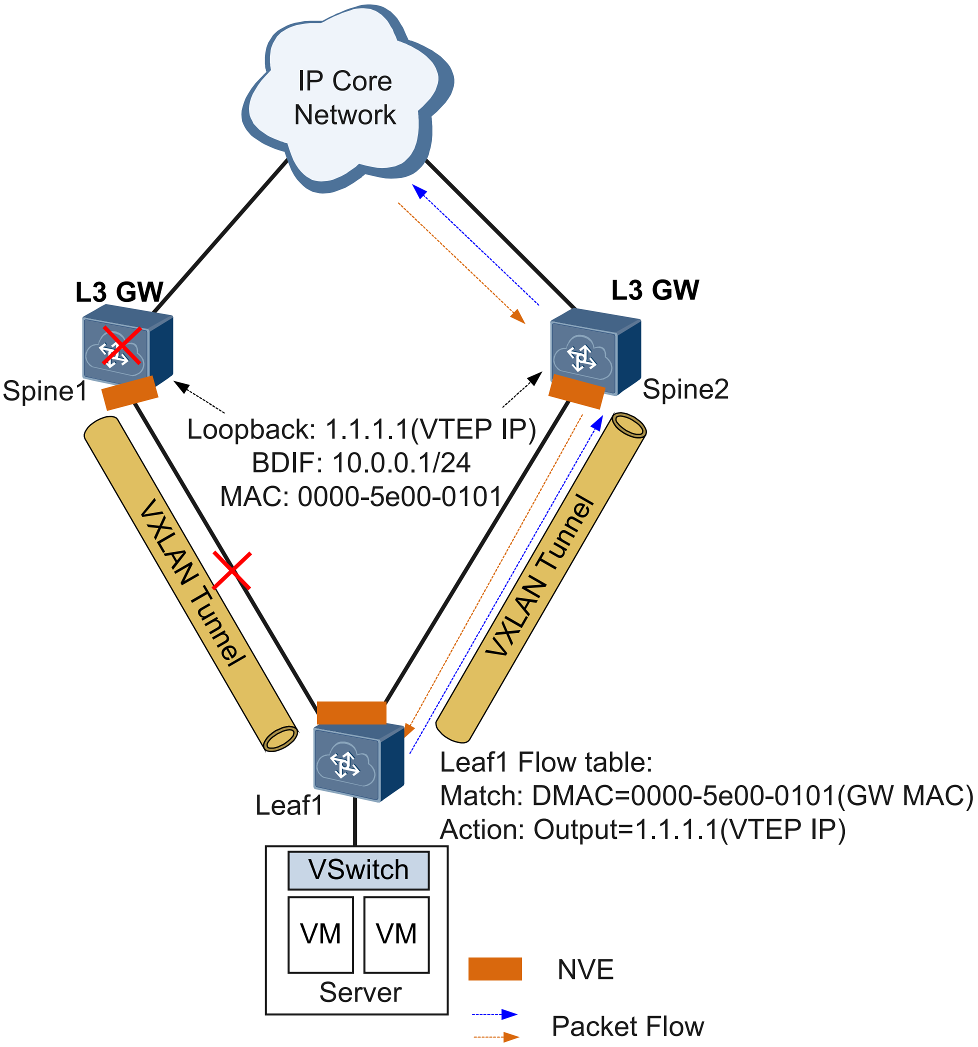
- The routes that come from Spine1 and are saved on Leaf1 are deleted. According to route selection rules, routes of Spine2 are preferred to reach the IP core network.

  NOTE

  If there are multiple links between Leaf1 and Spine1, routes of Spine1 will not be deleted if one link fails. Therefore, Leaf1 can still learn two gateway routes through a routing protocol, and select an optimal path according to route selection rules.

- Spine1's network segment routes are deleted due to the link or device fault. According to route selection rules, routes of Spine2 are preferred to forward traffic from the IP core network to the server. Packets do not reach Spine1.

Figure 3-22 A downlink of a spine switch or the spine switch fails

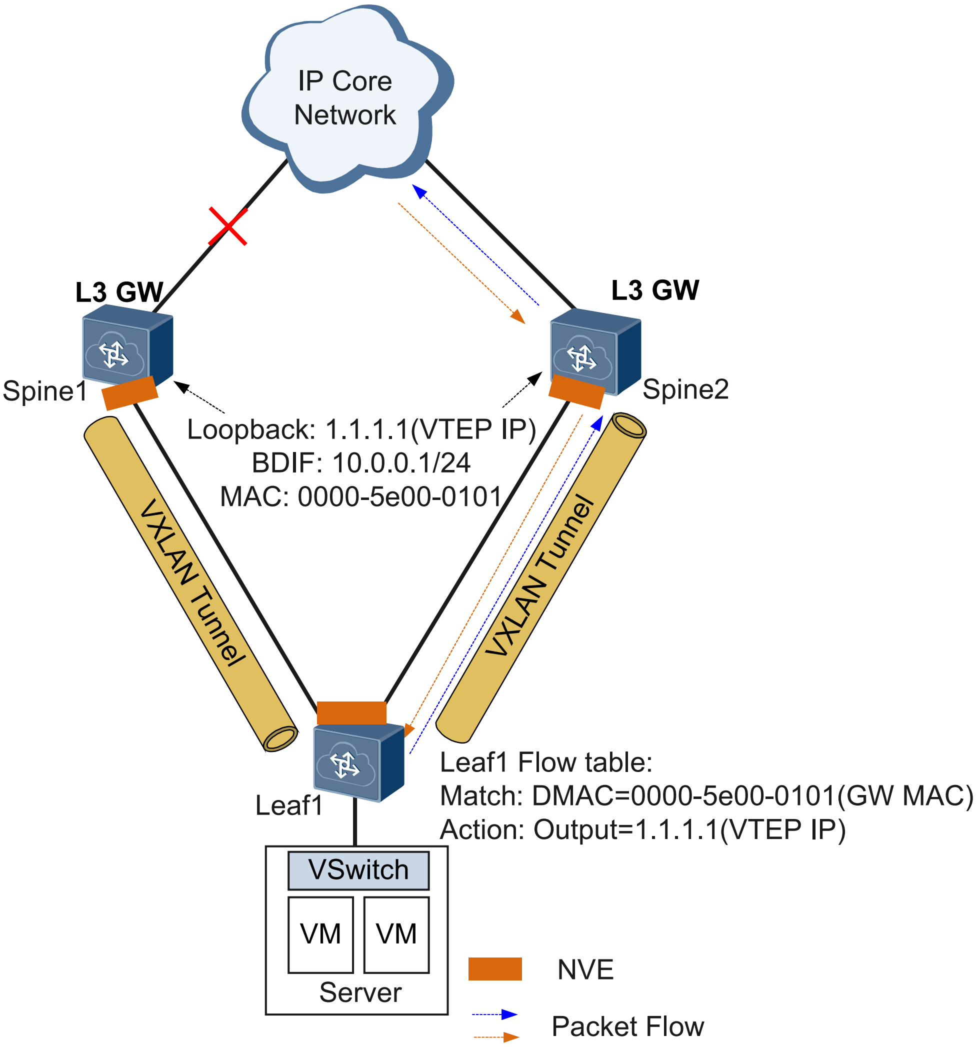


- When the link between Spine1 and the IP core network fails:
  - You can also configure the Monitor Link group so that downlink interface synchronizes the uplink interface status. On Spine1, the routes to the IP core network are deleted. On Leaf1, the saved routes from Spine1 are also deleted.

    Later, traffic from Leaf1 reaches the IP core network through Spine2. Similarly, Spine2 sends traffic from the IP core network to Leaf1.

– The open programmability system (OPS) mechanism can also be used to run Python scripts on devices. When Spine1 detects an uplink fault, Spine1 automatically decreases the priority of the advertised VTEP host route, so that all traffic is bypassed to Spine2. After the uplink fault recovers, Spine1 automatically restores the priority of the advertised VTEP host route, so that traffic is load balanced between Spine1 and Spine2. This method can implement smooth fault recovery.

Figure 3-23 An uplink of a spine switch fails



ARP Entry Synchronization

In an all-active gateway scenario, multiple gateways advertise routes of the same subnet to the upper-layer routing device so that the upper-layer routing device has equal-cost routes to the specified network segment. Traffic from the upper-layer routing device is sent to a gateway through an equal-cost route. If there is no ARP entry of the destination host on the gateway, ARP packets are flooded and traffic is discarded.

To ensure correct traffic forwarding, all-active gateways must synchronize ARP entries. That is, when any host in the subnet where a gateway is deployed goes online, all gateways learn the ARP entry of the host. The device provides the following modes to implement ARP entry synchronization:

- Controller mode: On all-active gateways, dynamic ARP learning is disabled on their VBDIF interfaces. The controller manages and controls ARP entries uniformly. When a host goes online, the controller obtains the ARP entry of the host and delivers the ARP entry to all-active gateways.
- Single-node mode: The device automatically learns ARP entries, but does not depend on the controller. The working mechanism is as follows:
  a. A user specifies the IP addresses of all neighbors of a gateway in a DFS group, so that the gateway establishes neighbor relationships with devices with specified IP addresses.
  b. After neighbor relationships are established, the gateways synchronize ARP entries from each other.

     Two methods are available for synchronizing ARP entries:
     - Real-time synchronization: After receiving an ARP request packet, a gateway synchronizes the new entry or the change to an existing entry to other gateways to ensure ARP entry consistency on the gateways.
     - Batch synchronization: A gateway working for a period synchronizes its large number of ARP entries to a new gateway or an existing gateway that recovers from a fault in a batch.

3.10 VXLAN Dual-Active Access

Background

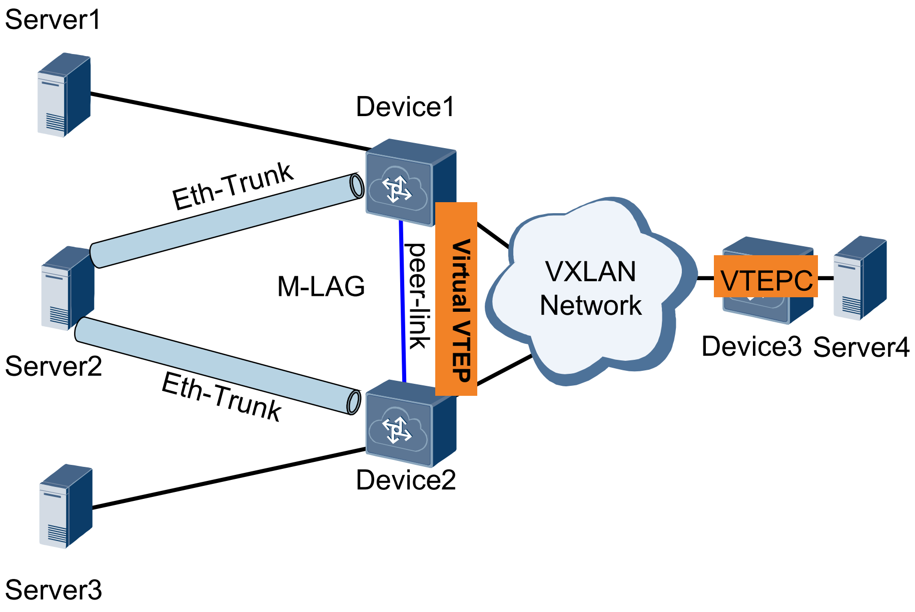
To improve reliability, servers are often dual-homed to a VXLAN network through double network adapters. When one network adapter of a server fails, services are not interrupted.

In the preceding scenario, only the active network adapter can receive and send packets, whereas the standby network adapter cannot. This results in a waste of the network adapter and link bandwidth. Two network adapters are required to work in dual-active mode to forward traffic simultaneously, making full use of network adapters and bandwidth resources.

The following problems may occur:

- The server may receive the same traffic from two upstream devices connected to it.
- The network device that communicates with the server continuously receives traffic from two devices, causing frequent MAC address flapping on the network device.

Figure 3-24 VXLAN dual-active access networking



VXLAN dual-active can solve the preceding problems. As shown in **Figure 3-24**, Server2 is dual-homed to a VXLAN network.

- Multi-chassis link aggregation group (M-LAG) technologies virtualize two access devices connected to the server into one device, eliminating the redundant path.
- Two dual-homing devices use the same Virtual VTEP address. For the remote device, the two devices function as one logical device that connects to the VXLAN network. MAC address flapping is therefore eliminated.

Concepts

In **Figure 3-24**, the concepts relevant to VXLAN dual-active access are described as follows:

- **vVTEP**

  In VXLAN dual-active access networking, when the same VTEP IP address is for the access devices connected to a dual-homed server, the devices encapsulate the same VTEP IP address in VXLAN packets. For other devices on the same VXLAN network, the two devices function as one logical device.

- **Peer-link**

  There must be a direct link between two devices where M-LAG is deployed and the link must be a peer-link. A peer-link is a protection link.

  After an interface is configured as a peer-link interface, the device automatically creates a QinQ sub-interface for each VNI on the interface. The QinQ sub-interface is used to add the two M-LAG-enabled devices to the corresponding BD of the VNI. Users cannot perform operations on the QinQ sub-interface.

  When traffic enters the peer-link interface, the switch can map IP packets based on the DSCP priority only and map non-IP packets based on the priority by the **port priority** command only.

- **Dynamic Fabric Service (DFS) group**

A DFS group is used for device pairing to ensure correct service packet forwarding in VXLAN dual-active access networking.

- M-LAG interface

  An M-LAG interface is an Eth-Trunk that is established between two M-LAG-enabled devices and connect to a server.

Working Mechanism of Access-Side M-LAG

The following describes M-LAG protocol packets and their functions.

- M-LAG negotiation packet

  As shown in **Figure 3-24**, after the M-LAG configuration is complete, the devices exchange M-LAG negotiation packets over the peer-link to pair with each other. After they pair into a DFS group, they negotiate to determine the master and backup states.

- M-LAG heartbeat packet

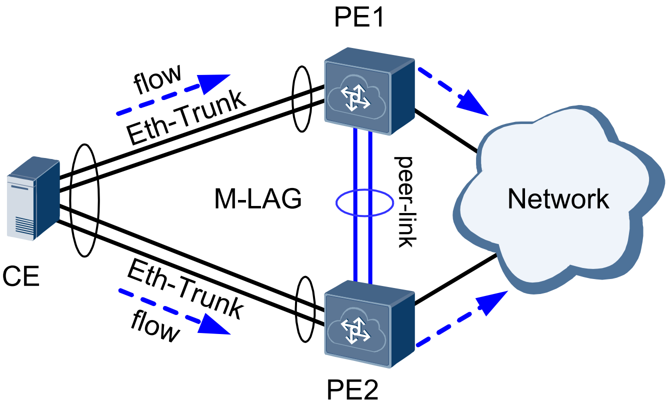
  As shown in **Figure 3-24**, after completing master/backup negotiation, the devices send M-LAG heartbeat packets over network-side links to detect the status of the remote device.

The following figure describes how devices configured with M-LAG determine the master/ backup and link status when the network is normal and faulty.

- In VXLAN dual-active access networking that is running normally:

  The Eth-Trunk links are both in Up state. PE1 and PE2 load balance traffic. Services are isolated unidirectionally on peer-links and at the M-LAG and network sides to prevent loops on the network.

Figure 3-25 No fault occurs



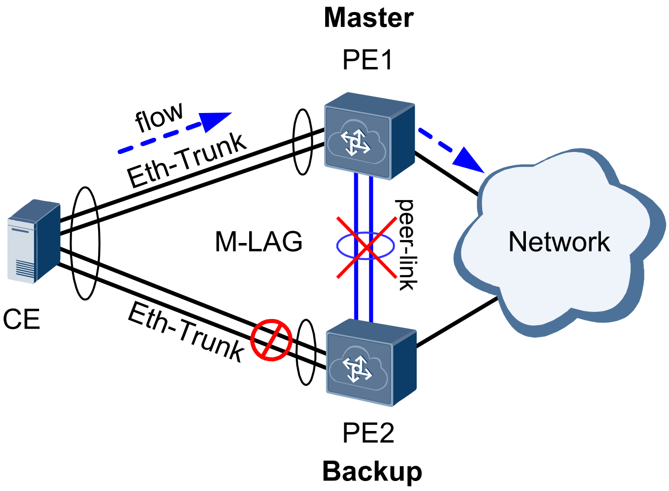
- When the peer-link fails:

  The master and slave states of the devices determine the Eth-Trunk status. The Eth-Trunk on the master device is still Up. The Eth-Trunk on the backup device becomes Down, and the dual-homing networking changes into a single-homing networking. If the peer-link fails but the heartbeat status is normal, the M-LAG interface on the slave device enters the Error-Down state. When the peer-link recovers, the physical interface in Error-Down state are restored to Up state.

NOTE

If a peer-link interface and M-LAG interface are deployed on the same card, a failure of the peer-link interface causes the M-LAG interface to fail. In this situation, the Eth-Trunks at both ends become Down, causing a traffic forwarding failure and service interruption. To improve reliability, you are advised to deploy the peer-link interface and M-LAG interface on different cards.

Figure 3-26 Peer-link fails



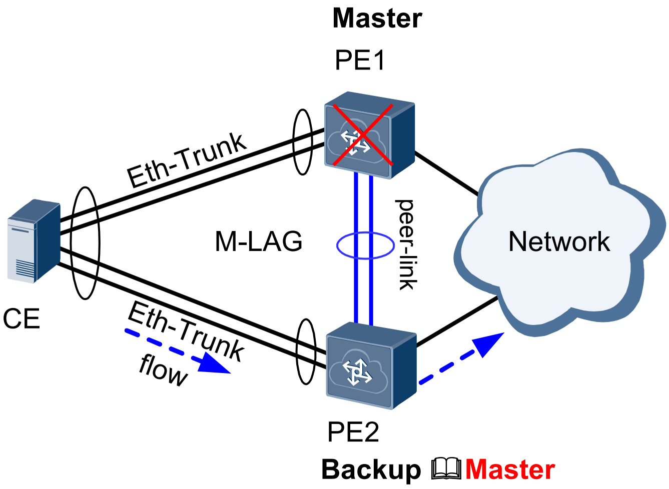
- When the master device fails:

  The backup device becomes the master device and continues forwarding traffic, and its Eth-Trunk is still in Up state. The Eth-Trunk on the master device becomes Down, and the dual-homing networking changes into a single-homing networking.

  NOTE

  If the backup device fails, the master and backup states remain unchanged and the Eth-Trunk of the backup device becomes Down. The Eth-Trunk on the master device is still in Up state and continues forwarding traffic, and the dual-homing networking changes into a single-homing networking.

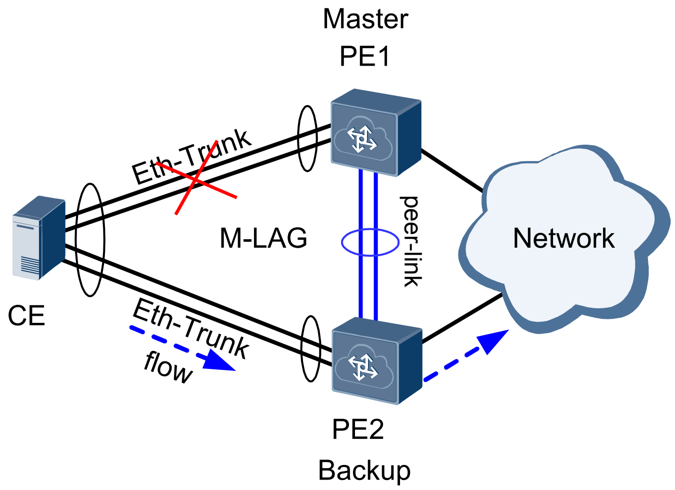
Figure 3-27 Master device fails



- When the Eth-Trunk on the VXLAN network fails, the following situations occur:

  The M-LAG master and backup states remain unchanged, and traffic is switched to another Eth-Trunk. The faulty Eth-Trunk becomes Down. M-LAG stops traffic forwarding on the faulty Eth-Trunk, and the dual-homing networking changes into a single-homing networking.

Figure 3-28 Eth-Trunk fails



Packet Forwarding Process in VXLAN Dual-Active Access Networking

In VXLAN dual-active access networking, the same vVTEP address is manually configured on Device1 and Device2 so that Device1 and Device2 encapsulate the same vVTEP address in VXLAN packets.

In addition, Device1 and Device2 check the vVTEP address of each other. If Device1 and Device2 have the same vVTEP address, they exchange their VTEP addresses and MAC addresses. When the peer-link or one device fails, VXLAN notifies the other device quickly to change the dual-homing networking into a single-homing networking.

As shown in **Figure 3-24**, a peer-link is deployed between Device1 and Device2, and the two devices use the same vVTEP address. Server2, Device1, and Device2 constitute VXLAN dual-active access networking. VXLAN protocol processes traffic of different types and from different directions differently.

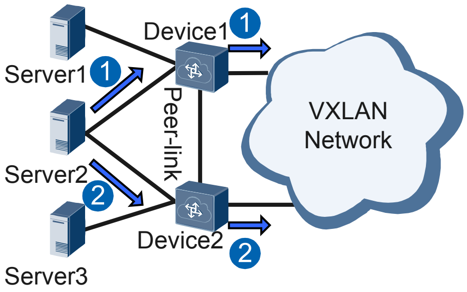
- Unicast traffic from a dual-active interface

  Device1 and Device2 work in load balancing mode to forward traffic together.

  NOTE

  Numbers 1 and 2 in the figure represent different types of traffic.

Figure 3-29 Unicast traffic from a dual-active interface

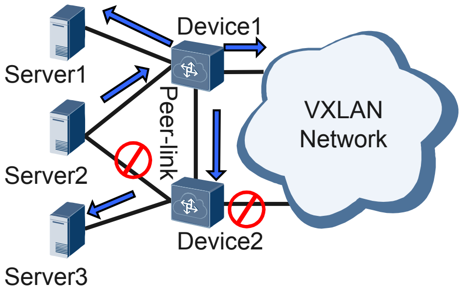


- BUM traffic from a dual-active interface

  BUM traffic from Server2 is load balanced between Device1 and Device2. The following uses the forwarding process on Device1 as an example.

  Device1 encapsulates the vVTEP address into the received BUM traffic and forwards the traffic to each next-hop device. When the traffic arrives at Device2, Device2 forwards the traffic only to Server3 but not to Server2 or the VXLAN network side according to the unidirectional isolation mechanism.

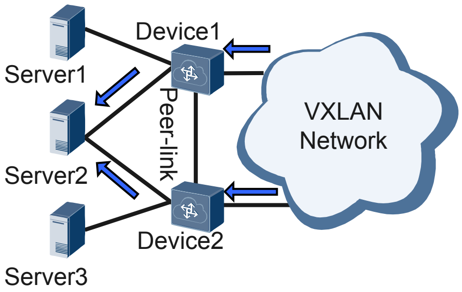
Figure 3-30 BUM traffic from a dual-active interface



- Unicast traffic from the VXLAN network

  If the unicast traffic is sent to a dual-active interface, traffic is encapsulated with the vVTEP address and is load balanced between Device1 and Device2, which then forward the traffic to the attached dual-homed device.

Figure 3-31 Unicast traffic from the VXLAN network

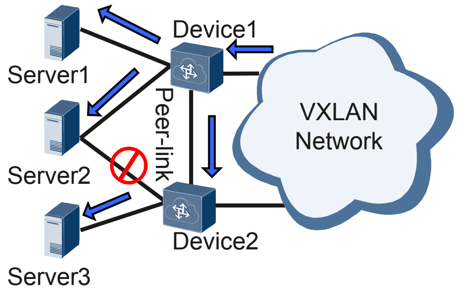


- BUM traffic from the VXLAN network

  The BUM traffic that is sent to a dual-active interface and encapsulated with the vVTEP address is load balanced between Device1 and Device2. The following uses Device1 as an example.

  Device1 decapsulates the traffic and then forwards the traffic to each user-side interface. Because the peer-link is isolated from the backup interface, traffic arriving at Device2 is not forwarded to Server2, avoiding routing loops.

Figure 3-32 BUM traffic from the VXLAN network



3.11 Application for Inter-Domain Active-Active VXLAN Gateways

Background

With the fast development of mobile Internet, data centers provide increasingly diversified services and carry explosively growing traffic. Users become more sensitive to service interruptions caused by failures or inefficient maintenance on computing, storage, or network devices.

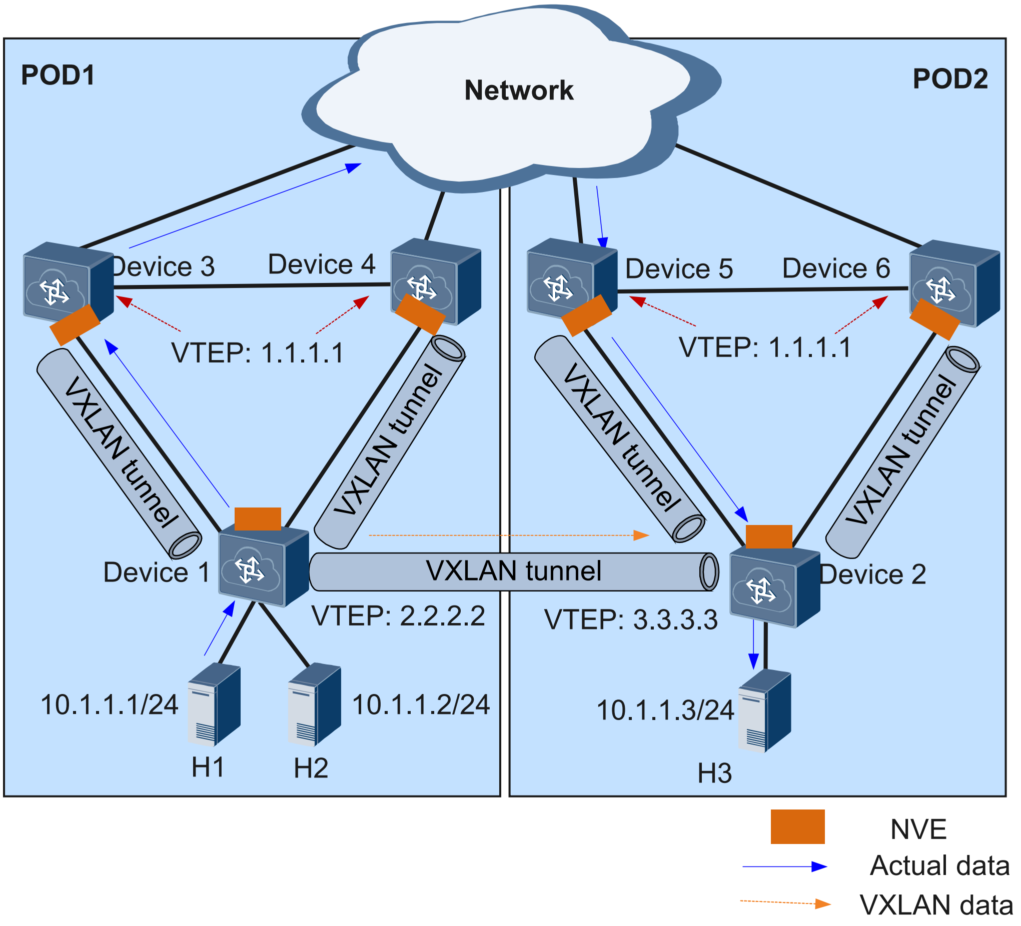
In legacy networking, inter-domain applications are widely used. To ensure service continuity, implementing inter-domain VXLAN interconnection is of great importance.

Inter-domain communication can be implemented across DCs or across points of delivery (PODs) in the same data center.

Inter-domain active-active VXLAN gateways allow inter-domain communication between hosts on different network segments and inter-domain VM migration. In each domain, two active VXLAN gateways are deployed to balance or back up traffic for higher network reliability.

On the network shown in **Figure 3-33**, Device 1 is dual-homed to Layer 3 gateways Device 3 and Device 4 through VXLAN tunnels in POD1; Device 2 is dual-homed to Layer 3 gateways Device 5 and Device 6 through VXLAN tunnels in POD2; Device 1 and Device 2 also establish a VXLAN tunnel for intra-segment communication. The four Layer 3 gateways (Device 3, Device 4, Device 5, and Device 6) have the same MAC address and Layer 2 VTEP's IP address, allowing them to function as the same VTEP. This VTEP provides gateway services for traffic sent to any gateway among them.

Figure 3-33 Inter-domain active-active VXLAN gateways



Inter-domain active-active VXLAN gateways can be deployed in the following scenarios:

- Inter-domain communication between hosts on the same network segment (using known unicast packets or BUM packets)
- Inter-domain communication between hosts on different network segments
- Inter-domain VM migration

Inter-Domain Communication Between Hosts on the Same Network Segment

The forwarding process for known unicast packets on the same network segment in different domains is the same as that for known unicast packets on the same network segment in the same domain.

As shown in **Figure 3-33**, H1 and H3 reside on the same network segment but in different PODs. When H1 sends a known unicast data packet to H3, the packet forwarding process is as follows:

1. After Device 1 receives H1's packet, it determines the Layer 2 BD of the packet based on the access interface and VLAN information and searches for the outbound interface and encapsulation information in the BD.
2. Device 1's VTEP performs VXLAN encapsulation based on the outbound interface and encapsulation information and forwards the packet to Device 3.

3. Device 3 sends the VXLAN packet to Device 5, which further forwards the packet to Device 2.

   NOTE

   - Path selection in active-active VXLAN gateway scenarios is determined by the configured routing protocol. In **Figure 3-33**, the packet is forwarded along the path Device 1 -> Device 3 -> Device 5 -> Device 2.
   - The inter-domain Layer 3 gateways synchronize their routing entries using BGP EVPN.

4. Upon receipt of the VXLAN packet, Device 2's VTEP verifies the VXLAN packet based on the UDP destination port number, source and destination IP addresses, and VNI. Device 2 obtains the Layer 2 BD based on the VNI and performs VXLAN decapsulation to obtain the inner Layer 2 packet.
5. Device 2 searches for the outbound interface and encapsulation information in the Layer 2 BD and forwards the packets to H3.

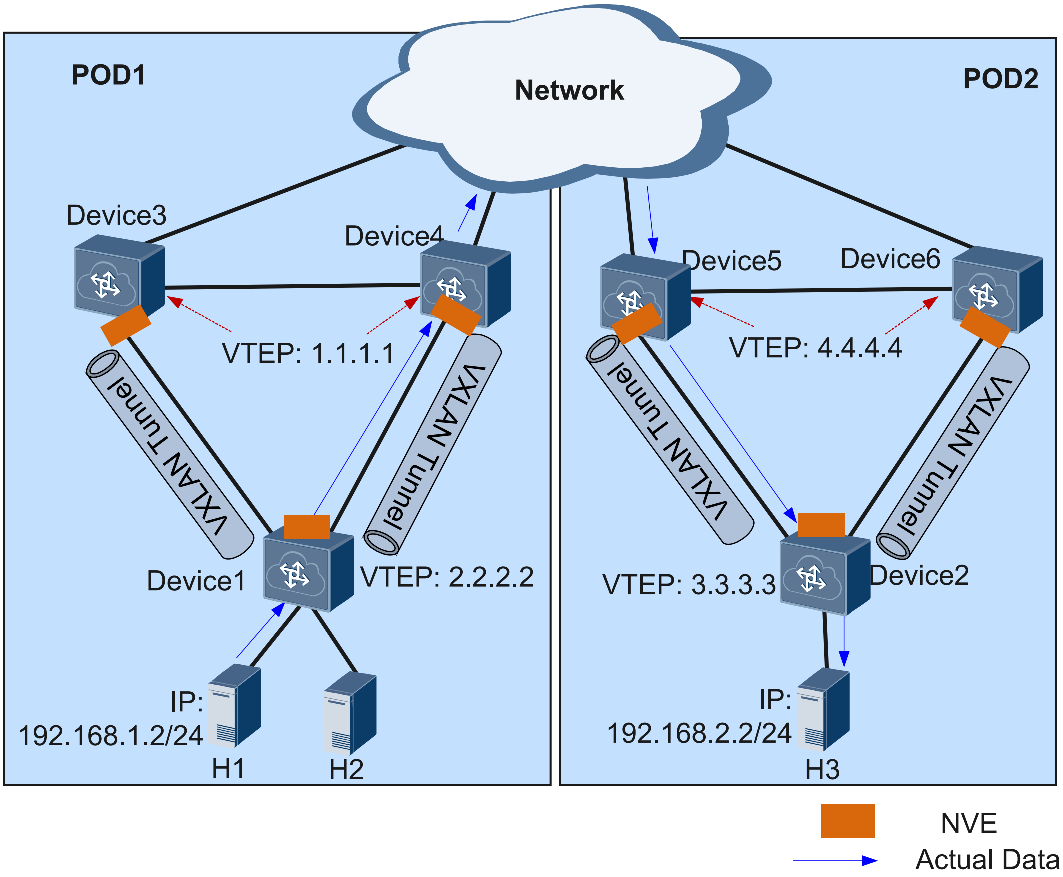
The forwarding process for BUM packets on the same network segment in different domains is the same as that for BUM packets on the same network segment in the same domain. For details, see **BUM Packet Forwarding Process**.

Inter-Domain Communication Between Hosts on Different Network Segments

To implement inter-domain communication between hosts on different network segments, Layer 3 gateways are needed to perform Layer 3 forwarding. To prevent traffic bypass, Layer 3 gateways must use BGP to advertise host routes to each other.

As shown in **Figure 3-34**, H1 and H3 reside on different network segments in different PODs. Device 1 is dual-homed to Layer 3 gateways Device 3 and Device 4, and Device 2 is dual-homed to Layer 3 gateways Device 5 and Device 6. These Layer 3 gateways share the same VBDIF interface's IP address, MAC address, and VTEP's IP address.

Figure 3-34 Inter-domain active-active VXLAN gateways for communication between hosts on different network segments



When H1 sends a data packet to H3, the packet forwarding process for Layer 3 gateways is as follows:

1. After Device 4 receives the VXLAN packet, it decapsulates the packet and checks whether the destination MAC address in the inner packet is the MAC address of the local VBDIF interface.
   - If so, Device 4 forwards the inner packet to Device 5 on the destination network segment.
   - If not, Device 4 searches the outbound interface and encapsulation information in the Layer 2 BD.
2. Upon receipt of the inner packet, Device 5 removes the inner Ethernet header, parses the destination IP address, and searches the routing table for a next hop address. Then, Device 5 searches the ARP table based on the next hop address to determine the destination MAC address, VXLAN tunnel's outbound interface, and VNI.
   - If the VXLAN tunnel's outbound interface and VNI cannot be found, Device 5 performs Layer 3 forwarding.
   - If the VXLAN tunnel's outbound interface and VNI can be found, Device 5 follows 3.
3. Device 5 encapsulates the inner packet into a VXLAN packet again, with the source MAC address in the Ethernet header of the inner packet being the MAC address of Device 5's VBDIF interface.

NOTE

Path selection in active-active VXLAN gateway scenarios is determined by the configured routing protocol. In **Figure 3-34**, the packet is forwarded along the path Device 1 -> Device 4 -> Device 5 -> Device 2.

Inter-Domain VM migration

- **Service Description**

  Enterprises on data center networks deploy server virtualization to implement IT resource integration, improve resource usage, and reduce network costs. With wider deployment of server virtualization, more VMs are running on physical servers, and more applications are running in virtualization environments. This brings challenges to virtual networks and requires inter-domain VM migration.

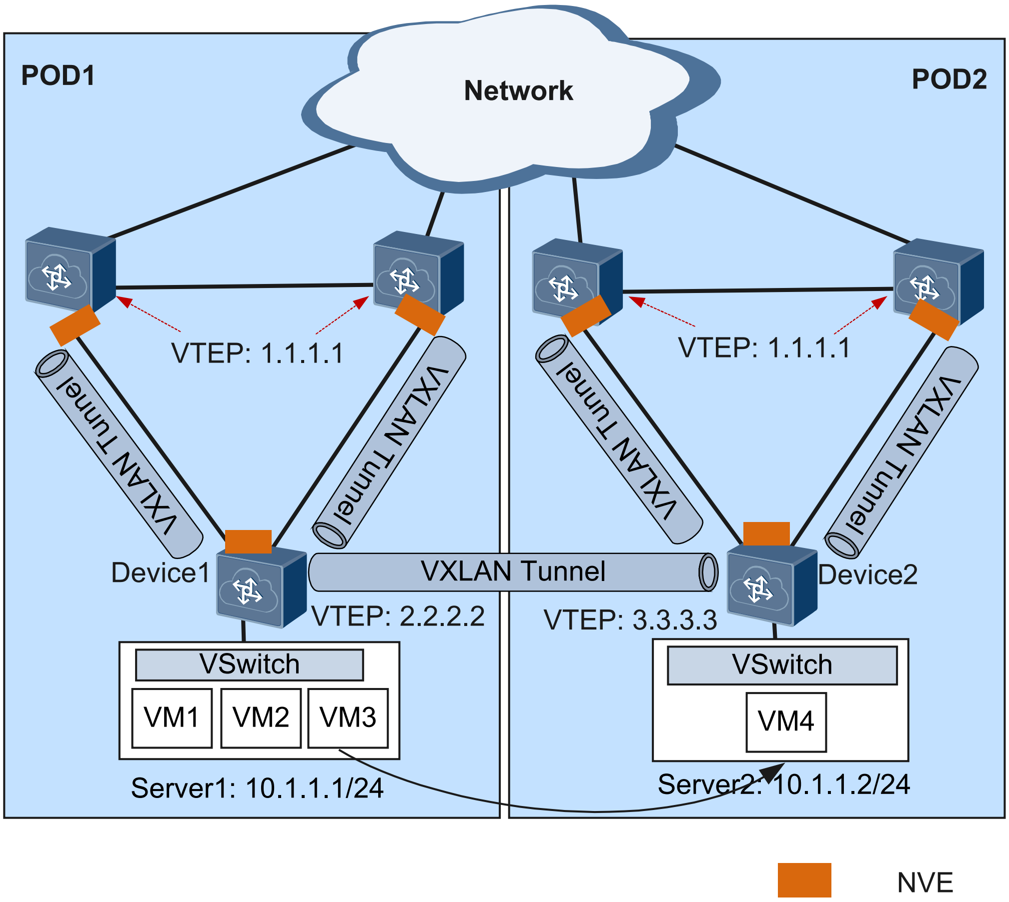
  Deploying inter-domain active-active VXLAN gateways allows inter-domain VM migration without affecting services.

- **Networking Description**

  On the network shown in **Figure 3-35**, Server 1 and Server 2 reside on the same network segment. Server 1 is deployed on POD1's Device 1, and Server 2 is deployed on POD2's Device 2. Device 1 and Device 2 are dual-homed to different Layer 3 gateways.

  Server 1 has three VMs, and Server 2 has only one VM. Server 1's computing space is insufficient, whereas Server 2's is not fully used. Therefore, VM3 needs to be migrated to Server 2 without affecting services.

Figure 3-35 VM migration networking



- **Feature Deployment**

  In inter-domain active-active VXLAN gateway scenarios, the following conditions must be met for VM migration:

  - The Layer 3 gateways in POD1 and POD2 have the same Layer 2 VTEP's IP address and MAC address configured.
  - BGP runs between inter-domain Layer 3 gateways to advertise host routes.

  The process for migrating VM3 in POD1 to POD2 is as follows:

  a. Migrate VM3 from POD1's Device 1 to POD2's Device 2.

  b. VM3 sends gratuitous ARP or RARP packets to the gateways and other devices in POD2.

  c. Upon receipt, the gateways in POD2 update host routes accordingly and use BGP to advertise the host routes.

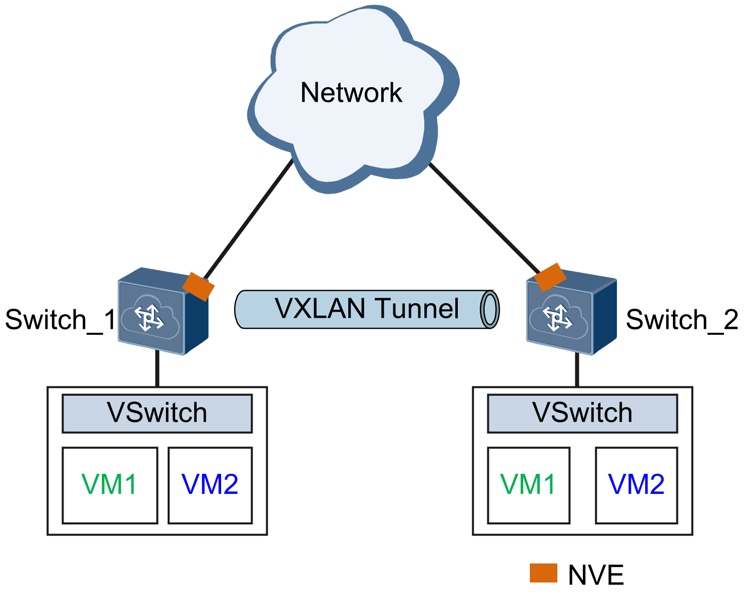
  d. After receiving the host routes advertised from POD2, the gateways in POD1 find that VM3 has been migrated and send ARP aging probe packets.

  e. The gateways in POD1 do not receive any response, and therefore delete the ARP entries of VM3 and withdraw the old host routes.

3.12 VXLAN QoS

VXLAN quality of service (QoS) provides differentiated service in VXLAN applications. A device implements mapping between QoS priorities in original packets, internal priorities (local precedence assigned by the device to differentiate service classes of packets), and priorities of encapsulated packets. In this way, packets are processed differently based on internal priorities.

Figure 3-36 VXLAN networking



On the network as shown in **Figure 3-36**, VXLAN QoS implements mapping between QoS priorities in original packets, internal priorities, and priorities of encapsulated packets according to the following process:

1. An original packet arrives at a Layer 2 sub-interface on Switch_1. Switch_1 maps the 802.1p priority of the original packet to the internal priority, that is per-hop behavior (PHB) and color, based on the DiffServ profile bound to the specified VLAN on the sub-interface, and then sends the packet to the specified queue.
2. Before the packet enters the VXLAN tunnel from Switch_1, Switch_1 encapsulates the packet with a VXLAN header, UDP header, IP header, and Ethernet header in turn, and then maps the packet's internal priority to the 802.1p priority or differentiated services code point (DSCP) priority based on the default profile in the DiffServ domain. The packet is then transmitted over the VXLAN tunnel based on the 802.1p priority or DSCP priority.
3. When the packet leaves the tunnel, its 802.1p priority or DSCP priority (only the DSCP priority is trusted when an Ethernet interface works in Layer 3 mode) is mapped to the internal priority based on the default profile in the DiffServ domain. The packet then enters the queue matching the internal priority.
4. Finally, the internal priority is mapped to the 802.1p priority based on the profile in the DiffServ domain bound to the specified VLAN on the sub-interface. The packet is transmitted through the outbound interface based on the 802.1p priority.

For details, see Applying the DiffServ Domain.

4 Applications

About This Chapter

This section describes VXLAN applications.

4.1 Application for Communication Between Terminal Users on a VXLAN

4.2 Application for Communication Between Terminal Users on a VXLAN and Legacy Network

4.3 Application in VM Migration Scenarios

4.1 Application for Communication Between Terminal Users on a VXLAN

Service Description

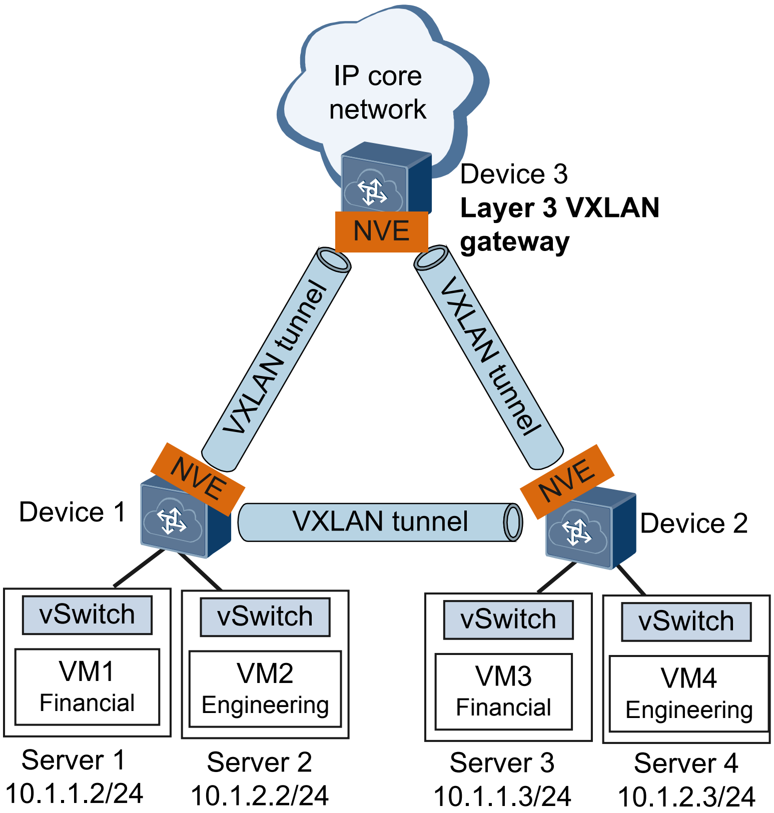
Currently, data centers are expanding on a large scale for enterprises and carriers, with increasing deployment of virtualization and cloud computing. In addition, to accommodate more services while reducing maintenance costs, data centers are employing large Layer 2 and virtualization technologies.

As server virtualization is implemented in the physical network infrastructure for data centers, VXLAN, an NVO3 technology, has adapted to the trend by providing virtualization solutions for data centers.

Networking Description

On the network shown in **Figure 4-1**, an enterprise has VMs deployed in different data centers. Different network segments run different services. The VMs running the same service or different services in different data centers need to communicate with each other. For example, VMs of the financial department residing on the same network segment need to communicate, and VMs of the financial and engineering departments residing on different network segments also need to communicate.

Figure 4-1 Communication between terminal users on a VXLAN



Feature Deployment

As shown in **Figure 4-1**:

- Deploy Device 1 and Device 2 as Layer 2 VXLAN gateways and establish a VXLAN tunnel between Device 1 and Device 2 to allow communication between terminal users on the same network segment.
- Deploy Device 3 as a Layer 3 VXLAN gateway and establish a VXLAN tunnel between Device 1 and Device 3 and between Device 2 and Device 3 to allow communication between terminal users on different network segments.

Configure VXLAN on devices to trigger VXLAN tunnel establishment and dynamic learning of ARP and MAC address entries. By now, terminal users on the same network segment and different network segments can communicate through the Layer 2 and Layer 3 VXLAN gateways based on ARP and routing entries.

4.2 Application for Communication Between Terminal Users on a VXLAN and Legacy Network

Service Description

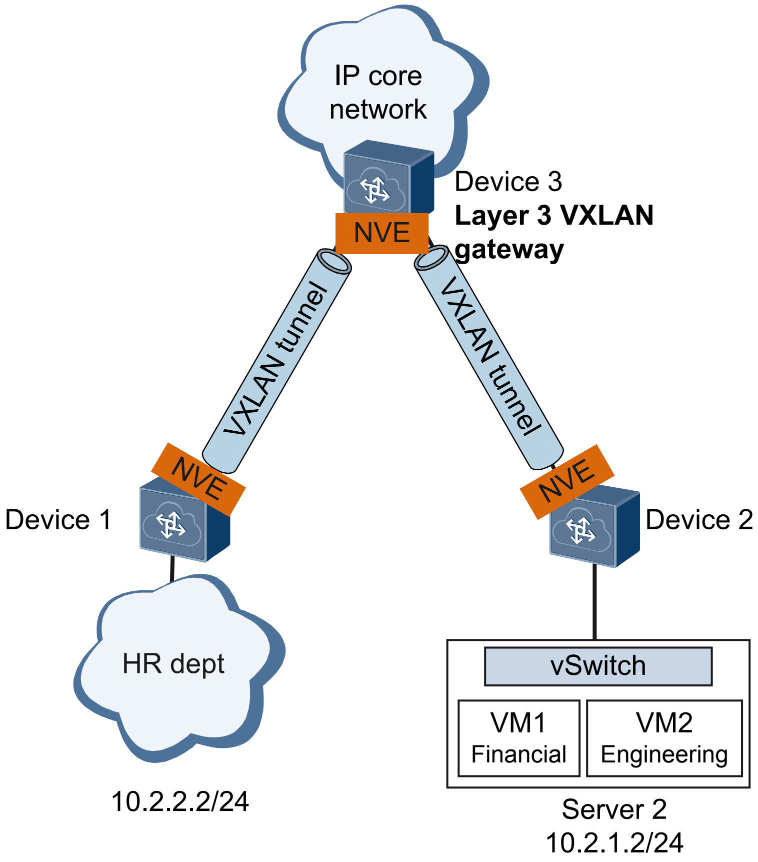
Currently, data centers are expanding on a large scale for enterprises and carriers, with increasing deployment of virtualization and cloud computing. In addition, to accommodate more services while reducing maintenance costs, data centers are employing large Layer 2 and virtualization technologies.

As server virtualization is implemented in the physical network infrastructure for data centers, VXLAN, an NVO3 technology, has adapted to the trend by providing virtualization solutions for data centers, allowing intra-VXLAN communication and communication between VXLANs and legacy networks.

Networking Description

On the network shown in **Figure 4-2**, an enterprise has VMs deployed for the finance and engineering departments and a legacy network for the human resource department. The finance and engineering departments need to communicate with the human resource department.

Figure 4-2 Communication between terminal users on a VXLAN and legacy network



Feature Deployment

As shown in **Figure 4-2**:

Deploy Device 1 and Device 2 as Layer 2 VXLAN gateways and Device 3 as a Layer 3 VXLAN gateway. The VXLAN gateways are VXLANs' edge devices connecting to legacy networks and are responsible for VXLAN encapsulation and decapsulation. Establish a VXLAN tunnel between Device 1 and Device 3 and between Device 2 and Device 3 for VXLAN packet transmission.

When the human resource department sends a packet to VM1 of the financial department, the process is as follows:

1. Device 1 receives the packet and encapsulates it into a VXLAN packet before sending it to Device 3.
2. Upon receipt, Device 3 decapsulates the VXLAN packet and removes the Ethernet header in the inner packet, parses the destination IP address, and searches the routing table for a next hop address. Then, Device 3 searches the ARP table based on the next hop address to determine the destination MAC address, VXLAN tunnel's outbound interface, and VNI.
3. Device 3 encapsulates the VXLAN tunnel's outbound interface and VNI into the packet and sends the VXLAN packet to Device 2.
4. Upon receipt, Device 2 decapsulates the VXLAN packet, finds the outbound interface based on the destination MAC address, and forwards the packet to VM1.

4.3 Application in VM Migration Scenarios

Service Description

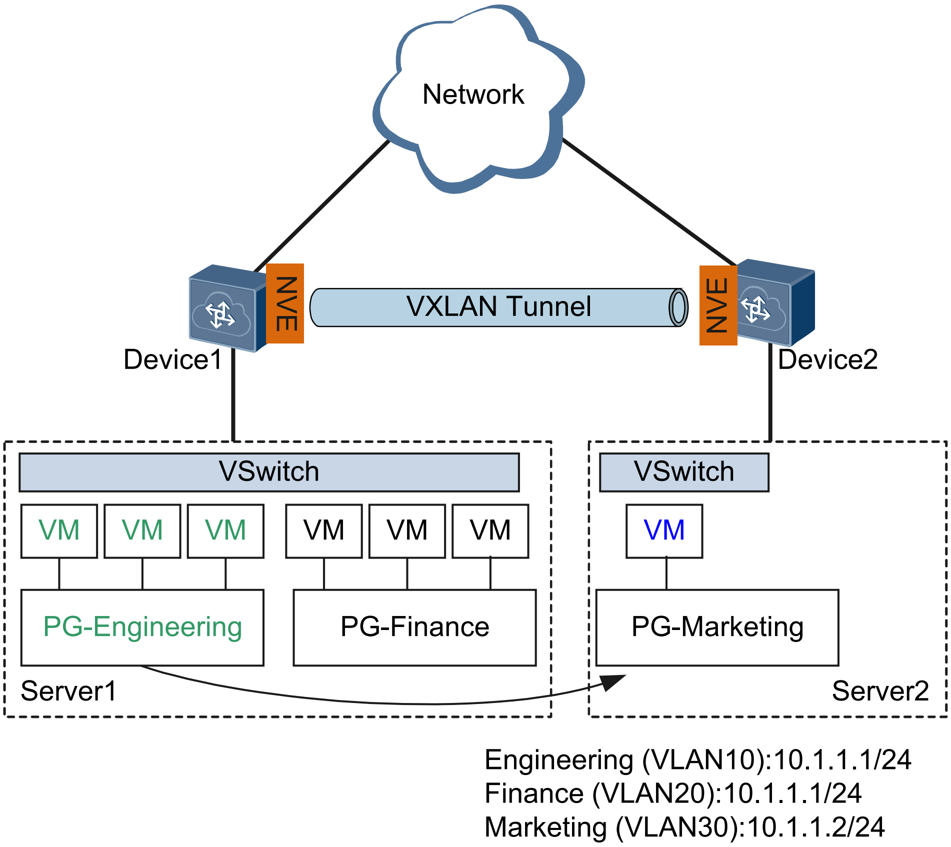
Enterprises on data center networks deploy server virtualization to implement IT resource integration, improve resource usage, and reduce network costs. With the wider deployment of server virtualization, more VMs are running in physical servers, and more applications are running in virtualization environments, which brings challenges to virtual networks.

Networking Description

On the network shown in **Figure 4-3**, an enterprise has two clusters in the data center: engineering and financial departments in Server1 and the marketing department in Server2.

The computation space on Server1 is inadequate, whereas that on Server2 is not fully utilized. The network administrator wants to migrate the engineering department to Server2 without affecting services.

Figure 4-3 Enterprise distribution networking



Feature Deployment

To ensure that services are not interrupted during the migration of the engineering department, the IP and MAC addresses of the engineering department must remain unchanged. This requires that the two Servers belong to the same Layer 2 network. Conventional methods would require additional physical devices for traffic distribution and may also result in network loops and additional system and management costs.

VXLAN can be used to migrate the engineering department to Server 2. VXLAN is a network virtualization technique that uses MAC-in-UDP encapsulation. All terminal users who are reachable at Layer 3 can construct a large Layer 2 network as long as the physical network supports IP forwarding.

The migration process for the engineering department is as follows:

1. Migrate the engineering department from Server 1 to Server 2.
2. After migration, the VMs of the engineering department send gratuitous ARP or RARP packets to notify Device 2 and other devices of the migration event.
3. Device 1 receives the packets and replaces existing MAC address and ARP tables with new MAC address and ARP tables of the post-migrated VMs.

VXLAN allows the engineering department to migrate, whereas the network is unaware of it. After the engineering department is migrated from Server 1 to Server 2, tenants send gratuitous ARP or RARP packets. The MAC address and ARP tables of pre-migrated VMs saved on gateways are replaced by new MAC address and ARP tables of post-migrated VMs.

5 Configuration Notes

This section describes VXLAN configuration notes.

Involved Network Elements

You can configure VXLAN on an AC controller or in a single-node mode. Different network elements (NEs) are required for the three configuration modes. During the configuration, select a proper controller version.

Configuration Mode	Product	Description
AC controller mode	Agile Controller	Uses the NETCONF protocol to control VXLAN tunnel setup between devices and uses the OpenFlow protocol to control packet forwarding over the tunnels.
Single-node mode	No other NEs are required.	

License Support

VXLAN is a basic feature of the CE12800 series switches and is not under license control.

Version Support

Table 5-1 Products and minimum version supporting VXLAN

Series	Product Model	Minimum Version Required in AC Controller Mode	Minimum Version Required in Single-Node Mode
CE12800	CE12804/CE12808/ CE12812/CE12816	V100R006C00	V100R005C00

Series	Product Model	Minimum Version Required in AC Controller Mode	Minimum Version Required in Single-Node Mode
	CE12804S/ CE12808S	V100R006C00	V100R005C00

Feature Dependencies and Limitations

VXLAN Specifications and Performance

Table 5-2 lists VXLAN specifications.

NOTE

The values in the following table indicate the maximum values when the network transmits only the VXLAN service. If the service configurations of the real network differ from those of the test network, the values may be different from those provided here.

Table 5-2 VXLAN specifications

Item	Specifications
Number of BDs	● When a card is not working in enhanced mode: 4,000 ● When a card is working in enhanced mode: 32,000

Common VXLAN Constraints

- VXLAN deployment constraints
  - VXLAN can only be deployed on IPv4 networks.
  - In an earlier version than V200R001C00, STP cannot be configured on a user-side interface of a VXLAN tunnel. Starting from V200R001C00, STP can be configured on a user-side interface of a VXLAN tunnel that accesses the VXLAN as a VLAN.
  - It is recommended that the controller mode and single-node mode be not used simultaneously for networking. The controller mode applies on large-scale networks, and the single-node mode applies on small- and medium-scale networks.
  - The device supports only head-end replication, but not multicast replication, for broadcast, Unknown unicast, and multicast (BUM) packets.
  - CE switches do not support fragmentation or reassembly of VXLAN packets. On a VXLAN network where both CE and non-CE devices are deployed, when packets are fragmented on a non-CE device, the packets cannot be reassembled by a CE device. In this case, packet forwarding fails. To prevent this problem, you are advised to set the maximum frame length to be at most 1400 bytes on the server.
  - On the device at the VXLAN tunnel egress, all VXLAN packets cannot be redirected to interfaces.

- After all-active gateways are configured, you need to specify the non-gateway IP address as the source IP address for sending ICMP Echo Request packets when pinging the host address from a gateway.
- In V100R005C10 and earlier versions, switches can connect to the VXLAN network through Layer 2 sub-interfaces only. Starting from V100R006C00, switches can also connect to the VXLAN network through VLANs.
- Distributed gateways do not support VXLAN network access through VLANs.
- ARP broadcast suppression is not supported when VLANs are connected to the VXLAN.
- By default, the enhanced mode of the NVO3 gateway is not configured, and the device uses the loopback mode. That is, the device first loops back packets that are encapsulated with the NVO3 header, and then forwards the packets. In this situation, traffic transmitted over the NVO3 tunnel cannot be forwarded at line speed when the traffic volume is heavy. There is high probability of packet loss if the VXLAN traffic to be forwarded and encapsulated at Layer 2 or VXLAN traffic to be forwarded and encapsulated/decapsulated at Layer 3 exceeds 50% of a line card's total forwarding capacity. There is higher probability if the device functions as a VXLAN Layer 3 gateway.
- When you upgrade a device running an earlier version than V100R006C00, delete the VXLAN configuration before the upgrade if the configuration exists and the corresponding VXLAN tunnel is Down. Otherwise, functions using ACLs may not take effect after the upgrade.
- In V100R005C00, after a Layer 2 sub-interface is added to a BD, you cannot create a VBDIF interface for the BD.

  From V100R005C10, after a Layer 2 sub-interfaces with the flow encapsulation type set to **default** is added to a BD, you cannot create a VBDIF interface for the BD.
- If NVE interfaces of Layer 2 and Layer 3 modes are configured on the device simultaneously, you must specify different source VTEP IP addresses for them.
- After centralized all-active VXLAN gateways are created, you are not advised to delete or shut down the VBDIF interfaces on all these gateways. Ensure that traffic is directed to another gateway based on proper route planning before you delete or shut down a VBDIF interface on a gateway. Otherwise, some traffic may be lost.
- After distributed gateway is enabled on a Layer 3 gateway, the Layer 3 gateway discards network-side ARP messages and learns only user-side ARP messages.
- In VXLAN dual-active access scenarios, when traffic enters the peer-link interface, the switch can map IP packets based on the DSCP priority only and map non-IP packets based on the priority by the **port priority** command only.
- When one server is connected to two active access devices on the VXLAN network, you cannot run the **encapsulation** command to config the encapsulation type of the Layer 2 sub-interfaces configured on the dual-homed interfaces to **default**.
- Starting from V200R001C00, Layer 2 sub-interfaces where the encapsulation mode is QinQ can connect to the VXLAN network. When connecting these Layer 2 sub-interfaces to the VXLAN network, pay attention to the following restrictions:
  - In a dual-active VXLAN access scenario, the encapsulation mode of a Layer 2 sub-interface cannot be QinQ.
  - A static MAC address cannot be configured for a Layer 2 sub-interface where the encapsulation mode is QinQ.

        - If a Layer 2 sub-interface where the encapsulation mode is QinQ is bound to a BD, the corresponding BDIF interface cannot be created for the BD. Similarly, if a BDIF interface is created for a BD, a Layer 2 sub-interface where the encapsulation mode is QinQ cannot be bound to the BD.
        - If a Layer 2 sub-interface where the encapsulation mode is QinQ is bound to a BD, ARP broadcast suppression cannot be configured for the BD. Similarly, if ARP broadcast suppression is configured for a BD, a Layer 2 sub-interface where the encapsulation mode is QinQ cannot be bound to the BD.
        - If the **port vlan-mapping** command is configured on a main interface, a Layer 2 sub-interface where the encapsulation mode is QinQ cannot be configured on the main interface. Similarly, if a Layer 2 sub-interface where the encapsulation mode is QinQ is configured on a main interface, the **port vlan-mapping** command cannot be configured on the main interface.
        - The **rewrite pop double** command must be configured for all or none Layer 2 sub-interfaces where the encapsulation mode is QinQ in the same BD.

- Constraints on VXLAN traffic statistics collection
    - In BD-based traffic statistics, the statistics in the outbound direction for packets forwarded from the Ethernet to VXLAN network do not include the number of bytes added during VXLAN encapsulation. The statistics in the inbound direction for packets forwarded from the VXLAN to Ethernet network include the number of bytes added during VXLAN encapsulation.
    - BD-based traffic statistics in the outbound direction do not contain packets forwarded at Layer 3.
    - Traffic statistics collection for VPN instances is not supported on VBDIF interfaces.
    - There is a slight difference for outgoing traffic on the device enabled with BD-based traffic statistics in earlier versions of V100R005C10. The maximum difference of each packet is 16 bytes. This issue has been fixed in V100R005C10 and later versions.
    - Traffic statistics collection for VXLAN tunnel can only collect known unicast traffic statistics in the outbound direction.
    - Among BD traffic statistics collection, MQC traffic statistics collection, and VXLAN tunnel traffic statistics collection, the first has the highest priority and the last has the lowest priority. If two or all of them are configured, only the one with the highest priority takes effect.
    - When the enhanced mode for the NVO3 gateway is configured, VXLAN tunnel traffic statistics contain only the number of bytes of incoming packets transmitted between VXLAN networks.
    - After the enhanced mode for the NVO3 gateway is configured, BD traffic statistics in the outbound direction are inaccurate.
    - When the device collects packet statistics based on the VXLAN tunnel and VNI, it can only collect statistics on packets forwarded at Layer 2 in the outbound direction.
- Constraints between VXLAN and other features
    - If both VXLAN and other services are configured, delivery of other services may fail due to insufficient ACL resources. Only some services can be deployed when VXLAN is configured. For details, see section "Using VXLAN with Other Services" in **CE12800 ACL Resources**.
    - VXLAN cannot be configured simultaneously with GRE, SVF, TRILL, FCoE, MPLS, and multicast VPN. Starting from V100R005C00, you can configure SVF,

TRILL, or FCoE after the VXLAN configuration is deleted. However, the modification takes effect only after the device restarts.

- In V100R003C10 and V100R005C00, VXLAN takes effect only in the admin-VS.
- Starting from V100R005C10, only the admin-VS in port mode supports VXLAN; all VSs in group mode support VXLAN.
- In V100R005C10, the admin-VS shares the BD specification with all the VSs in group mode. That is, if the number of BDs created in a VS reaches the upper limit, no BD can be created in other VS. This constraint has been fixed in V100R006C00 and later versions.
- A QoS group where member interfaces are VLANs or VLANIF interfaces cannot be used with ARP broadcast packet suppression on a VXLAN network.
- After the device has VXLAN configured, the external TCAM cannot be configured to increase the multicast entry space.
- The EA series LPUs and EC LPUs at the GE rate remove the first 16 bytes from obtained or mirrored EVN and VXLAN packets.
- The device does not support MPLS encapsulation after VXLAN encapsulation, or VXLAN decapsulation after MPLS decapsulation.
- In VXLAN active-active or all-active access scenarios, devices that have VXLAN active-active or all-active access configured cannot be upgraded using ISSU.
- sFlow cannot collect inner information about VXLAN packets.
- In an earlier version than V200R001C00, NetStream cannot collect inner information about VXLAN packets. Starting from V200R001C00, NetStream can collect inner information about VXLAN packets.
- In versions earlier than V200R001C00, port isolation does not take effect for packets that are encapsulated through VXLAN tunnels. In V200R001C00 and later versions, port isolation takes effect only for Layer 2 packets that are encapsulated at the VXLAN service access side.
- Port security and DHCP snooping do not take effect on interfaces connected to the VXLAN.
- DHCP snooping can only be configured on a VXLAN network-side interface, and cannot be configured with DHCP relay simultaneously.
- Do not configure the EVN function when VNI IDs ranging from 1 to 4096 exist on the switch. Similarly, do not configure VNI IDs ranging from 1 to 4096 when the EVN function is configured.
- Starting from V100R005C10, use either of the following methods on the NVO3-enabled device to prevent service overlay failures:
  - Run the **assign forward nvo3 acl extend enable** command to enable the ACL extension function and then restart the device.
  - If the network-side interface of the VXLAN tunnel or the EVN tunnel is not an Eth-Trunk interface, run the **assign forward nvo3 eth-trunk hash disable** command to disable the Eth-Trunk interface from load balancing NVO3 packets in optimized mode.
- The VLAN, VXLAN, carrier VLAN, main interface, and Eth-Trunk where card interoperability mode is set to enhanced share system resources. If system resources are insufficient, the configuration may fail.

Specific Constraints in AC Controller Mode

- When VMs are online, do not run commands on the device to modify configurations delivered by the AC controller; otherwise, the VXLAN service cannot run properly. For example, do not run commands to delete the BD, cancel the mapping between the VNI and BD, modify the VTEP IP address, delete the VBDIF interface of a Layer 3 gateway, or modify the IP address of the VBDIF interface when VMs are online.

6 Configuring VXLAN (Through the Agile Controller-DCN)

This section describes how to configure VXLAN using the Agile Controller-DCN.

Prerequisites

Before configuring VXLAN through the Agile Controller-DCN, complete the following tasks:

- Ensuring reachable routes between the devices
- Configure the devices to communicate with the Agile Controller-DCN through SNMP and NETCONF. For details, see **SNMP Configuration** and **NETCONF Configuration**.
- Running the **ip tunnel mode vxlan** command on the device to set the tunnel mode to VXLAN.
- Running the **assign forward nvo3 acl extend enable** command on the device to enable the NVO3 ACL extension function.

Context

After NETCONF is configured on a device, the Agile Controller-DCN (NETCONF Manager) can use the NETCONF protocol to manage the device (NETCONF Agent). Other VXLAN-related configuration and entries except those described in this section are delivered to the device through the Agile Controller-DCN. For details about the Agile Controller-DCN configuration, see corresponding documents.

NOTICE

When VMs are online, do not run commands on the device to modify configurations delivered by the Agile Controller-DCN; otherwise, the VXLAN service cannot run properly. For example, do not run commands to delete the BD, cancel the mapping between the VNI and BD, modify the VTEP IP address, delete the VBDIF interface of a Layer 3 gateway, or modify the IP address of the VBDIF interface when VMs are online.

Procedure

- **Table 6-1** lists the functions that can only be configured on the switch but cannot be delivered by the Agile Controller-DCN.

Table 6-1 Functions that can only be configured on the switch but cannot be delivered by the Agile Controller-DCN

Function	Deployment Position	Configuration Procedure	Remarks
(Optional) Configure all-active gateways for the VXLAN network.	Centralized all-active VXLAN gateway	1. Run the **system-view** command to enter the system view. 2. Run the **dfs-group** *dfs-group-id* command to create a DFS group and enter the DFS group view. By default, no DFS group is created. 3. Run the **source ip** *ip-address* command to bind an IPv4 address to the DFS group. By default, no IPv4 address is bound to the DFS group. 4. (Optional) Run the **udp port** *port-number* command to specify a UDP port number for the DFS group. By default, the UDP port number of the DFS group is 61467. 5. Run the **active-active-gateway** command to create all-active gateways and enter the all-active gateway view. By default, no all-active gateway is created. 6. Run the **peer** *ip-address* [**vpn-instance** *vpn-instance-name*] command to configure the IP address of an all-active gateway peer. By default, no all-active gateway peer is configured. 7. Run the **commit** command to commit the configuration.	This step is required when you want to deploy centralized all-active VXLAN gateways for the network. When configuring centralized all-active VXLAN gateways, ensure that the VTEP source IP address, ingress replication list, and IP address and MAC address of the VBDIF interface are the same on all the all-active gateways.

Function	Deployment Position	Configuration Procedure	Remarks
(Optional) Configure the VXLAN gateway to work in non-loopback mode.	VXLAN gateway	By default, a Layer 3 VXLAN gateway works in loopback mode. That is, the Layer 3 VXLAN gateway first loops back VXLAN-encapsulated packets, and then forwards the packets. In this situation, traffic transmitted over the VXLAN tunnel cannot be forwarded at line speed when the traffic volume is heavy. To solve the preceding problem, configure the Layer 3 VXLAN gateway to work in non-loopback mode. For details, see **7.12 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode**.	-
(Optional) Optimize ACL resource usage on the VXLAN.	All VXLAN nodes	If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, BD traffic statistics collection, and DHCP on a switch deployed with VXLAN services, there is high probability that the other services fail to be configured. In this case, use either of the following methods to reduce the probability of service deployment failures. ● Enabling the VXLAN service extension function ● Disabling an Eth-Trunk from load balancing VXLAN packets in optimized mode For details, see **7.10 (Optional) Optimizing ACL Resource Usage on the VXLAN**.	-

Function	Deployment Position	Configuration Procedure	Remarks
(Optional) Optimize load balancing on the VXLAN network.	All VXLAN nodes	On a VXLAN network, VXLAN packets can be load balanced through ECMP or Eth-Trunks. To enable load balancing or improve the load balancing effect, enable either of the following functions: ● Enable load balancing of VXLAN packets through ECMP in optimized mode. ● Enable an Eth-Trunk to load balance VXLAN packets in optimized mode. For details, see **7.11 (Optional) Optimizing Load Balancing on the VXLAN Network**.	This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.
(Optional) Enable the VXLAN path detection function.	All VXLAN nodes	1. Run the **system-view** command to enter the system view. 2. Run the **vxlan path detection enable** command to enable the VXLAN path detection function. By default, the VXLAN path detection function is disabled. 3. Run the **commit** command to commit the configuration.	This step is required if the Agile Controller-DCN of V200R001C00SPC200 is used and needs to perform VTEP all path detection and VTEP traffic simulation detection.

----End

7 Configuring VXLAN in Single-Node, Centralized Gateway, and Static Mode

About This Chapter

When VXLAN in centralized gateway mode for static tunnel establishment is deployed, traffic across network segments is forwarded through Layer 3 VXLAN gateways to implement centralized traffic management.

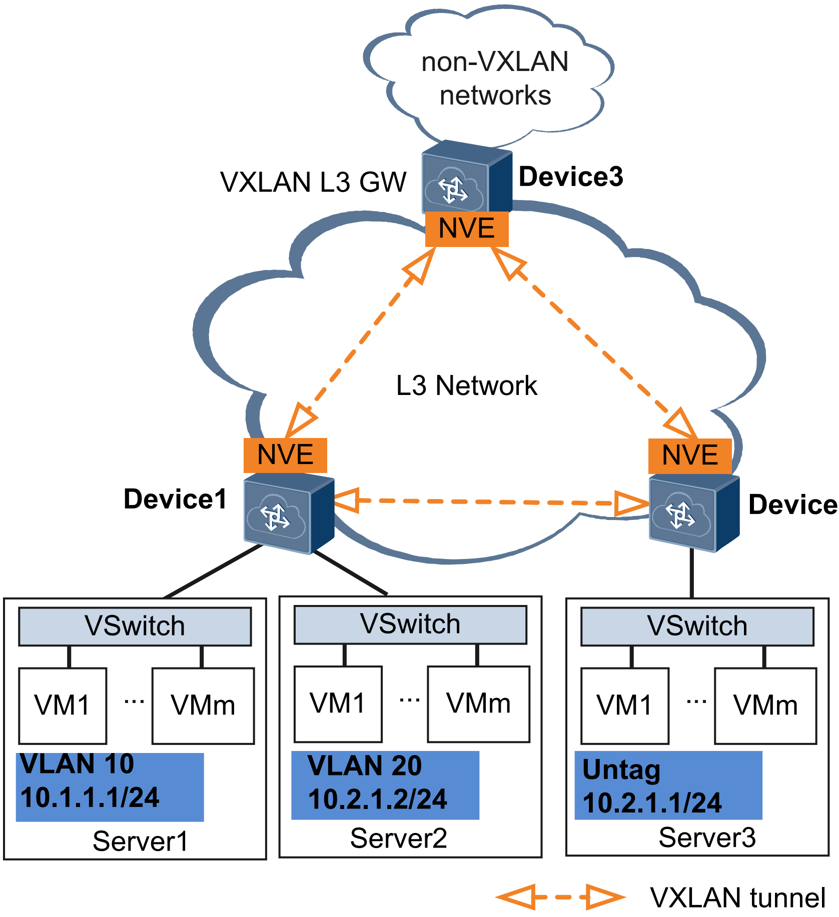
Usage Scenario

An enterprise has allocated VMs in different locations to a tenant. Some of the VMs reside on the same network segment, and the others reside on different network segments. To allow communication between VMs, deploy Layer 2 and Layer 3 VXLAN gateways and establish VXLAN tunnels.

On the network shown in **Figure 7-1**, Server 2 and Server 3 belong to the same network segment and access the VXLAN through Device 1 and Device 2, respectively; Server 1 and Server 2 belong to different network segments and both access the VXLAN through Device 1.

- To allow VM 1 on Server 2 and VM 1 on Server 3 to communicate, deploy Layer 2 VXLAN gateways on Device 1 and Device 2 and establish a VXLAN tunnel between Device 1 and Device 2 so that tenants on the same network segment can communicate.
- To allow VM 1 on Server 1 and VM 1 on Server 3 to communicate, deploy a Layer 3 VXLAN gateway on Device 3 and establish a VXLAN tunnel between Device 1 and Device 3 and between Device 2 and Device 3 so that tenants on different network segments can communicate.

Figure 7-1 VXLAN in centralized gateway mode



Pre-configuration Tasks

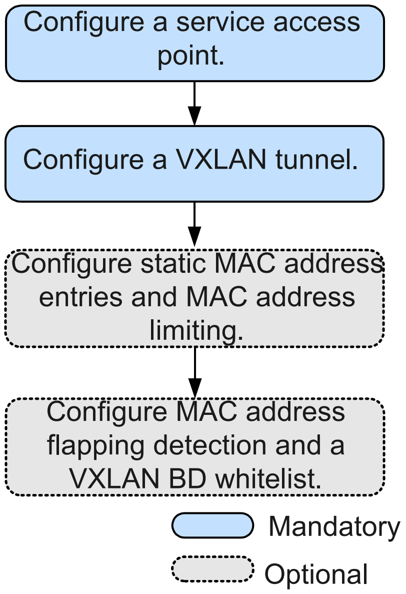
Before configuring VXLAN in centralized gateway mode for static tunnel establishment, ensure that the network is reachable at Layer 3.

Configuration Procedures

NOTE

In a dual-active VXLAN access scenario, two access devices to which a host is dual homed are simulated as a VTEP to prevent loops or MAC address flapping. In this case, ensure that **VXLAN access point** and **VXLAN tunnel configuration** on the two devices are the same.

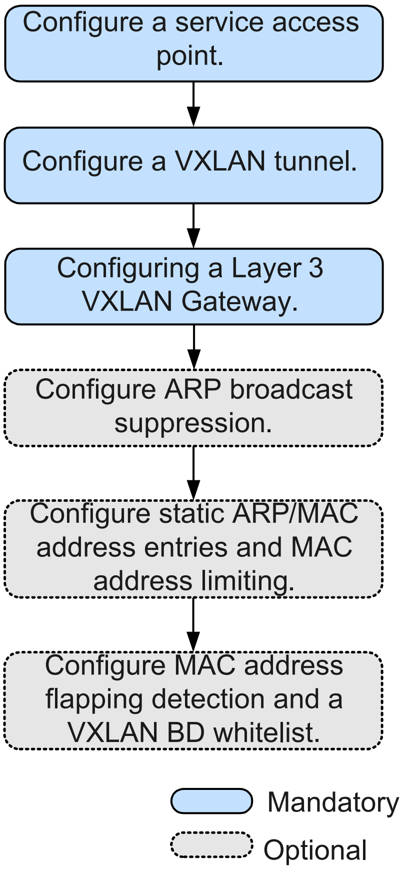
Figure 7-2 Flowchart for configuring intra-segment communication through centralized VXLAN gateways


Configure a service access point.
Configure a VXLAN tunnel.
Configure static MAC address entries and MAC address limiting.
Configure MAC address flapping detection and a VXLAN BD whitelist.
Mandatory
Optional

NOTE

To implement intra-segment communication through centralized VXLAN gateways, configure only static MAC address entries and MAC address limiting described in **7.8 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting**.

Figure 7-3 Flowchart for configuring inter-segment communication through centralized VXLAN gateways


Configure a service access point.
Configure a VXLAN tunnel.
Configuring a Layer 3 VXLAN Gateway.
Configure ARP broadcast suppression.
Configure static ARP/MAC address entries and MAC address limiting.
Configure MAC address flapping detection and a VXLAN BD whitelist.
Mandatory
Optional

7.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function

7.2 Configuring a Service Access Point
A VXLAN service access point can be a Layer 2 sub-interface or VLAN.

7.3 Configuring a VXLAN Tunnel
VXLAN uses MAC-in-UDP encapsulation to extend Layer 2 networks, allowing a large number of tenant accesses to virtual networks.

7.4 Configuring a Layer 3 VXLAN Gateway
To allow users on different network segments to communicate, a Layer 3 VXLAN gateway must be deployed, and the default gateway address of the users must be the IP address of the VBDIF interface of the Layer 3 gateway.

7.5 (Optional) Configuring Centralized All-Active Gateways for the VXLAN Network

7.6 (Optional) Configuring ARP Broadcast Suppression
When tenants communicate with each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks and may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on Layer 2 VXLAN gateways.

7.7 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages

7.8 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting

Static ARP entries or MAC address entries can be configured for traffic forwarding, and MAC address limiting can be configured to improve VXLAN security.

7.9 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection
In specific VXLAN applications, when a device connects to a load balancing server equipped with two network interface cards, the server's MAC address may be learned by two interfaces on the device. This is a normal situation where MAC address flapping detection is not needed. In this case, configure a VXLAN BD whitelist for MAC address flapping detection.

7.10 (Optional) Optimizing ACL Resource Usage on the VXLAN

7.11 (Optional) Optimizing Load Balancing on the VXLAN Network

7.12 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode

7.13 Checking the Configurations
After configuring VXLAN in centralized gateway mode for static tunnel establishment, check VXLAN tunnel, VNI, and VBDIF interface information.

7.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function

Context

- Configuring a tunnel mode: You need to set the tunnel mode to VXLAN when configuring the VXLAN feature; otherwise, the configurations do not take effect.
- Enabling the VXLAN ACL extension function: By default, the VXLAN ACL extension function is disabled on the device. If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, and BD traffic statistics collection, on the device deployed with VXLAN services, there is high probability that the other services fail to be configured. You can enable the VXLAN ACL extension function to lower the configuration failure probability.

To ensure normal forwarding of VXLAN packets, the VXLAN tunnel mode must have been configured and the VXLAN ACL extension function must have been enabled on Layer 2 and Layer 3 VXLAN gateways.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip tunnel mode vxlan
```

The tunnel mode is set to VXLAN.

By default, the tunnel mode is VXLAN.

NOTE

Only the Admin-VS in port mode supports this command, and all VSs in group mode support this command.

Step 3 Run:

```
assign forward nvo3 acl extend enable
```

The VXLAN ACL extension function is enabled.

By default, the VXLAN ACL extension function is disabled.

NOTE

You can run this command on the admin-VS only. The configuration takes effect for all the VSs.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Follow-up Procedure

After configuring the VXLAN tunnel mode and enabling the VXLAN ACL extension function, you need to save the configuration and restart the switch to make the configuration take effect.

7.2 Configuring a Service Access Point

A VXLAN service access point can be a Layer 2 sub-interface or VLAN.

Context

When a Layer 2 sub-interface is used as a service access point, different encapsulation types can be configured for the sub-interface to transmit various types of data packets. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD. Table 7-1 describes the different encapsulation types.

Table 7-1 Traffic encapsulation types

Traffic Encapsulation Type	Description
dot1q	If a Dot1q sub-interface receives a single-tagged VLAN packet, the sub-interface forwards only the packet with a specifie VLAN ID. If a Dot1q sub-interface receives a double-tagged VLAN packet, the sub-interface forwards only the packet with a specified outer VLAN ID. ● When performing VXLAN encapsulation on packets, a Dot1q Layer 2 sub-interface removes the outer tags of the packets. ● When performing VXLAN decapsulation on packets, a Dot1q Layer 2 sub-interface replaces the VLAN tags with specified VLAN tags if the inner packets carry VLAN tags, or adds specified VLAN tags to the packets if the inner packets do not carry VXLAN tags. When setting the encapsulation type to **dot1q** for a Layer 2 sub-interface, note the following: ● The VLAN IDs specified for the Layer 2 sub-interface cannot be the same as either the VLAN IDs of packets allowed to pass through the corresponding Layer 2 interfaces or the MUX VLAN IDs. ● Layer 2 and Layer 3 sub-interfaces cannot have the same VLAN IDs specified.

Traffic Encapsulation Type	Description
untag	An untagged Layer 2 sub-interface receives only packets that do not carry VLAN tags. ● When performing VXLAN encapsulation on packets, an untagged Layer 2 sub-interface does not add any VLAN tag to the packets. ● When performing VXLAN decapsulation on packets, an untagged Layer 2 sub-interface removes the VLAN tags of single-tagged inner packets or the outer VLAN tags of double-tagged inner packets. When setting the encapsulation type to **untag** for a Layer 2 sub-interface, note the following: ● Ensure that the corresponding physical interface of the sub-interface does not have any configuration, and is removed from the default VLAN. ● Untagged Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● An interface can have only one untagged Layer 2 sub-interface configured.
qinq	A QinQ sub-interface receives only tagged packets with specified inner and outer VLAN tags. ● When performing VXLAN encapsulation on packets, a QinQ sub-interface removes two VLAN tags from packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintains the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. ● When performing VXLAN decapsulation on packets, a QinQ sub-interface adds two specific VLAN tags to packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintain the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. **NOTE** The traffic behavior for QinQ interfaces bound to the same BD must be the same. QinQ interfaces do not support DHCP Snooping or VBDIF and cannot be bound to the same BD as Dot1q sub-interfaces. A QinQ interface can have only one outer VLAN tag and one inner VLAN tag.

Traffic Encapsulation Type	Description
default	A default Layer 2 sub-interface receives all packets, irrespective of whether the packets carry VLAN tags. When performing VXLAN encapsulation and decapsulation on packets, a default Layer 2 sub-interface does not process VLAN tags of the packets. When setting the encapsulation type to **default** for a Layer 2 sub-interface, note the following: ● Ensure that the interface for the Layer 2 sub-interface is not added to any VLAN. ● Default Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● If a default Layer 2 sub-interface is created for an interface, the interface cannot have other types of Layer 2 sub-interfaces configured.

NOTE

When a sub-interface that is configured with dot1q and QinQ receives double-tagged VLAN packets, the QinQ sub-interface preferentially processes the packets. For example, if a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of 20, the QinQ sub-interface preferentially processes the packet. If a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of non-20, the dot1q sub-interface preferentially processes the packet.

If a VLAN is used as a service access point, it can be bound to a BD for data packets in the VLAN to be transmitted through this BD.

Configure a service access point on a Layer 2 gateway.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bridge-domain bd-id
```

A BD is created, and the BD view is displayed.

By default, no BD is created.

Step 3 (Optional) Run:

```
description description
```

A description is configured for the BD.

By default, no description is configured for a BD.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Configure a service access point.

- Configure a VLAN as a service access point.

  NOTE

  – In a distributed gateway scenario, a VLAN cannot be configured as a VXLAN service access point.

  – You cannot bind a VLAN to a BD while configuring the ARP broadcast suppression function. After a VLAN is configured as a VXLAN service access point, do not configure ARP broadcast suppression.

  – After a VLAN is bound to a BD, the BD becomes the broadcast domain. Therefore, other service configurations such as DHCP Snooping and IGMP Snooping in the VLAN become invalid.

  – VLAN and BD use 1:1 mapping. That is, a VLAN can be bound to only one BD, and only one VLAN can be bound to a BD.

  a. Run:

  ```
  bridge-domain bd-id
  ```

  The view of an existing BD is displayed.

  b. Run:

  ```
  l2 binding vlan vlan-id
  ```

  A global VLAN is bound to the BD.

  By default, VLANs are not bound to any BD.

  NOTE

  Before performing this step, ensure that a global VLAN has been created. After binding the global VLAN to the BD, add the related device interfaces to the VLAN.

  c. Run:

  ```
  commit
  ```

  The configuration is committed.

- Configure a Layer 2 sub-interface as a service access point.

  a. Run:

  ```
  interface interface-type interface-number.subnum mode l2
  ```

  A Layer 2 sub-interface is created, and the sub-interface view is displayed.

  By default, no Layer 2 sub-interface is created.

  NOTE

  Before running this command, ensure that the Layer 2 interface for which a Layer 2 sub-interface is created does not have the **port link-type dot1q-tunnel** command configuration. If this configuration exists, run the **undo port link-type** command to delete the configuration.

  b. Run:

  ```
  encapsulation { dot1q [ vid ce-vid ] | default | untag  | qinq [ vid pe-
  vid ce-vid ce-vid ] }
  ```

  An encapsulation type is configured for the Layer 2 sub-interface.

  By default, no encapsulation type is configured for Layer 2 sub-interfaces.

  c. (Optional) Run:

  ```
  rewrite pop double
  ```

The sub-interface is enabled to remove double VLAN tags from received packets if the encapsulation type of the sub-interface is set to QinQ in **Step 5.b**.

By default, a Layer 2 sub-interface with the encapsulation type being QinQ is enabled to transparently transmit received packets.

d. Run:

```
bridge-domain bd-id
```

The Layer 2 sub-interface is added to a BD so that the sub-interface can transmit data packets through this BD.

By default, Layer 2 sub-interfaces are not added to any BD.

NOTE

After a Layer 2 sub-interface with the flow encapsulation type set to **default** is added to a BD, you cannot create a VBDIF interface for the BD.

e. Run:

```
commit
```

The configuration is committed.

----End

7.3 Configuring a VXLAN Tunnel

VXLAN uses MAC-in-UDP encapsulation to extend Layer 2 networks, allowing a large number of tenant accesses to virtual networks.

Context

After you configure local and remote VNIs and VTEP's IP addresses, a VXLAN tunnel is statically created. This configuration is simple, as no protocols are involved.

To ensure VXLAN packet forwarding, VXLAN tunnels must be configured on both Layer 2 and Layer 3 VXLAN gateways.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bridge-domain bd-id
```

The BD view is displayed.

Step 3 Run:

```
vxlan vni vni-id
```

A VNI is created and mapped to the BD.

By default, no VNI is created.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Run:

```
interface nve nve-number
```

An NVE interface is created, and the NVE interface view is displayed.

By default, no NVE interface is created.

Step 6 Run:

```
source ip-address
```

An IP address is configured for the source VTEP.

By default, no IP address is configured for a source VTEP.

Either a physical interface's IP address or loopback interface address can be specified for a source VTEP. Using the loopback interface address as the source VTEP's IP address is recommended.

Step 7 Run:

```
vni vni-id head-end peer-list ip-address &<1-10>
```

An ingress replication list is configured.

By default, no ingress replication list is configured.

NOTE

BUM packet forwarding is implemented only using ingress replication. To establish a VXLAN tunnel between a Huawei device and a non-Huawei device, ensure that the non-Huawei device also has ingress replication configured. Otherwise, communication fails.

Step 8 Run:

```
commit
```

The configuration is committed.

----End

7.4 Configuring a Layer 3 VXLAN Gateway

To allow users on different network segments to communicate, a Layer 3 VXLAN gateway must be deployed, and the default gateway address of the users must be the IP address of the VBDIF interface of the Layer 3 gateway.

Context

A tenant is identified by a VNI. VNIs can be mapped to BDs in 1:1 mode so that a BD can function as a VXLAN network entity to transmit VXLAN data packets. A VBDIF interface is a Layer 3 logical interface created for a BD. After an IP address is configured for a VBDIF interface of a BD, the VBDIF interface can function as the gateway for tenants in the BD for Layer 3 forwarding. VBDIF interfaces allow Layer 3 communication between VXLANs on different network segments and between VXLANs and non-VXLANs, and implement Layer 2 network access to a Layer 3 network.

VBDIF interfaces are configured on Layer 3 VXLAN gateways for inter-segment communication, and are not needed in the case of intra-segment communication.

NOTE

The DHCP relay function can be configured on the VBDIF interface so that hosts can request IP addresses from the external DHCP server.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface vbdif bd-id
```

A VBDIF interface is created, and the VBDIF interface view is displayed.

Step 3 Run:

```
ip address ip-address { mask | mask-length } [ sub ]
```

An IP address is configured for the VBDIF interface to implement Layer 3 interworking.

By default, no IP address is configured for interfaces.

Step 4 (Optional) Run:

```
mac-address mac-address
```

A MAC address is configured for the VBDIF interface.

By default, the MAC address of a VBDIF interface is the system MAC address.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

7.5 (Optional) Configuring Centralized All-Active Gateways for the VXLAN Network

Context

On a traditional network, Virtual Router Redundancy Protocol (VRRP) is used to protect gateways. One gateway is in the active state and the others in the standby state, leading to low gateway utilization. When a gateway fails, the new active gateway needs to be reelected, and the convergence performance upon a gateway fault is low.

After you configure all-active VXLAN gateways, the multiple VXLAN Layer 3 gateways can be virtualized into one VXLAN gateway and traffic is forwarded through any gateway. The all-active VXLAN gateway function improves gateway utilization and convergence performance.

Perform the following operations on the Layer 3 VXLAN gateway.

NOTE

- In the VXLAN centralized all-active gateways networks, if the uplink of spine fails, user-side traffic may fail to be forwarded and therefore are discarded. To prevent this problem, associate uplink and downlink interfaces with the Monitor Link group. When the uplink interface becomes Down, the downlink interface also becomes Down. This prevent user-side traffic from being discarded. For details about the monitor-link configuration, see Configuring the Uplink and Downlink Interfaces in a Monitor Link Group.
- After deploying VXLAN centralized all-active gateways, you may need to reset the device after a device upgrade or patch installation. Pay attention to the following points:
  - Before the reset: On the access side, tear down the equal-cost multi-path routing (ECMP) paths between the access device and the VTEP address of the Spine device to decrease the priority of the advertised VTEP host route. This prevents the access device from forwarding traffic to the spine device. On the network side, decrease the priority of the VXLAN gateway route advertised by the device to prevent the upstream device from forwarding traffic to the device.
  - After the reset: Synchronize ARP entries and then restore the original priorities of the VTEP host route and the VXLAN gateway route, to avoid the traffic loss caused by insufficient ARP entries. For example, it takes approximately 20 minutes to synchronize 128K ARP entries.
- After you shut down or delete the VBDIF interface on the all-active gateway, the traffic cannot be switched to another gateway. The access device forwards traffic to this device based on the ECMP route of the VTEP, but this device cannot forward traffic to another all-active gateway, causing traffic loss.
- To prevent traffic forwarding before ARP entry synchronization when the device is reset, it is recommended that you configure the delayed route advertisement mechanism. Take OSPF as an example. Run the command **stub-router on-startup** [*interval*] **include-stub** in the OSPF process related to the VTEP and VBDIF routes to configure the stub server interval for the device when the device restarts or fails (for example, 20 minutes for 128K ARP entries).

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Configure the Layer 3 VXLAN gateway function on switches that work as all-active gateways. For details, see **7.3 Configuring a VXLAN Tunnel** and **7.4 Configuring a Layer 3 VXLAN Gateway**. The following table lists the commands.

Procedure	Command	Description
Assign a source IP address to the VTEP.	**source** *ip-address*	Ensure that the gateways have the same VTEP source IP address.
Assign an ingress replication list for a VNI.	**vni** *vni-id* **head-end peer-list** *ip-address* &<1-10>	Ensure that the gateways have the same ingress replication for a VNI. NOTE BUM packet forwarding is implemented only using ingress replication. To establish a VXLAN tunnel between a Huawei device and a non-Huawei device, ensure that the non-Huawei device also has ingress replication configured. Otherwise, communication fails.

Procedure	Command	Description
Assign an IP address to each VBDIF interface.	**ip address** *ip-address* { *mask* | *mask-length* } [**sub**]	Ensure that the VBDIF interfaces of the gateways have the same IP address.
Assign a MAC address to each VBDIF interface.	**mac-address** *mac-address*	Ensure that the VBDIF interfaces of the gateways have the same MAC address.

Step 3 If you want to configure the all-active gateway function on the device, configure a DFS group to synchronize ARP entries between all-active gateways.

1. Run:

   ```
   dfs-group dfs-group-id
   ```

   A DFS group is created and its view is displayed.

   By default, no DFS group is created.

2. Run:

   ```
   source ip ip-address
   ```

   The DFS group is bound to an IPv4 address.

   By default, a DFS group is not bound to any IPv4 address.

3. (Optional) Run:

   ```
   udp port port-number
   ```

   The UDP port number of the DFS group is set.

   By default, the UDP port number of the DFS group is 61467.

4. Run:

   ```
   active-active-gateway
   ```

   An all-active gateways is created and its view is displayed.

   By default, no all-active gateway is created.

5. Run:

   ```
   peer ip-address [ vpn-instance vpn-instance-name ]
   ```

   The all-active gateway peer is configured.

   By default, no all-active gateway peer is configured.

   NOTE

   You can specify a maximum of fifteen IP addresses for all-active gateway peers in the all-active gateway view of a DFS group.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.6 (Optional) Configuring ARP Broadcast Suppression

When tenants communicate with each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks and may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on Layer 2 VXLAN gateways.

Context

After you enable ARP broadcast suppression on Layer 2 VXLAN gateway, configure Ethernet Virtual Network Border Gateway Protocol (EVN BGP) on Layer 2 and Layer 3 VXLAN gateways to allow ARP broadcast suppression to take effect. EVN BGP can then generate host information based on learned ARP entries. If a Layer 3 VXLAN gateway is enabled in the VBDIF interface view to use EVN BGP to advertise information to Layer 2 VXLAN gateways, the Layer 3 VXLAN gateway will advertise the host information generated by EVN BGP to Layer 2 VXLAN gateways. After the Layer 2 VXLAN gateways receive ARP broadcast packets, they convert the ARP broadcast packets into unicast packets based on the learned host information before forwarding the packets out. This decreases the number of broadcast packets in a BD, improving network performance.

Configuring BGP RRs is recommended to simplify BGP configuration. Layer 3 VXLAN gateways are generally used as RRs, and Layer 2 VXLAN gateways as RR clients.

Procedure

Step 1 Configure EVN BGP on Layer 2 and Layer 3 VXLAN gateways to establish EVN BGP peer relationships.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   evn bgp
   ```

   The EVN BGP view is displayed.

3. Run:

   ```
   source-address ip-address
   ```

   A source address is configured for establishing an EVN BGP peer relationship. This source address can be used to generate a router ID, a next-hop address, and an EVN instance's RD.

   By default, no source address is specified for establishing an EVN BGP peer relationship.

4. Run:

   ```
   peer ip-address
   ```

   An EVN BGP peer address is specified.

   By default, no EVN BGP peer is specified.

   If a BGP RR is deployed, each VXLAN gateway only needs to establish an EVN BGP peer relationship with the RR.

5. Run:

   ```
   commit
   ```

The configuration is committed.

Step 2 Configure a Layer 3 VXLAN gateway as a BGP RR.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   evn bgp
   ```

   The EVN BGP view is displayed.

3. Configure a dynamic or static EVN BGP RR. Dynamic and static RRs cannot coexist. Determine which type of RR to configure based on actual requirements.

   - To configure a dynamic RR, run the **server enable** command.

     By default, no dynamic RR is configured.

     After a dynamic RR is configured, all devices that establish EVN BGP peer relationships with the RR become the dynamic RR clients.

   - To configure a static RR, run the **peer** *ipv4-address* **reflect-client** command.

     By default, no static RR is configured.

     Only the specified peer can become a static RR client.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 3 Enable EVN BGP on a Layer 3 VXLAN gateway to collect host information.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   interface vbdif bd-id
   ```

   The VBDIF interface view is displayed.

3. Run:

   ```
   arp collect host enable
   ```

   EVN BGP is enabled to collect host information.

   By default, EVN BGP is disabled from collecting host information.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 4 Enable Layer 2 and Layer 3 VXLAN gateways to use EVN BGP to advertise host information.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   host collect protocol bgp
   ```

   EVN BGP is enabled to advertise host information.

By default, EVN BGP is disabled from advertising host information.

3. Run:

```
commit
```

The configuration is committed.

Step 5 Enable ARP broadcast suppression on a Layer 2 VXLAN gateway.

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
bridge-domain bd-id
```

The BD view is displayed.

3. Run:

```
arp broadcast-suppress enable
```

ARP broadcast suppression is enabled.

By default, ARP broadcast suppression is disabled.

4. Run:

```
commit
```

The configuration is committed.

----End

7.7 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages

Context

When the device wants to communicate with another device in the same network segment, it queries ARP entries to direct packet forwarding. If the device fails to find the corresponding ARP entry from the forwarding plane, it sends an ARP Miss message to the CPU. The ARP Miss message will trigger the device to send an ARP broadcast packet to start ARP learning. In some cases, customers may want to limit the number of broadcast packets on the VXLAN network. You can then disable a VBDIF interface from sending ARP Miss messages to achieve this purpose.

After a VBDIF interface is disabled from sending ARP Miss messages, the device cannot learn ARP entries from this VBDIF interface, so ARP entries must be manually configured on it.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface vbdif bd-id
```

A VBDIF interface is created and the VBDIF interface view is displayed.

Step 3 Run:

```
arp miss disable
```

The VBDIF interface is disabled from sending ARP Miss messages.

By default, a VBDIF interface can send ARP Miss messages.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

7.8 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting

Static ARP entries or MAC address entries can be configured for traffic forwarding, and MAC address limiting can be configured to improve VXLAN security.

Context

- Static ARP entries are manually configured and maintained. They can be neither aged nor overwritten by dynamic ARP entries. Therefore, configuring static ARP entries on Layer 3 VXLAN gateways enhances communication security. If a static ARP entry is configured on a device, the device can communicate with a peer device that has a specified IP address using only the specified MAC address. Network attackers cannot modify the mapping between the IP and MAC addresses, which ensures communication between the two devices.
- After the source NVE on a VXLAN tunnel receives broadcast, unknown unicast, and multicast (BUM) packets, the local VTEP sends a copy of the BUM packets to every VTEP in the ingress replication list. Configuring static MAC address entries helps reduce broadcast traffic and prevent unauthorized data access from bogus users.
- The maximum number of MAC addresses that a device can learn can be configured to limit the number of access users and prevent against attacks on MAC address tables. If the device has learned the maximum number of MAC addresses allowed, no more addresses can be learned. The device can also be configured to discard packets after learning the maximum allowed number of MAC addresses, improving network security.
- If Layer 3 VXLAN gateway does not need to learn MAC addresses of packets in a BD, MAC address learning can be disabled from the BD to conserve MAC address entry resources. If the network topology of a VXLAN becomes stable and MAC address entry learning is complete, MAC address learning can also be disabled.

Configuring static MAC address entries and MAC address limiting applies to Layer 2 VXLAN gateways; configuring static ARP entries applies to Layer 3 VXLAN gateways; disabling MAC address limiting applies to both Layer 2 and Layer 3 VXLAN gateways.

Procedure

- Configure a static ARP entry.
  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

b. Run:

```
arp static ip-address mac-address vni vni-id source-ip source-ip peer-ip
peer-ip
```

A static ARP entry is configured.

By default, no static ARP entry is configured.

NOTE

ip-address must belong to the same network segment as the Layer 3 gateway's IP address.

c. Run:

```
commit
```

The configuration is committed.

- Configure a static MAC address entry.

a. Run:

```
system-view
```

The system view is displayed.

b. Run:

```
mac-address static mac-address bridge-domain bd-id source source-ip-
address peer peer-ip vni vni-id
```

A static MAC address entry is configured.

By default, no static MAC address entry is configured.

c. Run:

```
commit
```

The configuration is committed.

- Configure MAC address limiting.

a. Run:

```
system-view
```

The system view is displayed.

b. Run:

```
bridge-domain bd-id
```

The BD view is displayed.

c. Run:

```
mac-address limit { action { discard | forward } | maximum max | alarm
{ disable | enable } } *
```

MAC address limiting is configured.

By default, MAC address limiting is not configured.

d. Run:

```
commit
```

The configuration is committed.

- Disable MAC address learning.

a. Run:

```
system-view
```

The system view is displayed.

b. Run:

```
bridge-domain bd-id
```

The BD view is displayed.

c. Run:

```
mac-address learning disable
```

MAC address learning is disabled.

By default, MAC address learning is enabled for a BD.

d. Run:

```
commit
```

The configuration is committed.

----End

7.9 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection

In specific VXLAN applications, when a device connects to a load balancing server equipped with two network interface cards, the server's MAC address may be learned by two interfaces on the device. This is a normal situation where MAC address flapping detection is not needed. In this case, configure a VXLAN BD whitelist for MAC address flapping detection.

Context

By default, MAC address flapping detection is enabled globally. After a BD is added to a MAC address flapping detection whitelist, detection is not performed for this BD. Even if MAC address flapping occurs in the BD, the occurrence generates neither an alarm nor a record.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
mac-address flapping detection exclude bridge-domain bd-id1 [ to bd-id2 ]
```

A VXLAN BD whitelist for MAC address flapping detection is configured.

By default, no VXLAN BD whitelist for MAC address flapping detection is configured.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

7.10 (Optional) Optimizing ACL Resource Usage on the VXLAN

Context

If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, BD traffic statistics collection, and DHCP on a switch deployed with VXLAN services, there is high probability that the other services fail to be configured. In this case, use either of the following methods to reduce the probability of service deployment failures.

- Enabling the VXLAN service extension function:

  This function takes effect only on VXLAN-enabled cards except the CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA series.

  NOTE

  After this function is enabled, only CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA can forward packets with 230 to 294 bytes received from the VXLAN tunnels to cards of the same series.

- Disabling an Eth-Trunk from load balancing VXLAN packets in optimized mode:
  - You are advised to disable load balancing of VXLAN packets in optimized mode when neither the user side nor the network side of a VXLAN tunnel is an Eth-Trunk.
  - You can disable load balancing of VXLAN packets in optimized mode when either the user side or the network side of a VXLAN tunnel is an Eth-Trunk. After this function is disabled, an Eth-Trunk does not evenly load balance VXLAN packets or does not perform load balancing.

  NOTE

  This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.

Procedure

Step 1 Enable the VXLAN service extension function.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 service extend enable
   ```

   The VXLAN service extension function is enabled.

   By default, the VXLAN service extension function is disabled.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Disable an Eth-Trunk from load balancing VXLAN packets in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 eth-trunk hash disable
   ```

   An Eth-Trunk is disabled from load balancing VXLAN packets in optimized mode.

   By default, an Eth-Trunk load balances VXLAN packets in optimized mode.

3. Run:

```
commit
```

The configuration is committed.

----End

7.11 (Optional) Optimizing Load Balancing on the VXLAN Network

Context

On a VXLAN network, VXLAN packets can be load balanced through ECMP or Eth-Trunks. To enable load balancing or improve the load balancing effect, enable either of the following functions:

- Enable load balancing of VXLAN packets through ECMP in optimized mode.
- Enable an Eth-Trunk to load balance VXLAN packets in optimized mode.

NOTE

This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.

Procedure

Step 1 Enable load balancing of VXLAN packets through ECMP in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 ecmp hash enable
   ```

   Load balancing of VXLAN packets through ECMP in optimized mode is enabled.

   By default, load balancing of VXLAN packets through ECMP in optimized mode is disabled.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Enable an Eth-Trunk to load balance VXLAN packets in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 eth-trunk hash enable
   ```

   An Eth-Trunk is enabled to load balance VXLAN packets in optimized mode.

   By default, an Eth-Trunk is disabled from load balancing VXLAN packets in optimized mode.

3. Run:

   ```
   commit
   ```

The configuration is committed.

----End

7.12 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode

Context

By default, a Layer 3 VXLAN gateway works in loopback mode. That is, the Layer 3 VXLAN gateway first loops back VXLAN-encapsulated packets, and then forwards the packets. In this situation, traffic transmitted over the VXLAN tunnel cannot be forwarded at line speed when the traffic volume is heavy.

To solve the preceding problem, configure the Layer 3 VXLAN gateway to work in non-loopback mode.

- Layer 2 non-loopback mode: The device determines whether to loop back the received packets according to the destination MAC address in the inner Ethernet header of VXLAN packets where the received packets are encapsulated. If the destination MAC address is not the MAC address of the VXLAN Layer 3 gateway, the device directly forwards the packets at line speed.
- Layer 3 non-loopback mode: The device determines whether to loop back the received packets according to the destination MAC address in the inner Ethernet header of VXLAN packets where the received packets are encapsulated. If the destination MAC address is the MAC address of the VXLAN Layer 3 gateway, the device directly forwards the packets at line speed.

NOTE

- GE interface cards do not support the Layer 2 non-loopback mode. Among non-GE interface cards, EA series cards support only the Layer 2 non-loopback mode, and other series cards support Layer 2 and Layer 3 non-loopback modes.
- Traffic will be interrupted when you change the enhanced mode of a Layer 3 VXLAN gateway. Therefore, you are advised to change the enhanced mode of a Layer 3 VXLAN gateway when there is no or little traffic on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
assign forward nvo3-gateway enhanced { l2 | l3 }
```

The Layer 3 VXLAN gateway is configured to work in non-loopback mode.

By default, the enhanced mode of the Layer 3 VXLAN gateway is not set and the switch works in loopback mode.

NOTE

When you set the enhanced mode of the VXLAN gateway to Layer 3 non-loopback, you also need to run the **assign forward nvo3 service extend enable** command to enable the VXLAN service extension function. In this case, ensure that the switch does not have CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA cards installed, or the CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA cards do not transmit the VXLAN service.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

7.13 Checking the Configurations

After configuring VXLAN in centralized gateway mode for static tunnel establishment, check VXLAN tunnel, VNI, and VBDIF interface information.

Prerequisites

VXLAN in centralized gateway mode has been configured for static tunnel establishment.

Procedure

- Run the **display bridge-domain** [*bd-id* [**brief** | **verbose**]] command to check BD configurations.
- Run the **display interface nve** [*nve-number* | **main**] command to check NVE interface information.
- Run the **display vxlan peer** [**vni** *vni-id*] command to check ingress replication lists of a VNI or all VNIs.
- Run the **display vxlan tunnel** [*tunnel-id*] [**verbose**] command to check VXLAN tunnel information.
- Run the **display vxlan vni** [*vni-id* [**verbose**]] command to check VNI information.
- Run the **display interface vbdif** [*bd-id*] command to check VBDIF interface information and statistics.
- Run the **display dfs-group** *dfs-group-id* **active-active-gateway** command to check information of all-active gateways in a DFS group.
- Run the **display arp broadcast-suppress user bridge-domain** *bd-id* command to check the ARP broadcast suppression table of a BD.
- Run the **display arp** [**network** *network-address* [*network-mask* | *mask-length*]] **static** command to check static ARP entries.
- Run the **display mac-address static bridge-domain** *bd-id* command to check static MAC address entries in a BD.
- Run the **display mac-address limit bridge-domain** *bd-id* command to check MAC address limiting configurations of a BD.

- Run the **display mac-address flapping** command to check the MAC address flapping detection configuration.

----End

8 Configuring VXLAN in Single-Node, Centralized Gateway, and BGP EVPN Mode

About This Chapter

When VXLAN in centralized gateway mode using BGP EVPN is deployed, traffic across network segments is forwarded through Layer 3 VXLAN gateways to implement centralized traffic management.

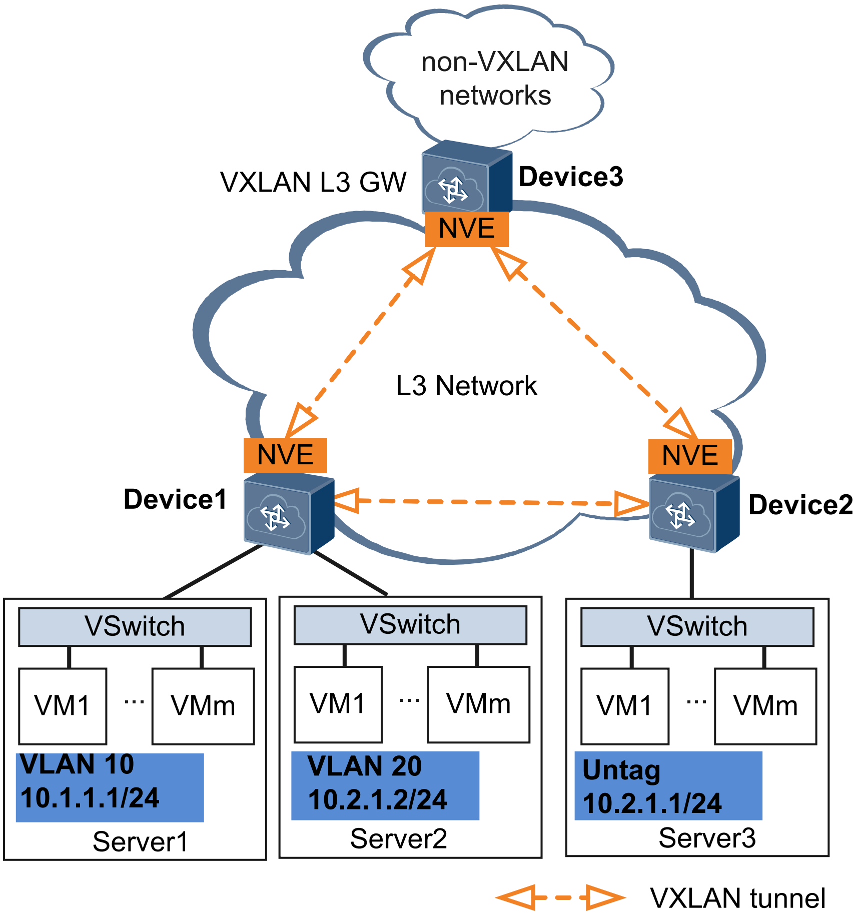
Usage Scenario

An enterprise has allocated VMs in different locations to a tenant. Some of the VMs reside on the same network segment, and the others reside on different network segments. To allow communication between VMs, deploy Layer 2 and Layer 3 VXLAN gateways and establish VXLAN tunnels.

On the network shown in **Figure 8-1**, Server 2 and Server 3 belong to the same network segment and access the VXLAN through Device 1 and Device 2, respectively; Server 1 and Server 2 belong to different network segments and both access the VXLAN through Device 1.

- To allow VM 1 on Server 2 and VM 1 on Server 3 to communicate, deploy Layer 2 VXLAN gateways on Device 1 and Device 2 and establish a VXLAN tunnel between Device 1 and Device 2 so that tenants on the same network segment can communicate.
- To allow VM 1 on Server 1 and VM 1 on Server 3 to communicate, deploy a Layer 3 VXLAN gateway on Device 3 and establish a VXLAN tunnel between Device 1 and Device 3 and between Device 2 and Device 3 so that tenants on different network segments can communicate.

Figure 8-1 VXLAN in centralized gateway mode



Pre-configuration Tasks

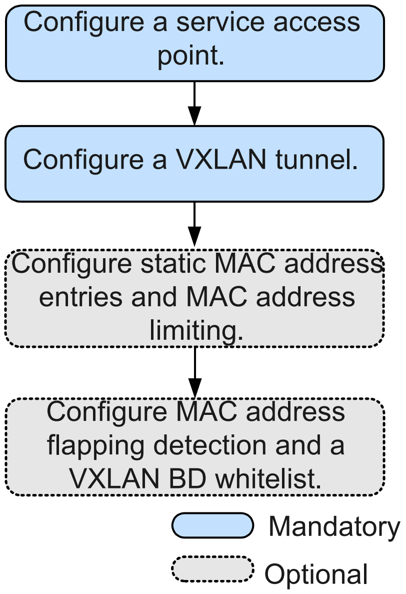
Before configuring VXLAN in centralized gateway mode for static tunnel establishment, ensure that the network is reachable at Layer 3.

Configuration Procedures

NOTE

In a dual-active VXLAN access scenario, two access devices to which a host is dual homed are simulated as a VTEP to prevent loops or MAC address flapping. In this case, ensure that **VXLAN access point** and **VXLAN tunnel configuration** on the two devices are the same.

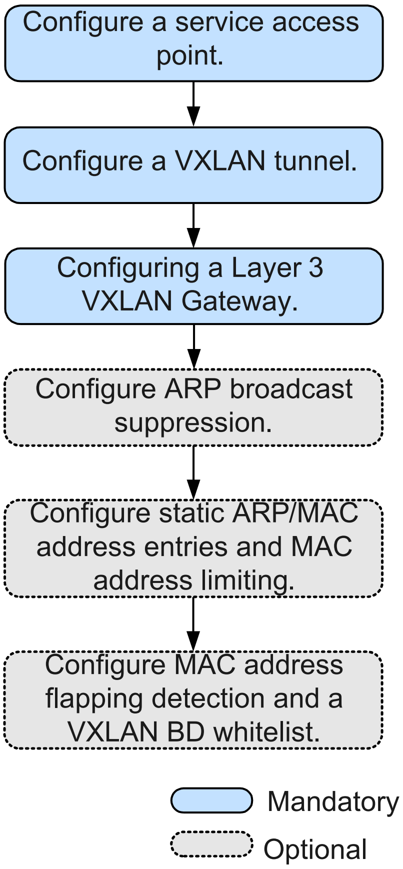
Figure 8-2 Flowchart for configuring intra-segment communication through centralized VXLAN gateways


Configure a service access point.
Configure a VXLAN tunnel.
Configure static MAC address entries and MAC address limiting.
Configure MAC address flapping detection and a VXLAN BD whitelist.
Mandatory
Optional

NOTE

To implement intra-segment communication through centralized VXLAN gateways, configure only static MAC address entries and MAC address limiting described in **7.8 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting**.

Figure 8-3 Flowchart for configuring inter-segment communication through centralized VXLAN gateways


Configure a service access point.
Configure a VXLAN tunnel.
Configuring a Layer 3 VXLAN Gateway.
Configure ARP broadcast suppression.
Configure static ARP/MAC address entries and MAC address limiting.
Configure MAC address flapping detection and a VXLAN BD whitelist.
Mandatory
Optional

8.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function

8.2 Configuring a Service Access Point
A VXLAN service access point can be a Layer 2 sub-interface or VLAN.

8.3 Configuring a VXLAN Tunnel
To allow VXLAN tunnel establishment using EVPN, configure EVPN as the VXLAN control plane, establish a BGP EVPN peer relationship, configure an EVPN instance, and configure ingress replication.

8.4 Configuring a Layer 3 VXLAN Gateway
To allow users on different network segments to communicate, a Layer 3 VXLAN gateway must be deployed, and the default gateway address of the users must be the IP address of the VBDIF interface of the Layer 3 gateway.

8.5 (Optional) Configuring Centralized All-Active Gateways for the VXLAN Network

8.6 (Optional) Configuring ARP Broadcast Suppression
When tenants communicate with each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks and may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on Layer 2 VXLAN gateways.

8.7 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages

8.8 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting
Static ARP entries or MAC address entries can be configured for traffic forwarding, and MAC address limiting can be configured to improve VXLAN security.

8.9 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection
In specific VXLAN applications, when a device connects to a load balancing server equipped with two network interface cards, the server's MAC address may be learned by two interfaces on the device. This is a normal situation where MAC address flapping detection is not needed. In this case, configure a VXLAN BD whitelist for MAC address flapping detection.

8.10 (Optional) Optimizing ACL Resource Usage on the VXLAN

8.11 (Optional) Optimizing Load Balancing on the VXLAN Network

8.12 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode

8.13 Checking the Configurations
After configuring VXLAN in centralized gateway mode for dynamic tunnel establishment, check VXLAN tunnel, VNI, and VBDIF interface information.

8.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function

Context

- Configuring a tunnel mode: You need to set the tunnel mode to VXLAN when configuring the VXLAN feature; otherwise, the configurations do not take effect.
- Enabling the VXLAN ACL extension function: By default, the VXLAN ACL extension function is disabled on the device. If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, and BD traffic statistics collection, on the device deployed with VXLAN services, there is high probability that the other services fail to be configured. You can enable the VXLAN ACL extension function to lower the configuration failure probability.

To ensure normal forwarding of VXLAN packets, the VXLAN tunnel mode must have been configured and the VXLAN ACL extension function must have been enabled on Layer 2 and Layer 3 VXLAN gateways.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip tunnel mode vxlan
```

The tunnel mode is set to VXLAN.

By default, the tunnel mode is VXLAN.

NOTE

Only the Admin-VS in port mode supports this command, and all VSs in group mode support this command.

Step 3 Run:

```
assign forward nvo3 acl extend enable
```

The VXLAN ACL extension function is enabled.

By default, the VXLAN ACL extension function is disabled.

NOTE

You can run this command on the admin-VS only. The configuration takes effect for all the VSs.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Follow-up Procedure

After configuring the VXLAN tunnel mode and enabling the VXLAN ACL extension function, you need to save the configuration and restart the switch to make the configuration take effect.

8.2 Configuring a Service Access Point

A VXLAN service access point can be a Layer 2 sub-interface or VLAN.

Context

When a Layer 2 sub-interface is used as a service access point, different encapsulation types can be configured for the sub-interface to transmit various types of data packets. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD. **Table 8-1** describes the different encapsulation types.

Table 8-1 Traffic encapsulation types

Traffic Encapsulation Type	Description
dot1q	If a Dot1q sub-interface receives a single-tagged VLAN packet, the sub-interface forwards only the packet with a specifie VLAN ID. If a Dot1q sub-interface receives a double-tagged VLAN packet, the sub-interface forwards only the packet with a specified outer VLAN ID. ● When performing VXLAN encapsulation on packets, a Dot1q Layer 2 sub-interface removes the outer tags of the packets. ● When performing VXLAN decapsulation on packets, a Dot1q Layer 2 sub-interface replaces the VLAN tags with specified VLAN tags if the inner packets carry VLAN tags, or adds specified VLAN tags to the packets if the inner packets do not carry VXLAN tags. When setting the encapsulation type to **dot1q** for a Layer 2 sub-interface, note the following: ● The VLAN IDs specified for the Layer 2 sub-interface cannot be the same as either the VLAN IDs of packets allowed to pass through the corresponding Layer 2 interfaces or the MUX VLAN IDs. ● Layer 2 and Layer 3 sub-interfaces cannot have the same VLAN IDs specified.

Traffic Encapsulation Type	Description
untag	An untagged Layer 2 sub-interface receives only packets that do not carry VLAN tags. ● When performing VXLAN encapsulation on packets, an untagged Layer 2 sub-interface does not add any VLAN tag to the packets. ● When performing VXLAN decapsulation on packets, an untagged Layer 2 sub-interface removes the VLAN tags of single-tagged inner packets or the outer VLAN tags of double-tagged inner packets. When setting the encapsulation type to **untag** for a Layer 2 sub-interface, note the following: ● Ensure that the corresponding physical interface of the sub-interface does not have any configuration, and is removed from the default VLAN. ● Untagged Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● An interface can have only one untagged Layer 2 sub-interface configured.
qinq	A QinQ sub-interface receives only tagged packets with specified inner and outer VLAN tags. ● When performing VXLAN encapsulation on packets, a QinQ sub-interface removes two VLAN tags from packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintains the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. ● When performing VXLAN decapsulation on packets, a QinQ sub-interface adds two specific VLAN tags to packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintain the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. **NOTE** The traffic behavior for QinQ interfaces bound to the same BD must be the same. QinQ interfaces do not support DHCP Snooping or VBDIF and cannot be bound to the same BD as Dot1q sub-interfaces. A QinQ interface can have only one outer VLAN tag and one inner VLAN tag.

Traffic Encapsulation Type	Description
default	A default Layer 2 sub-interface receives all packets, irrespective of whether the packets carry VLAN tags. When performing VXLAN encapsulation and decapsulation on packets, a default Layer 2 sub-interface does not process VLAN tags of the packets. When setting the encapsulation type to **default** for a Layer 2 sub-interface, note the following: ● Ensure that the interface for the Layer 2 sub-interface is not added to any VLAN. ● Default Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● If a default Layer 2 sub-interface is created for an interface, the interface cannot have other types of Layer 2 sub-interfaces configured.

NOTE

When a sub-interface that is configured with dot1q and QinQ receives double-tagged VLAN packets, the QinQ sub-interface preferentially processes the packets. For example, if a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of 20, the QinQ sub-interface preferentially processes the packet. If a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of non-20, the dot1q sub-interface preferentially processes the packet.

If a VLAN is used as a service access point, it can be bound to a BD for data packets in the VLAN to be transmitted through this BD.

Configure a service access point on a Layer 2 gateway.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bridge-domain bd-id
```

A BD is created, and the BD view is displayed.

By default, no BD is created.

Step 3 (Optional) Run:

```
description description
```

A description is configured for the BD.

By default, no description is configured for a BD.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Configure a service access point.

- Configure a VLAN as a service access point.

  NOTE

  – In a distributed gateway scenario, a VLAN cannot be configured as a VXLAN service access point.

  – You cannot bind a VLAN to a BD while configuring the ARP broadcast suppression function. After a VLAN is configured as a VXLAN service access point, do not configure ARP broadcast suppression.

  – After a VLAN is bound to a BD, the BD becomes the broadcast domain. Therefore, other service configurations such as DHCP Snooping and IGMP Snooping in the VLAN become invalid.

  – VLAN and BD use 1:1 mapping. That is, a VLAN can be bound to only one BD, and only one VLAN can be bound to a BD.

  a. Run:

  ```
  bridge-domain bd-id
  ```

  The view of an existing BD is displayed.

  b. Run:

  ```
  l2 binding vlan vlan-id
  ```

  A global VLAN is bound to the BD.

  By default, VLANs are not bound to any BD.

  NOTE

  Before performing this step, ensure that a global VLAN has been created. After binding the global VLAN to the BD, add the related device interfaces to the VLAN.

  c. Run:

  ```
  commit
  ```

  The configuration is committed.

- Configure a Layer 2 sub-interface as a service access point.

  a. Run:

  ```
  interface interface-type interface-number.subnum mode l2
  ```

  A Layer 2 sub-interface is created, and the sub-interface view is displayed.

  By default, no Layer 2 sub-interface is created.

  NOTE

  Before running this command, ensure that the Layer 2 interface for which a Layer 2 sub-interface is created does not have the **port link-type dot1q-tunnel** command configuration. If this configuration exists, run the **undo port link-type** command to delete the configuration.

  b. Run:

  ```
  encapsulation { dot1q [ vid ce-vid ] | default | untag  | qinq [ vid pe-
  vid ce-vid ce-vid ] }
  ```

  An encapsulation type is configured for the Layer 2 sub-interface.

  By default, no encapsulation type is configured for Layer 2 sub-interfaces.

  c. (Optional) Run:

  ```
  rewrite pop double
  ```

The sub-interface is enabled to remove double VLAN tags from received packets if the encapsulation type of the sub-interface is set to QinQ in **Step 5.b**.

By default, a Layer 2 sub-interface with the encapsulation type being QinQ is enabled to transparently transmit received packets.

d. Run:

```
bridge-domain bd-id
```

The Layer 2 sub-interface is added to a BD so that the sub-interface can transmit data packets through this BD.

By default, Layer 2 sub-interfaces are not added to any BD.

NOTE

After a Layer 2 sub-interface with the flow encapsulation type set to **default** is added to a BD, you cannot create a VBDIF interface for the BD.

e. Run:

```
commit
```

The configuration is committed.

----End

8.3 Configuring a VXLAN Tunnel

To allow VXLAN tunnel establishment using EVPN, configure EVPN as the VXLAN control plane, establish a BGP EVPN peer relationship, configure an EVPN instance, and configure ingress replication.

Context

In centralized VXLAN gateway scenarios, perform the following steps on the Layer 2 and Layer 3 VXLAN gateways to use EVPN for establishing VXLAN tunnels:

1. Configure EVPN as the VXLAN control plane. Subsequent EVPN configurations can then be performed.
2. Configure a BGP EVPN peer relationship. Configure VXLAN gateways to establish BGP EVPN peer relationships so that they can exchange EVPN routes. If an RR has been deployed, each VXLAN gateway only needs to establish a BGP EVPN peer relationship with the RR.
3. (Optional) Configure an RR. The deployment of RRs reduces the number of BGP EVPN peer relationships to be established, simplifying configuration. A live-network device can be used as an RR, or a standalone RR can be deployed. Layer 3 VXLAN gateways are generally used as RRs, and Layer 2 VXLAN gateways as RR clients.
4. Configure an EVPN instance. EVPN instances are used to receive and advertise EVPN routes.
5. Configure ingress replication. After ingress replication is configured for a VNI, the system uses BGP EVPN to construct a list of remote VTEPs. After a VXLAN gateway receives BUM packets, its sends a copy of the BUM packets to every VXLAN gateway in the list.

Procedure

Step 1 Configure EVPN as the VXLAN control plane.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   evpn-overlay enable
   ```

   EVPN is configured as the VXLAN control plane.

   By default, EVPN is not configured as the VXLAN control plane.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Configure a BGP EVPN peer relationship.

1. Run:

   ```
   bgp as-number [ instance instance-name ]
   ```

   BGP is enabled, and the BGP or BGP multi-instance view is displayed.

   By default, the BGP is disabled. If an RR has been deployed, each VXLAN gateway only needs to establish a BGP EVPN peer relationship with the RR.

2. (Optional) Run:

   ```
   router-id ipv4-address
   ```

   A router ID is set.

   By default, no BGP Router ID is configured, and the Router ID configured for the route management module through the **router id** command is used.

3. Run:

   ```
   peer ipv4-address as-number as-number
   ```

   The peer device is configured as a BGP peer.

   By default, no BGP peer is configured, and no AS number is specified for a peer or peer group.

4. (Optional) Run:

   ```
   peer ipv4-address connect-interface interface-type interface-number [ ipv4-
   source-address ]
   ```

   A source interface and a source address are specified to set up a TCP connection with the BGP peer.

   By default, the outbound interface of a BGP packet serves as the source interface of a BGP packet.

   NOTE

   When loopback interfaces are used to establish a BGP connection, running the **peer connect-interface** command on both ends is recommended to ensure the connectivity. If this command is run on only one end, the BGP connection may fail to be established.

5. (Optional) Run:

   ```
   peer ipv4-address ebgp-max-hop [ hop-count ]
   ```

   The maximum number of hops is set for an EBGP EVPN connection.

   The default value of *hop-count* is 255.

In most cases, a directly connected physical link must be available between EBGP EVPN peers. If you want to establish EBGP EVPN peer relationships between indirectly connected peers, run the **peer ebgp-max-hop** command. The command also can configure the maximum number of hops for an EBGP EVPN connection.

NOTE

When the IP address of loopback interface to establish an EBGP EVPN peer relationship, run the **peer ebgp-max-hop** (of which the value of hop-count is not less than 2) command. Otherwise, the peer relationship fails to be established.

6. Run:

   ```
   l2vpn-family evpn
   ```

   The BGP-EVPN address family view or BGP multi-instance EVPN view is displayed.

   By default, the BGP-EVPN address family or BGP multi-instance EVPN view is disabled.

7. Run:

   ```
   peer { ipv4-address | group-name } enable
   ```

   The device is enabled to exchange EVPN routes with a specified peer or peer group.

   By default, only the peer in the BGP IPv4 unicast address family view is automatically enabled.

8. (Optional) Run:

   ```
   peer { group-name | ipv4-address } route-policy route-policy-name { import |
   export }
   ```

   A routing policy is specified for routes received from or to be advertised to a BGP EVPN peer or peer group.

   After the routing policy is applied, the routes received from or to be advertised to a specified BGP EVPN peer or peer group will be filtered, ensuring that only desired routes are imported or advertised. This configuration helps manage routes and reduce required routing entries and system resources.

9. (Optional) Run:

   ```
   peer { group-name | ipv4-address } mac-limit number [ percentage ] [ alert-
   only | idle-forever | idle-timeout times ]
   ```

   The maximum number of MAC advertisement routes that can be received from each peer is configured.

   If an EVPN instance may import many invalid MAC advertisement routes from peers and these routes occupy a large proportion of the total MAC advertisement routes. If the received MAC advertisement routes exceed the specified maximum number, the system displays an alarm, instructing users to check the validity of the MAC advertisement routes received in the EVPN instance.

10. Run:

    ```
    quit
    ```

    Exit from the BGP-EVPN address family view or BGP multi-instance EVPN view.

11. Run:

    ```
    quit
    ```

    Exit from the BGP or BGP multi-instance view.

12. Run:

```
commit
```

The configuration is committed.

Step 3 (Optional) Configure a Layer 3 VXLAN gateway as an RR. If an RR is configured, each VXLAN gateway only needs to **establish a BGP EVPN peer relationship** with the RR, reducing the number of BGP EVPN peer relationships to be established and simplifying configuration.

1. Run:

   ```
   bgp as-number [ instance instance-name ]
   ```

   The BGP or BGP multi-instance view is displayed.

2. Run:

   ```
   l2vpn-family evpn
   ```

   The BGP-EVPN address family view or BGP multi-instance EVPN view is displayed.

3. Run:

   ```
   peer { ipv4-address | group-name } enable
   ```

   The device is enabled to exchange EVPN routes with a specified peer or peer group.

   By default, only the peer in the BGP IPv4 unicast address family view is automatically enabled.

4. (Optional) Run:

   ```
   peer { ipv4-address | group-name } next-hop-invariable
   ```

   The device is prevented from changing the next hop address of a route when advertising the route to an EBGP peer.

   By default, a BGP EVPN speaker changes the next hops of routes to the interface that it uses to establish EBGP EVPN peer relationships before advertising these routes to EBGP EVPN peers.

5. Run:

   ```
   peer { ipv4-address | group-name } reflect-client
   ```

   The device is configured as an RR and an RR client is specified.

   By default, the route reflector and its client are not configured.

6. Run:

   ```
   undo policy vpn-target
   ```

   The function to filter received EVPN routes based on VPN targets is disabled. If you do not perform this step, the RR will fail to receive and reflect the routes sent by clients.

7. Run:

   ```
   quit
   ```

   Exit from the BGP-EVPN address family view or BGP multi-instance EVPN view.

8. Run:

   ```
   quit
   ```

   Exit from the BGP or BGP multi-instance view.

9. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 4 Configure an EVPN instance.

1. Run:

   ```
   bridge-domain bd-id
   ```

   The BD view is displayed.

   By default, no bridge domain is created.

2. Run:

   ```
   vxlan vni vni-id
   ```

   A VNI is created and mapped to the BD.

   By default, no VNI is created.

3. Run:

   ```
   evpn
   ```

   An EVPN instance is created.

   By default, no EVPN instance is created for VXLANs.

4. Run:

   ```
   route-distinguisher { route-distinguisher | auto }
   ```

   An RD is configured for the EVPN instance. The two ends of a VXLAN tunnel can share an RD or use different RDs.

   By default, no RD is configured for BD EVPN instances.

5. Run:

   ```
   vpn-target { vpn-target &<1-8> | auto } [ both | export-extcommunity | import-
   extcommunity ]
   ```

   VPN targets are configured for the EVPN instance. The export VPN target of the local end must be the same as the import VPN target of the remote end, and the import VPN target of the local end must be the same as the export VPN target of the remote end.

   By default, no VPN target is configured for BD EVPN instances.

6. (Optional) Run:

   ```
   import route-policy policy-name
   ```

   The current EVPN instance is associated with an import routing policy.

   By default, an EVPN instance matches the export VPN targets of received routes against its import VPN targets to determine whether to import these routes. To control route import more precisely, perform this step to associate the EVPN instance with an import routing policy and set attributes for eligible routes.

7. (Optional) Run:

   ```
   export route-policy policy-name
   ```

   The current EVPN instance is associated with an export routing policy.

   By default, an EVPN instance adds all VPN targets in the export VPN target list to EVPN routes to be advertised to its peers. To control route export more precisely, perform this step to associate the EVPN instance with an export routing policy and set attributes for eligible routes.

8. Run:

   ```
   quit
   ```

   The EVPN instance view is exited.

9. Run:

   ```
   quit
   ```

   Return to the system view.

10. Run:

    ```
    commit
    ```

    The configuration is committed.

Step 5 Configure an ingress replication list.

1. Run:

   ```
   interface nve nve-number
   ```

   An NVE interface is created, and the NVE interface view is displayed.

2. Run:

   ```
   source ip-address
   ```

   An IP address is configured for the source VTEP.

   By default, no IP address is configured for any source VTEP.

3. Run:

   ```
   vni vni-id head-end peer-list protocol bgp
   ```

   An ingress replication list is configured.

   By default, no ingress replication list is configured for any VNI.

   NOTE

   BUM packet forwarding is implemented only using ingress replication. To establish a VXLAN tunnel between a Huawei device and a non-Huawei device, ensure that the non-Huawei device also has ingress replication configured. Otherwise, communication fails.

4. Run:

   ```
   quit
   ```

   Return to the system view.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

8.4 Configuring a Layer 3 VXLAN Gateway

To allow users on different network segments to communicate, a Layer 3 VXLAN gateway must be deployed, and the default gateway address of the users must be the IP address of the VBDIF interface of the Layer 3 gateway.

Context

A tenant is identified by a VNI. VNIs can be mapped to BDs in 1:1 mode so that a BD can function as a VXLAN network entity to transmit VXLAN data packets. A VBDIF interface is a Layer 3 logical interface created for a BD. After an IP address is configured for a VBDIF interface of a BD, the VBDIF interface can function as the gateway for tenants in the BD for Layer 3 forwarding. VBDIF interfaces allow Layer 3 communication between VXLANs on different network segments and between VXLANs and non-VXLANs, and implement Layer 2 network access to a Layer 3 network.

VBDIF interfaces are configured on Layer 3 VXLAN gateways for inter-segment communication, and are not needed in the case of intra-segment communication.

NOTE

The DHCP relay function can be configured on the VBDIF interface so that hosts can request IP addresses from the external DHCP server.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface vbdif bd-id
```

A VBDIF interface is created, and the VBDIF interface view is displayed.

Step 3 Run:

```
ip address ip-address { mask | mask-length } [ sub ]
```

An IP address is configured for the VBDIF interface to implement Layer 3 interworking.

By default, no IP address is configured for interfaces.

Step 4 (Optional) Run:

```
mac-address mac-address
```

A MAC address is configured for the VBDIF interface.

By default, the MAC address of a VBDIF interface is the system MAC address.

Step 5 Run:

```
commit
```

The configuration is committed.

----End

8.5 (Optional) Configuring Centralized All-Active Gateways for the VXLAN Network

Context

On a traditional network, Virtual Router Redundancy Protocol (VRRP) is used to protect gateways. One gateway is in the active state and the others in the standby state, leading to low gateway utilization. When a gateway fails, the new active gateway needs to be reelected, and the convergence performance upon a gateway fault is low.

After you configure all-active VXLAN gateways, the multiple VXLAN Layer 3 gateways can be virtualized into one VXLAN gateway and traffic is forwarded through any gateway. The all-active VXLAN gateway function improves gateway utilization and convergence performance.

Perform the following operations on the Layer 3 VXLAN gateway.

NOTE

- In the VXLAN centralized all-active gateways networks, if the uplink of spine fails, user-side traffic may fail to be forwarded and therefore are discarded. To prevent this problem, associate uplink and downlink interfaces with the Monitor Link group. When the uplink interface becomes Down, the downlink interface also becomes Down. This prevent user-side traffic from being discarded. For details about the monitor-link configuration, see Configuring the Uplink and Downlink Interfaces in a Monitor Link Group.
- After deploying VXLAN centralized all-active gateways, you may need to reset the device after a device upgrade or patch installation. Pay attention to the following points:
  - Before the reset: On the access side, tear down the equal-cost multi-path routing (ECMP) paths between the access device and the VTEP address of the Spine device to decrease the priority of the advertised VTEP host route. This prevents the access device from forwarding traffic to the spine device. On the network side, decrease the priority of the VXLAN gateway route advertised by the device to prevent the upstream device from forwarding traffic to the device.
  - After the reset: Synchronize ARP entries and then restore the original priorities of the VTEP host route and the VXLAN gateway route, to avoid the traffic loss caused by insufficient ARP entries. For example, it takes approximately 20 minutes to synchronize 128K ARP entries.
- After you shut down or delete the VBDIF interface on the all-active gateway, the traffic cannot be switched to another gateway. The access device forwards traffic to this device based on the ECMP route of the VTEP, but this device cannot forward traffic to another all-active gateway, causing traffic loss.
- To prevent traffic forwarding before ARP entry synchronization when the device is reset, it is recommended that you configure the delayed route advertisement mechanism. Take OSPF as an example. Run the command **stub-router on-startup** [*interval*] **include-stub** in the OSPF process related to the VTEP and VBDIF routes to configure the stub server interval for the device when the device restarts or fails (for example, 20 minutes for 128K ARP entries).

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Configure the Layer 3 VXLAN gateway function on switches that work as all-active gateways. For details, see **8.3 Configuring a VXLAN Tunnel** and **8.4 Configuring a Layer 3 VXLAN Gateway**. The following table lists the commands.

Procedure	Command	Description
Assign an ingress replication list for a VNI.	**vni** *vni-id* **head-end peer-list** *ip-address* &<1-10>	Ensure that the gateways have the same ingress replication for a VNI. NOTE BUM packet forwarding is implemented only using ingress replication. To establish a VXLAN tunnel between a Huawei device and a non-Huawei device, ensure that the non-Huawei device also has ingress replication configured. Otherwise, communication fails.

Procedure	Command	Description
Assign an IP address to each VBDIF interface.	**ip address** *ip-address* { *mask* | *mask-length* } [**sub**]	Ensure that the VBDIF interfaces of the gateways have the same IP address.
Assign a MAC address to each VBDIF interface.	**mac-address** *mac-address*	Ensure that the VBDIF interfaces of the gateways have the same MAC address.

Step 3 If you want to configure the all-active gateway function on the device, configure a DFS group to synchronize ARP entries between all-active gateways.

1. Run:

   ```
   dfs-group dfs-group-id
   ```

   A DFS group is created and its view is displayed.

   By default, no DFS group is created.

2. Run:

   ```
   source ip ip-address
   ```

   The DFS group is bound to an IPv4 address.

   By default, a DFS group is not bound to any IPv4 address.

3. (Optional) Run:

   ```
   udp port port-number
   ```

   The UDP port number of the DFS group is set.

   By default, the UDP port number of the DFS group is 61467.

4. Run:

   ```
   active-active-gateway
   ```

   An all-active gateways is created and its view is displayed.

   By default, no all-active gateway is created.

5. Run:

   ```
   peer ip-address [ vpn-instance vpn-instance-name ]
   ```

   The all-active gateway peer is configured.

   By default, no all-active gateway peer is configured.

   NOTE

   You can specify a maximum of fifteen IP addresses for all-active gateway peers in the all-active gateway view of a DFS group.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

8.6 (Optional) Configuring ARP Broadcast Suppression

When tenants communicate with each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks and may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on Layer 2 VXLAN gateways.

Context

After you enable ARP broadcast suppression on a Layer 2 VXLAN gateway, configure Border Gateway Protocol Ethernet Virtual Private Network (BGP EVPN) on Layer 2 and Layer 3 VXLAN gateways to allow ARP broadcast suppression to take effect. BGP EVPN can then generate host information based on learned ARP entries and advertise the host information to Layer 2 VXLAN gateways. After the Layer 2 VXLAN gateways receive ARP broadcast packets, they convert the ARP broadcast packets into unicast packets based on the learned host information before forwarding the packets out. This decreases the number of broadcast packets in a BD, improving network performance.

Configuring BGP RRs is recommended to simplify BGP configuration. Layer 3 VXLAN gateways are generally used as RRs, and Layer 2 VXLAN gateways as RR clients.

Procedure

Step 1 Configure BGP EVPN on Layer 2 and Layer 3 VXLAN gateways to advertise host information learned by Layer 2 VXLAN gateways.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   bgp as-number [ instance instance-name ]
   ```

   The BGP or BGP multi-instance view is displayed.

3. Run:

   ```
   l2vpn-family evpn
   ```

   The BGP-EVPN address family view or BGP multi-instance EVPN view is displayed.

   By default, the BGP-EVPN address family or BGP multi-instance EVPN view is disabled.

4. Configure advertisement of ARP or IRB routes to implement ARP broadcast suppression. The following two configurations cannot coexist.

   – To configure ARP route advertisement, run the **peer** { *ipv4-address* | *group-name* } **advertise arp** command.

   – To configure IRB route advertisement, run the **peer** { *ipv4-address* | *group-name* } **advertise irb** command.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Enable BGP EVPN on a Layer 3 VXLAN gateway to collect host information.

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
interface vbdif bd-id
```

The VBDIF interface view is displayed.

3. Run:

```
arp collect host enable
```

BGP EVPN is enabled to collect host information.

By default, BGP EVPN is disabled from collecting host information.

4. Run:

```
commit
```

The configuration is committed.

Step 3 Enable ARP broadcast suppression on a Layer 2 VXLAN gateway.

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
bridge-domain bd-id
```

The BD view is displayed.

3. Run:

```
arp broadcast-suppress enable
```

ARP broadcast suppression is enabled.

By default, ARP broadcast suppression is disabled.

4. Run:

```
commit
```

The configuration is committed.

----End

8.7 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages

Context

When the device wants to communicate with another device in the same network segment, it queries ARP entries to direct packet forwarding. If the device fails to find the corresponding ARP entry from the forwarding plane, it sends an ARP Miss message to the CPU. The ARP Miss message will trigger the device to send an ARP broadcast packet to start ARP learning. In some cases, customers may want to limit the number of broadcast packets on the VXLAN network. You can then disable a VBDIF interface from sending ARP Miss messages to achieve this purpose.

After a VBDIF interface is disabled from sending ARP Miss messages, the device cannot learn ARP entries from this VBDIF interface, so ARP entries must be manually configured on it.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface vbdif bd-id
```

A VBDIF interface is created and the VBDIF interface view is displayed.

Step 3 Run:

```
arp miss disable
```

The VBDIF interface is disabled from sending ARP Miss messages.

By default, a VBDIF interface can send ARP Miss messages.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

8.8 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting

Static ARP entries or MAC address entries can be configured for traffic forwarding, and MAC address limiting can be configured to improve VXLAN security.

Context

- Static ARP entries are manually configured and maintained. They can be neither aged nor overwritten by dynamic ARP entries. Therefore, configuring static ARP entries on Layer 3 VXLAN gateways enhances communication security. If a static ARP entry is configured on a device, the device can communicate with a peer device that has a specified IP address using only the specified MAC address. Network attackers cannot modify the mapping between the IP and MAC addresses, which ensures communication between the two devices.
- After the source NVE on a VXLAN tunnel receives broadcast, unknown unicast, and multicast (BUM) packets, the local VTEP sends a copy of the BUM packets to every VTEP in the ingress replication list. Configuring static MAC address entries helps reduce broadcast traffic and prevent unauthorized data access from bogus users.
- The maximum number of MAC addresses that a device can learn can be configured to limit the number of access users and prevent against attacks on MAC address tables. If the device has learned the maximum number of MAC addresses allowed, no more addresses can be learned. The device can also be configured to discard packets after learning the maximum allowed number of MAC addresses, improving network security.
- If Layer 3 VXLAN gateway does not need to learn MAC addresses of packets in a BD, MAC address learning can be disabled from the BD to conserve MAC address entry resources. If the network topology of a VXLAN becomes stable and MAC address entry learning is complete, MAC address learning can also be disabled.

Configuring static MAC address entries and MAC address limiting applies to Layer 2 VXLAN gateways; configuring static ARP entries applies to Layer 3 VXLAN gateways; disabling MAC address limiting applies to both Layer 2 and Layer 3 VXLAN gateways.

Procedure

- Configure a static ARP entry.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  arp static ip-address mac-address vni vni-id source-ip source-ip peer-ip
  peer-ip
  ```

  A static ARP entry is configured.

  By default, no static ARP entry is configured.

  NOTE

  ip-address must belong to the same network segment as the Layer 3 gateway's IP address.

  c. Run:

  ```
  commit
  ```

  The configuration is committed.

- Configure a static MAC address entry.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  mac-address static mac-address bridge-domain bd-id source source-ip-
  address peer peer-ip vni vni-id
  ```

  A static MAC address entry is configured.

  By default, no static MAC address entry is configured.

  c. Run:

  ```
  commit
  ```

  The configuration is committed.

- Configure MAC address limiting.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  bridge-domain bd-id
  ```

  The BD view is displayed.

  c. Run:

  ```
  mac-address limit { action { discard | forward } | maximum max | alarm
  { disable | enable } } *
  ```

  MAC address limiting is configured.

  By default, MAC address limiting is not configured.

d. Run:

```
commit
```

The configuration is committed.

- Disable MAC address learning.

a. Run:

```
system-view
```

The system view is displayed.

b. Run:

```
bridge-domain bd-id
```

The BD view is displayed.

c. Run:

```
mac-address learning disable
```

MAC address learning is disabled.

By default, MAC address learning is enabled for a BD.

d. Run:

```
commit
```

The configuration is committed.

----End

8.9 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection

In specific VXLAN applications, when a device connects to a load balancing server equipped with two network interface cards, the server's MAC address may be learned by two interfaces on the device. This is a normal situation where MAC address flapping detection is not needed. In this case, configure a VXLAN BD whitelist for MAC address flapping detection.

Context

By default, MAC address flapping detection is enabled globally. After a BD is added to a MAC address flapping detection whitelist, detection is not performed for this BD. Even if MAC address flapping occurs in the BD, the occurrence generates neither an alarm nor a record.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
mac-address flapping detection exclude bridge-domain bd-id1 [ to bd-id2 ]
```

A VXLAN BD whitelist for MAC address flapping detection is configured.

By default, no VXLAN BD whitelist for MAC address flapping detection is configured.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

8.10 (Optional) Optimizing ACL Resource Usage on the VXLAN

Context

If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, BD traffic statistics collection, and DHCP on a switch deployed with VXLAN services, there is high probability that the other services fail to be configured. In this case, use either of the following methods to reduce the probability of service deployment failures.

- Enabling the VXLAN service extension function:

  This function takes effect only on VXLAN-enabled cards except the CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA series.

  NOTE

  After this function is enabled, only CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA can forward packets with 230 to 294 bytes received from the VXLAN tunnels to cards of the same series.

- Disabling an Eth-Trunk from load balancing VXLAN packets in optimized mode:
  - You are advised to disable load balancing of VXLAN packets in optimized mode when neither the user side nor the network side of a VXLAN tunnel is an Eth-Trunk.
  - You can disable load balancing of VXLAN packets in optimized mode when either the user side or the network side of a VXLAN tunnel is an Eth-Trunk. After this function is disabled, an Eth-Trunk does not evenly load balance VXLAN packets or does not perform load balancing.

  NOTE

  This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.

Procedure

Step 1 Enable the VXLAN service extension function.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 service extend enable
   ```

   The VXLAN service extension function is enabled.

   By default, the VXLAN service extension function is disabled.

3. Run:

```
commit
```

The configuration is committed.

Step 2 Disable an Eth-Trunk from load balancing VXLAN packets in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 eth-trunk hash disable
   ```

   An Eth-Trunk is disabled from load balancing VXLAN packets in optimized mode.

   By default, an Eth-Trunk load balances VXLAN packets in optimized mode.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

8.11 (Optional) Optimizing Load Balancing on the VXLAN Network

Context

On a VXLAN network, VXLAN packets can be load balanced through ECMP or Eth-Trunks. To enable load balancing or improve the load balancing effect, enable either of the following functions:

- Enable load balancing of VXLAN packets through ECMP in optimized mode.
- Enable an Eth-Trunk to load balance VXLAN packets in optimized mode.

NOTE

This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.

Procedure

Step 1 Enable load balancing of VXLAN packets through ECMP in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 ecmp hash enable
   ```

   Load balancing of VXLAN packets through ECMP in optimized mode is enabled.

   By default, load balancing of VXLAN packets through ECMP in optimized mode is disabled.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Enable an Eth-Trunk to load balance VXLAN packets in optimized mode.

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
assign forward nvo3 eth-trunk hash enable
```

An Eth-Trunk is enabled to load balance VXLAN packets in optimized mode.

By default, an Eth-Trunk is disabled from load balancing VXLAN packets in optimized mode.

3. Run:

```
commit
```

The configuration is committed.

----End

8.12 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode

Context

By default, a Layer 3 VXLAN gateway works in loopback mode. That is, the Layer 3 VXLAN gateway first loops back VXLAN-encapsulated packets, and then forwards the packets. In this situation, traffic transmitted over the VXLAN tunnel cannot be forwarded at line speed when the traffic volume is heavy.

To solve the preceding problem, configure the Layer 3 VXLAN gateway to work in non-loopback mode.

- Layer 2 non-loopback mode: The device determines whether to loop back the received packets according to the destination MAC address in the inner Ethernet header of VXLAN packets where the received packets are encapsulated. If the destination MAC address is not the MAC address of the VXLAN Layer 3 gateway, the device directly forwards the packets at line speed.
- Layer 3 non-loopback mode: The device determines whether to loop back the received packets according to the destination MAC address in the inner Ethernet header of VXLAN packets where the received packets are encapsulated. If the destination MAC address is the MAC address of the VXLAN Layer 3 gateway, the device directly forwards the packets at line speed.

NOTE

- GE interface cards do not support the Layer 2 non-loopback mode. Among non-GE interface cards, EA series cards support only the Layer 2 non-loopback mode, and other series cards support Layer 2 and Layer 3 non-loopback modes.
- Traffic will be interrupted when you change the enhanced mode of a Layer 3 VXLAN gateway. Therefore, you are advised to change the enhanced mode of a Layer 3 VXLAN gateway when there is no or little traffic on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
assign forward nvo3-gateway enhanced { l2 | l3 }
```

The Layer 3 VXLAN gateway is configured to work in non-loopback mode.

By default, the enhanced mode of the Layer 3 VXLAN gateway is not set and the switch works in loopback mode.

NOTE

When you set the enhanced mode of the VXLAN gateway to Layer 3 non-loopback, you also need to run the **assign forward nvo3 service extend enable** command to enable the VXLAN service extension function. In this case, ensure that the switch does not have CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA cards installed, or the CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA cards do not transmit the VXLAN service.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

8.13 Checking the Configurations

After configuring VXLAN in centralized gateway mode for dynamic tunnel establishment, check VXLAN tunnel, VNI, and VBDIF interface information.

Prerequisites

VXLAN in centralized gateway mode has been configured for dynamic tunnel establishment.

Procedure

- Run the **display bridge-domain** [*bd-id* [**brief** | **verbose**]] command to check BD configurations.
- Run the **display interface nve** [*nve-number* | **main**] command to check NVE interface information.
- Run the **display evpn vpn-instance** [*vpn-instance-name*] command to check EVPN instance information.
- Run the **display bgp** [**instance** *instance-name*] **evpn peer** [[*ipv4-address*] **verbose**] command to check BGP EVPN peer information.
- Run the **display vxlan peer** [**vni** *vni-id*] command to check ingress replication lists of a VNI or all VNIs.
- Run the **display vxlan tunnel** [*tunnel-id*] [**verbose**] command to check VXLAN tunnel information.

- Run the **display vxlan vni** [*vni-id* [**verbose**]] command to check VNI information.
- Run the **display interface vbdif** [*bd-id*] command to check VBDIF interface information and statistics.
- Run the **display dfs-group** *dfs-group-id* **active-active-gateway** command to check information of all-active gateways in a DFS group.
- Run the **display arp broadcast-suppress user bridge-domain** *bd-id* command to check the ARP broadcast suppression table of a BD.
- Run the **display arp** [**network** *network-address* [*network-mask* | *mask-length*]] **static** command to check static ARP entries.
- Run the **display mac-address static bridge-domain** *bd-id* command to check static MAC address entries in a BD.
- Run the **display mac-address limit bridge-domain** *bd-id* command to check MAC address limiting configurations of a BD.
- Run the **display mac-address flapping** command to check the MAC address flapping detection configuration.
- Run the **display bgp** [**instance** *instance-name*] **evpn all routing-table** command to check EVPN route information.
- Run the **display mac-address inactive** [**evn**] [**slot** *slot-id*] command to check the MAC address entries that fail to be delivered.
- Run the **display mac-address total-number evn** [**vlan** *vlan-id*] command to check the number of EVN MAC address entries.
- Run the **display mac-address evn** [**vlan** *vlan-id*] command to check EVN MAC address entries.

----End

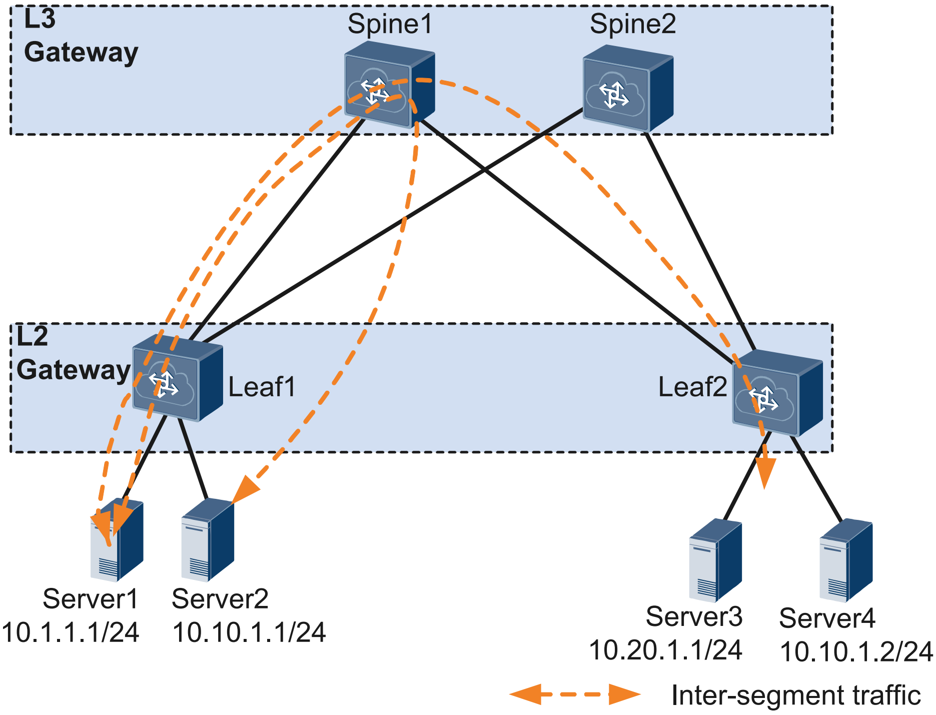
9 Configuring VXLAN in Single-Node, Distributed Gateway, and MP-BGP Mode

About This Chapter

Distributed VXLAN gateways can be configured to address problems that occur in legacy centralized VXLAN gateway networking, for example, forwarding paths are not optimal, and the ARP entry specification is a bottleneck.

Usage Scenario

Figure 9-1 Centralized VXLAN gateway networking

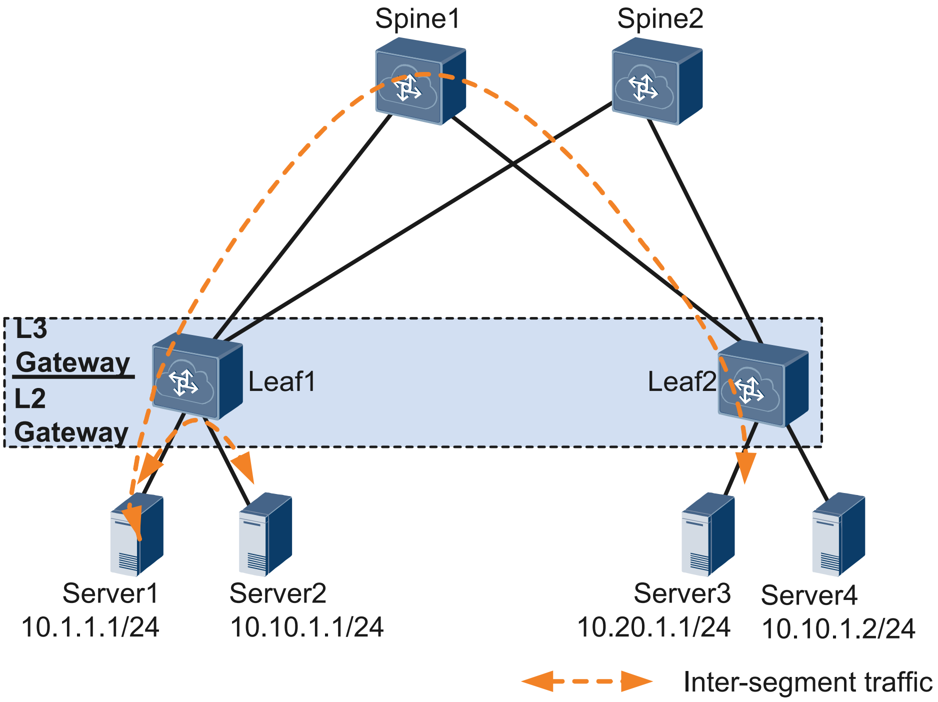


In legacy networking, a centralized Layer 3 gateway is deployed on a spine node. On the network shown in **Figure 9-1**, packets across different networks must be forwarded through a centralized Layer 3 gateway, resulting in the following problems:

- Forwarding paths are not optimal. All Layer 3 traffic must be transmitted to the centralized Layer 3 gateway for forwarding.
- The ARP entry specification is a bottleneck. ARP entries must be generated for tenants on the Layer 3 gateway. However, only a limited number of ARP entries can be configured for the Layer 3 gateway, impeding data center network expansion.

To address these problems, distributed VXLAN gateways can be configured. On the network shown in **Figure 9-2**, Server 1 and Server 2 on different network segments both connect to Leaf 1. When Server 1 and Server 2 communicate, traffic is forwarded only through Leaf 1, not through any spine node.

Figure 9-2 Distributed VXLAN gateway networking



Distributed VXLAN gateway networking has the following characteristics:

- Flexible deployment. A leaf node can function as both Layer 2 and Layer 3 VXLAN gateways.
- Improved network expansion capabilities. A leaf node only needs to learn the ARP entries of servers attached to it. A centralized Layer 3 gateway in the same scenario, however, has to learn the ARP entries of all servers on the network. Therefore, the ARP entry specification is no longer a bottleneck on a distributed VXLAN gateway.
- Enhanced network performance: Changing broadcast packets to unicast packets, leaf nodes can determine whether to broadcast ARP request packets received from tenants or servers. This reduces the number of ARP broadcast packets and improves network performance.

Pre-configuration Tasks

Before configuring VXLAN in distributed gateway mode using MP-BGP, ensure that BGP VPNv4 peer relationships have been established.

Configuration Procedures

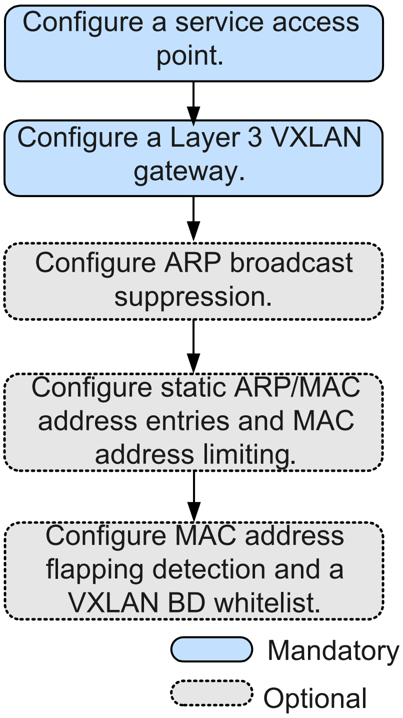
NOTE

The procedure for configuring intra-segment communication through distributed VXLAN gateways is the same as that for configuring intra-segment communication through centralized VXLAN gateways. For details, see **7 Configuring VXLAN in Single-Node, Centralized Gateway, and Static Mode**. This chapter describes the procedure for configuring inter-segment communication through distributed VXLAN gateways.

In a dual-active VXLAN access scenario, two access devices to which a host is dual homed are simulated as a VTEP to prevent loops or MAC address flapping. In this case, ensure that **VXLAN access point** and **VXLAN tunnel configuration** on the two devices are the same.

Figure 9-3 shows the flowchart for configuring inter-segment communication through distributed VXLAN gateways.

Figure 9-3 Flowchart for configuring inter-segment communication through distributed VXLAN gateways


Configure a service access point.
Configure a Layer 3 VXLAN gateway.
Configure ARP broadcast suppression.
Configure static ARP/MAC address entries and MAC address limiting.
Configure MAC address flapping detection and a VXLAN BD whitelist.
Mandatory
Optional

9.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function

9.2 Configuring a VXLAN Service Access Point
On VXLANs, Layer 2 sub-interfaces are used as service access points. These Layer 2 sub-interfaces can have different encapsulation types configured to transmit various types of data packets. A BD is a broadcast domain. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD.

9.3 Configuring a VXLAN Tunnel and a Layer 3 VXLAN Gateway

To isolate tenants at Layer 3, VPN is generally used. In a distributed VXLAN gateway scenario, when Layer 3 communication is implemented through Layer 3 gateways, the Layer 3 gateways must be bound to VPN instances. Then VXLAN tunnels can be established through BGP VPN peers. A Layer 3 VXLAN gateway performs VXLAN encapsulation and decapsulation to allow inter-segment VXLAN communication and access to external networks.

9.4 (Optional) Configuring ARP Broadcast Suppression
When tenants communicate with each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks and may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on VXLAN gateways.

9.5 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages

9.6 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting
Static ARP entries or MAC address entries can be configured for traffic forwarding, and MAC address limiting can be configured to improve VXLAN security.

9.7 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection
In specific VXLAN applications, when a device connects to a load balancing server equipped with two network interface cards, the server's MAC address may be learned by two interfaces on the device. This is a normal situation where MAC address flapping detection is not needed. In this case, configure a VXLAN BD whitelist for MAC address flapping detection.

9.8 (Optional) Optimizing ACL Resource Usage on the VXLAN

9.9 (Optional) Optimizing Load Balancing on the VXLAN Network

9.10 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode

9.11 Checking the Configurations
After configuring VXLAN in distributed gateway mode using MP-BGP, check the configurations, and you can find that VXLAN tunnels are dynamically established and are in the Up state.

9.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function

Context

- Configuring a tunnel mode: You need to set the tunnel mode to VXLAN when configuring the VXLAN feature; otherwise, the configurations do not take effect.
- Enabling the VXLAN ACL extension function: By default, the VXLAN ACL extension function is disabled on the device. If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, and BD traffic statistics collection, on the device deployed with VXLAN services, there is high probability that the other services fail to be configured. You can enable the VXLAN ACL extension function to lower the configuration failure probability.

To ensure normal forwarding of VXLAN packets, the VXLAN tunnel mode must have been configured and the VXLAN ACL extension function must have been enabled on Layer 2 and Layer 3 VXLAN gateways.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip tunnel mode vxlan
```

The tunnel mode is set to VXLAN.

By default, the tunnel mode is VXLAN.

NOTE

Only the Admin-VS in port mode supports this command, and all VSs in group mode support this command.

Step 3 Run:

```
assign forward nvo3 acl extend enable
```

The VXLAN ACL extension function is enabled.

By default, the VXLAN ACL extension function is disabled.

NOTE

You can run this command on the admin-VS only. The configuration takes effect for all the VSs.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Follow-up Procedure

After configuring the VXLAN tunnel mode and enabling the VXLAN ACL extension function, you need to save the configuration and restart the switch to make the configuration take effect.

9.2 Configuring a VXLAN Service Access Point

On VXLANs, Layer 2 sub-interfaces are used as service access points. These Layer 2 sub-interfaces can have different encapsulation types configured to transmit various types of data packets. A BD is a broadcast domain. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD.

Context

When a Layer 2 sub-interface is used as a service access point, different encapsulation types can be configured for the sub-interface to transmit various types of data packets. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD. **Table 9-1** describes the different encapsulation types.

Table 9-1 Traffic encapsulation types

Traffic Encapsulation Type	Description
dot1q	If a Dot1q sub-interface receives a single-tagged VLAN packet, the sub-interface forwards only the packet with a specifie VLAN ID. If a Dot1q sub-interface receives a double-tagged VLAN packet, the sub-interface forwards only the packet with a specified outer VLAN ID. ● When performing VXLAN encapsulation on packets, a Dot1q Layer 2 sub-interface removes the outer tags of the packets. ● When performing VXLAN decapsulation on packets, a Dot1q Layer 2 sub-interface replaces the VLAN tags with specified VLAN tags if the inner packets carry VLAN tags, or adds specified VLAN tags to the packets if the inner packets do not carry VXLAN tags. When setting the encapsulation type to **dot1q** for a Layer 2 sub-interface, note the following: ● The VLAN IDs specified for the Layer 2 sub-interface cannot be the same as either the VLAN IDs of packets allowed to pass through the corresponding Layer 2 interfaces or the MUX VLAN IDs. ● Layer 2 and Layer 3 sub-interfaces cannot have the same VLAN IDs specified.

Traffic Encapsulation Type	Description
untag	An untagged Layer 2 sub-interface receives only packets that do not carry VLAN tags. ● When performing VXLAN encapsulation on packets, an untagged Layer 2 sub-interface does not add any VLAN tag to the packets. ● When performing VXLAN decapsulation on packets, an untagged Layer 2 sub-interface removes the VLAN tags of single-tagged inner packets or the outer VLAN tags of double-tagged inner packets. When setting the encapsulation type to **untag** for a Layer 2 sub-interface, note the following: ● Ensure that the corresponding physical interface of the sub-interface does not have any configuration, and is removed from the default VLAN. ● Untagged Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● An interface can have only one untagged Layer 2 sub-interface configured.
qinq	A QinQ sub-interface receives only tagged packets with specified inner and outer VLAN tags. ● When performing VXLAN encapsulation on packets, a QinQ sub-interface removes two VLAN tags from packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintains the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. ● When performing VXLAN decapsulation on packets, a QinQ sub-interface adds two specific VLAN tags to packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintain the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. **NOTE** The traffic behavior for QinQ interfaces bound to the same BD must be the same. QinQ interfaces do not support DHCP Snooping or VBDIF and cannot be bound to the same BD as Dot1q sub-interfaces. A QinQ interface can have only one outer VLAN tag and one inner VLAN tag.

Traffic Encapsulation Type	Description
default	A default Layer 2 sub-interface receives all packets, irrespective of whether the packets carry VLAN tags. When performing VXLAN encapsulation and decapsulation on packets, a default Layer 2 sub-interface does not process VLAN tags of the packets. When setting the encapsulation type to **default** for a Layer 2 sub-interface, note the following: ● Ensure that the interface for the Layer 2 sub-interface is not added to any VLAN. ● Default Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● If a default Layer 2 sub-interface is created for an interface, the interface cannot have other types of Layer 2 sub-interfaces configured.

NOTE

When a sub-interface that is configured with dot1q and QinQ receives double-tagged VLAN packets, the QinQ sub-interface preferentially processes the packets. For example, if a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of 20, the QinQ sub-interface preferentially processes the packet. If a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of non-20, the dot1q sub-interface preferentially processes the packet.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bridge-domain bd-id
```

A BD is created, and the BD view is displayed.

By default, no BD is created.

Step 3 (Optional) Run:

```
description description
```

A description is configured for the BD.

By default, no description is configured for a BD.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Run:

```
interface interface-type interface-number.subnum mode l2
```

A Layer 2 sub-interface is created, and the sub-interface view is displayed.

By default, no Layer 2 sub-interface is created.

Before running this command, ensure that the Layer 2 interface for which a Layer 2 sub-interface is created does not have the **port link-type dot1q-tunnel** command configuration. If this configuration exists, run the **undo port link-type** command to delete the configuration.

Step 6 Run:

```
encapsulation { dot1q [ vid ce-vid ] | default | untag  | qinq [ vid pe-vid ce-
vid ce-vid ] }
```

An encapsulation type is configured for the Layer 2 sub-interface.

By default, no encapsulation type is configured for Layer 2 sub-interfaces.

Step 7 (Optional) Run:

```
rewrite pop double
```

The sub-interface is enabled to remove double VLAN tags from received packets if the encapsulation type of the sub-interface is set to QinQ in **Step 6**.

By default, a Layer 2 sub-interface with the encapsulation type being QinQ is enabled to transparently transmit received packets.

Step 8 Run:

```
bridge-domain bd-id
```

The Layer 2 sub-interface is added to a BD so that the sub-interface can transmit data packets through this BD.

By default, the Layer 2 sub-interface is not added to a BD.

Step 9 Run:

```
commit
```

The configuration is committed.

----End

9.3 Configuring a VXLAN Tunnel and a Layer 3 VXLAN Gateway

To isolate tenants at Layer 3, VPN is generally used. In a distributed VXLAN gateway scenario, when Layer 3 communication is implemented through Layer 3 gateways, the Layer 3 gateways must be bound to VPN instances. Then VXLAN tunnels can be established through BGP VPN peers. A Layer 3 VXLAN gateway performs VXLAN encapsulation and decapsulation to allow inter-segment VXLAN communication and access to external networks.

Context

A Layer 3 gateway assigns a Layer 2 VNI to each network segment (BD) and a Layer 3 VNI to each tenant identified by a VPN instance. Layer 2 VNIs are used for intra-segment communication. Layer 3 VNIs are used for inter-segment communication. During Layer 3 communication through a Layer 3 gateway, VNI IDs bound to VPN instances are transmitted to the remote Layer 3 gateway through a VXLAN tunnel. The remote Layer 3 gateway identifies VPNs based on tenants' VNI IDs to determine whether the tenants belong to the same VPN.

In a distributed VXLAN gateway scenario, inter-segment communication must be implemented at Layer 3. Therefore, Layer 3 gateways must learn host routes from each other. To implement inter-segment communication, Layer 3 VXLAN gateways must have the functions listed in **Table 9-2** configured.

Table 9-2 Functions required on Layer 3 VXLAN gateways

Function	Description
Host route advertisement	A Layer 3 VXLAN gateway must learn tenants' ARP entries, generate host routes based on the ARP entries, and use BGP to advertise the host routes to BGP peers.
Remote-nexthop attribute advertisement	BGP dynamically manages the VXLAN tunnel between Layer 3 VXLAN gateways and advertises the remote-nexthop attribute carrying the tunnel address, L3VPN VNI, and MAC address to BGP peers.

NOTE

If tenants on the same network segment connect to different Layer 3 VXLAN gateways, the Layer 3 VXLAN gateways must have the same IP address and MAC address configured. When tenants are moved to a different location, the tenants can retain Layer 3 gateway configurations, reducing maintenance workload.

BUM packet forwarding is implemented only using ingress replication. To establish a VXLAN tunnel between a Huawei device and a non-Huawei device, ensure that the non-Huawei device also has ingress replication configured. Otherwise, communication fails.

Procedure

Step 1 Assign a VNI to a tenant.

1. Run:

```
system-view
```

The system view is displayed.

2. Run:

```
bridge-domain bd-id
```

The BD view is displayed.

By default, no BD is created.

3. (Optional) Run:

```
vxlan vni vni-id
```

A VNI is created and bound to the BD.

By default, no VNI is created.

NOTE

The **vxlan vni** *vni-id* command must be run when VXLAN tunnels are used for intra-segment communication.

4. Run:

```
quit
```

Return to the system view.

Step 2 Assign a VNI to a VPN instance.

1. Run:

```
ip vpn-instance vpn-instance-name
```

The VPN instance view is displayed.

2. Run:

```
vxlan vnivni-id
```

A VNI is created and bound to the VPN instance.

By default, no VNI is created.

NOTE

The VNI created in the VPN instance view cannot be the same as that created in the BD view.

3. Run:

```
quit
```

Return to the system view.

Step 3 Configure a source VTEP address.

1. Run:

```
interface nve nve-number
```

An NVE interface is created, and the NVE interface view is displayed.

By default, no NVE interface is created.

2. Run:

```
mode l3
```

The NVE interface is configured to work in Layer 3 mode.

By default, an NVE interface works in Layer 2 mode.

3. Run:

```
source ip-address
```

An IP address is configured for the source VTEP.

By default, no IP address is configured for a source VTEP.

4. Run:

```
quit
```

Return to the system view.

Step 4 Bind the VPN instance to a Layer 3 gateway, enable distributed gateway, and configure host route advertisement.

1. Run:

```
interface vbdif bd-id
```

A VBDIF interface is created, and the VBDIF interface view is displayed.

By default, no VBDIF interface is created.

2. Run:

   ```
   ip binding vpn-instance vpn-instance-name
   ```

   A VPN instance is bound to the VBDIF interface.

   By default, no VPN instance is bound to any VBDIF interface.

3. Run:

   ```
   ip address ip-address { mask | mask-length } [ sub ]
   ```

   An IP address is configured for the VBDIF interface to implement Layer 3 interworking.

   By default, no IP address is configured for interfaces.

4. (Optional) Run:

   ```
   mac-address mac-address
   ```

   A MAC address is configured for the VBDIF interface.

   By default, the MAC address of a VBDIF interface is the system MAC address.

5. Run:

   ```
   arp distribute-gateway enable
   ```

   Distributed gateway is enabled.

   By default, distributed gateway is disabled.

   NOTE

   After distributed gateway is enabled on a Layer 3 gateway, the Layer 3 gateway discards network-side ARP messages and learns only user-side ARP messages.

6. Run:

   ```
   arp direct-route enable [ route-policy route-policy-name ]
   ```

   The VBDIF interface is enabled to advertise host routes.

   By default, VBDIF interfaces are disabled from advertising host routes.

7. Run:

   ```
   quit
   ```

   Return to the system view.

Step 5 Enable advertisement of the remote-nexthop attribute.

1. Run:

   ```
   bgp as-number-plain
   ```

   BGP is enabled, and the BGP view is displayed.

   By default, BGP is disabled.

2. Run:

   ```
   ipv4-family vpnv4
   ```

   The BGP-VPNv4 address family is enabled, and the BGP-VPNv4 address family view is displayed.

   By default, the BGP IPv4 address family is disabled.

3. Run:

   ```
   peer { group-name | ipv4-address } advertise remote-nexthop
   ```

   Advertisement of the remote-nexthop attribute is enabled.

   By default, advertisement of the remote-nexthop attribute is disabled.

Step 6 Run:

```
commit
```

The configuration is committed.

----End

9.4 (Optional) Configuring ARP Broadcast Suppression

When tenants communicate with each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks and may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on VXLAN gateways.

Context

After you enable ARP broadcast suppression on Layer 2 VXLAN gateway, configure Ethernet Virtual Network Border Gateway Protocol (EVN BGP) on Layer 2 and Layer 3 VXLAN gateways to allow ARP broadcast suppression to take effect. EVN BGP can then generate host information based on learned ARP entries. If a Layer 3 VXLAN gateway is enabled in the VBDIF interface view to use EVN BGP to advertise information to Layer 2 VXLAN gateways, the Layer 3 VXLAN gateway will advertise the host information generated by EVN BGP to Layer 2 VXLAN gateways. After the Layer 2 VXLAN gateways receive ARP broadcast packets, they convert the ARP broadcast packets into unicast packets based on the learned host information before forwarding the packets out. This decreases the number of broadcast packets in a BD, improving network performance.

Configuring BGP RRs is recommended to simplify BGP configuration. Spine nodes are generally used as RRs, and leaf nodes as RR clients.

Procedure

Step 1 Configure EVN BGP on a VXLAN gateway to establish EVN BGP peer relationships.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   evn bgp
   ```

   The EVN BGP view is displayed.

3. Run:

   ```
   source-address ip-address
   ```

   A source address is configured for establishing an EVN BGP peer relationship. This source address can be used to generate a router ID, a next-hop address, and an EVN instance's RD.

   By default, no source address is specified for establishing an EVN BGP peer relationship.

4. Run:

   ```
   peer ip-address
   ```

   An EVN BGP peer address is specified.

   By default, no EVN BGP peer is specified.

   If a BGP RR is deployed, each VXLAN gateway needs to establish an EVN BGP peer relationship only with the RR.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Configure a spine node as a BGP RR.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   evn bgp
   ```

   The EVN BGP view is displayed.

3. Configure a dynamic or static EVN BGP RR. Dynamic and static RRs cannot coexist. Determine which type of RR to configure based on actual requirements.

   - Run:

     ```
     server enable
     ```

     A dynamic RR is configured.

     After a dynamic RR is configured, all devices that establish EVN BGP peer relationships with the RR become the dynamic RR clients.

   - Run:

     ```
     peer ipv4-address reflect-client
     ```

     A static RR is configured.

     Only the specified peer can become a static RR client.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 3 Enable EVN BGP on a VXLAN gateway to collect host information.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   interface vbdif bd-id
   ```

   The VBDIF interface view is displayed.

3. Run:

   ```
   arp collect host enable
   ```

   EVN BGP is enabled to collect host information.

   By default, EVN BGP is disabled from collecting host information.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 4 Enable EVN BGP on a VXLAN gateway to advertise host information.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   host collect protocol bgp
   ```

   EVN BGP is enabled to advertise host information.

   By default, EVN BGP is disabled from advertising host information.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 5 Enable ARP broadcast suppression on a Layer 2 VXLAN gateway.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   bridge-domain bd-id
   ```

   The BD view is displayed.

3. Run:

   ```
   arp broadcast-suppress enable
   ```

   ARP broadcast suppression is enabled.

   By default, ARP broadcast suppression is disabled.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

9.5 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages

Context

When the device wants to communicate with another device in the same network segment, it queries ARP entries to direct packet forwarding. If the device fails to find the corresponding ARP entry from the forwarding plane, it sends an ARP Miss message to the CPU. The ARP Miss message will trigger the device to send an ARP broadcast packet to start ARP learning. In some cases, customers may want to limit the number of broadcast packets on the VXLAN network. You can then disable a VBDIF interface from sending ARP Miss messages to achieve this purpose.

After a VBDIF interface is disabled from sending ARP Miss messages, the device cannot learn ARP entries from this VBDIF interface, so ARP entries must be manually configured on it.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface vbdif bd-id
```

A VBDIF interface is created and the VBDIF interface view is displayed.

Step 3 Run:

```
arp miss disable
```

The VBDIF interface is disabled from sending ARP Miss messages.

By default, a VBDIF interface can send ARP Miss messages.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

9.6 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting

Static ARP entries or MAC address entries can be configured for traffic forwarding, and MAC address limiting can be configured to improve VXLAN security.

Context

- Static ARP entries are manually configured and maintained. They can be neither aged nor overwritten by dynamic ARP entries. Therefore, configuring static ARP entries on Layer 3 VXLAN gateways enhances communication security. If a static ARP entry is configured on a device, the device can communicate with a peer device that has a specified IP address using only the specified MAC address. Network attackers cannot modify the mapping between the IP and MAC addresses, which ensures communication between the two devices.
- After the source NVE on a VXLAN tunnel receives broadcast, unknown unicast, and multicast (BUM) packets, the local VTEP sends a copy of the BUM packets to every VTEP in the ingress replication list. Configuring static MAC address entries helps reduce broadcast traffic and prevent unauthorized data access from bogus users.
- The maximum number of MAC addresses that a device can learn can be configured to limit the number of access users and prevent against attacks on MAC address tables. If the device has learned the maximum number of MAC addresses allowed, no more addresses can be learned. The device can also be configured to discard packets after learning the maximum allowed number of MAC addresses, improving network security.
- If Layer 3 VXLAN gateway does not need to learn MAC addresses of packets in a BD, MAC address learning can be disabled from the BD to conserve MAC address entry resources. If the network topology of a VXLAN becomes stable and MAC address entry learning is complete, MAC address learning can also be disabled.

Configuring static MAC address entries and MAC address limiting applies to Layer 2 VXLAN gateways; configuring static ARP entries applies to Layer 3 VXLAN gateways; disabling MAC address limiting applies to both Layer 2 and Layer 3 VXLAN gateways.

Procedure

- Configure a static ARP entry.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  arp static ip-address mac-address vni vni-id source-ip source-ip peer-ip
  peer-ip
  ```

  A static ARP entry is configured.

  By default, no static ARP entry is configured.

  NOTE

  ip-address must belong to the same network segment as the Layer 3 gateway's IP address.

  c. Run:

  ```
  commit
  ```

  The configuration is committed.

- Configure a static MAC address entry.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  mac-address static mac-address bridge-domain bd-id source source-ip-
  address peer peer-ip vni vni-id
  ```

  A static MAC address entry is configured.

  By default, no static MAC address entry is configured.

  c. Run:

  ```
  commit
  ```

  The configuration is committed.

- Configure MAC address limiting.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  bridge-domain bd-id
  ```

  The BD view is displayed.

  c. Run:

  ```
  mac-address limit { action { discard | forward } | maximum max | alarm
  { disable | enable } } *
  ```

  MAC address limiting is configured.

  By default, MAC address limiting is not configured.

  d. Run:

  ```
  commit
  ```

  The configuration is committed.

- Disable MAC address learning.

a. Run:

```
system-view
```

The system view is displayed.

b. Run:

```
bridge-domain bd-id
```

The BD view is displayed.

c. Run:

```
mac-address learning disable
```

MAC address learning is disabled.

By default, MAC address learning is enabled for a BD.

d. Run:

```
commit
```

The configuration is committed.

----End

9.7 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection

In specific VXLAN applications, when a device connects to a load balancing server equipped with two network interface cards, the server's MAC address may be learned by two interfaces on the device. This is a normal situation where MAC address flapping detection is not needed. In this case, configure a VXLAN BD whitelist for MAC address flapping detection.

Context

By default, MAC address flapping detection is enabled globally. After a BD is added to a MAC address flapping detection whitelist, detection is not performed for this BD. Even if MAC address flapping occurs in the BD, the occurrence generates neither an alarm nor a record.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
mac-address flapping detection exclude bridge-domain bd-id1 [ to bd-id2 ]
```

A VXLAN BD whitelist for MAC address flapping detection is configured.

By default, no VXLAN BD whitelist for MAC address flapping detection is configured.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

9.8 (Optional) Optimizing ACL Resource Usage on the VXLAN

Context

If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, BD traffic statistics collection, and DHCP on a switch deployed with VXLAN services, there is high probability that the other services fail to be configured. In this case, use either of the following methods to reduce the probability of service deployment failures.

- Enabling the VXLAN service extension function:

  This function takes effect only on VXLAN-enabled cards except the CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA series.

  NOTE

  After this function is enabled, only CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA can forward packets with 230 to 294 bytes received from the VXLAN tunnels to cards of the same series.

- Disabling an Eth-Trunk from load balancing VXLAN packets in optimized mode:
  - You are advised to disable load balancing of VXLAN packets in optimized mode when neither the user side nor the network side of a VXLAN tunnel is an Eth-Trunk.
  - You can disable load balancing of VXLAN packets in optimized mode when either the user side or the network side of a VXLAN tunnel is an Eth-Trunk. After this function is disabled, an Eth-Trunk does not evenly load balance VXLAN packets or does not perform load balancing.

  NOTE

  This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.

Procedure

Step 1 Enable the VXLAN service extension function.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 service extend enable
   ```

   The VXLAN service extension function is enabled.

   By default, the VXLAN service extension function is disabled.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Disable an Eth-Trunk from load balancing VXLAN packets in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 eth-trunk hash disable
   ```

   An Eth-Trunk is disabled from load balancing VXLAN packets in optimized mode.

   By default, an Eth-Trunk load balances VXLAN packets in optimized mode.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

9.9 (Optional) Optimizing Load Balancing on the VXLAN Network

Context

On a VXLAN network, VXLAN packets can be load balanced through ECMP or Eth-Trunks. To enable load balancing or improve the load balancing effect, enable either of the following functions:

- Enable load balancing of VXLAN packets through ECMP in optimized mode.
- Enable an Eth-Trunk to load balance VXLAN packets in optimized mode.

NOTE

This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.

Procedure

Step 1 Enable load balancing of VXLAN packets through ECMP in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 ecmp hash enable
   ```

   Load balancing of VXLAN packets through ECMP in optimized mode is enabled.

   By default, load balancing of VXLAN packets through ECMP in optimized mode is disabled.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Enable an Eth-Trunk to load balance VXLAN packets in optimized mode.

1. Run:

   ```
   system-view
   ```

The system view is displayed.

2. Run:

```
assign forward nvo3 eth-trunk hash enable
```

An Eth-Trunk is enabled to load balance VXLAN packets in optimized mode.

By default, an Eth-Trunk is disabled from load balancing VXLAN packets in optimized mode.

3. Run:

```
commit
```

The configuration is committed.

----End

9.10 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode

Context

By default, a Layer 3 VXLAN gateway works in loopback mode. That is, the Layer 3 VXLAN gateway first loops back VXLAN-encapsulated packets, and then forwards the packets. In this situation, traffic transmitted over the VXLAN tunnel cannot be forwarded at line speed when the traffic volume is heavy.

To solve the preceding problem, configure the Layer 3 VXLAN gateway to work in non-loopback mode.

- Layer 2 non-loopback mode: The device determines whether to loop back the received packets according to the destination MAC address in the inner Ethernet header of VXLAN packets where the received packets are encapsulated. If the destination MAC address is not the MAC address of the VXLAN Layer 3 gateway, the device directly forwards the packets at line speed.
- Layer 3 non-loopback mode: The device determines whether to loop back the received packets according to the destination MAC address in the inner Ethernet header of VXLAN packets where the received packets are encapsulated. If the destination MAC address is the MAC address of the VXLAN Layer 3 gateway, the device directly forwards the packets at line speed.

NOTE

- GE interface cards do not support the Layer 2 non-loopback mode. Among non-GE interface cards, EA series cards support only the Layer 2 non-loopback mode, and other series cards support Layer 2 and Layer 3 non-loopback modes.
- Traffic will be interrupted when you change the enhanced mode of a Layer 3 VXLAN gateway. Therefore, you are advised to change the enhanced mode of a Layer 3 VXLAN gateway when there is no or little traffic on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
assign forward nvo3-gateway enhanced { l2 | l3 }
```

The Layer 3 VXLAN gateway is configured to work in non-loopback mode.

By default, the enhanced mode of the Layer 3 VXLAN gateway is not set and the switch works in loopback mode.

NOTE

When you set the enhanced mode of the VXLAN gateway to Layer 3 non-loopback, you also need to run the **assign forward nvo3 service extend enable** command to enable the VXLAN service extension function. In this case, ensure that the switch does not have CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA cards installed, or the CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA cards do not transmit the VXLAN service.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

9.11 Checking the Configurations

After configuring VXLAN in distributed gateway mode using MP-BGP, check the configurations, and you can find that VXLAN tunnels are dynamically established and are in the Up state.

Prerequisites

VXLAN in distributed gateway mode has been configured using MP-BGP.

Procedure

- Run the **display bridge-domain** [*bd-id* [**brief** | **verbose**]] command to check BD configurations.
- Run the **display interface nve** [*nve-number* | **main**] command to check NVE interface information.
- Run the **display vxlan peer** [**vni** *vni-id*] command to check ingress replication lists of a VNI or all VNIs.
- Run the **display vxlan tunnel** [*tunnel-id*] [**verbose**] command to check VXLAN tunnel information.
- Run the **display vxlan vni** [*vni-id* [**verbose**]] command to check VNI information.
- Run the **display interface vbdif** [*bd-id*] command to check VBDIF interface information and statistics.
- Run the **display dfs-group** *dfs-group-id* **active-active-gateway** command to check information of all-active gateways in a DFS group.
- Run the **display arp broadcast-suppress user bridge-domain** *bd-id* command to check the ARP broadcast suppression table of a BD.
- Run the **display arp** [**network** *network-address* [*network-mask* | *mask-length*]] **static** command to check static ARP entries.

- Run the **display mac-address static bridge-domain** *bd-id* command to check static MAC address entries in a BD.
- Run the **display mac-address limit bridge-domain** *bd-id* command to check MAC address limiting configurations of a BD.
- Run the **display mac-address flapping** command to check the MAC address flapping detection configuration.

----End

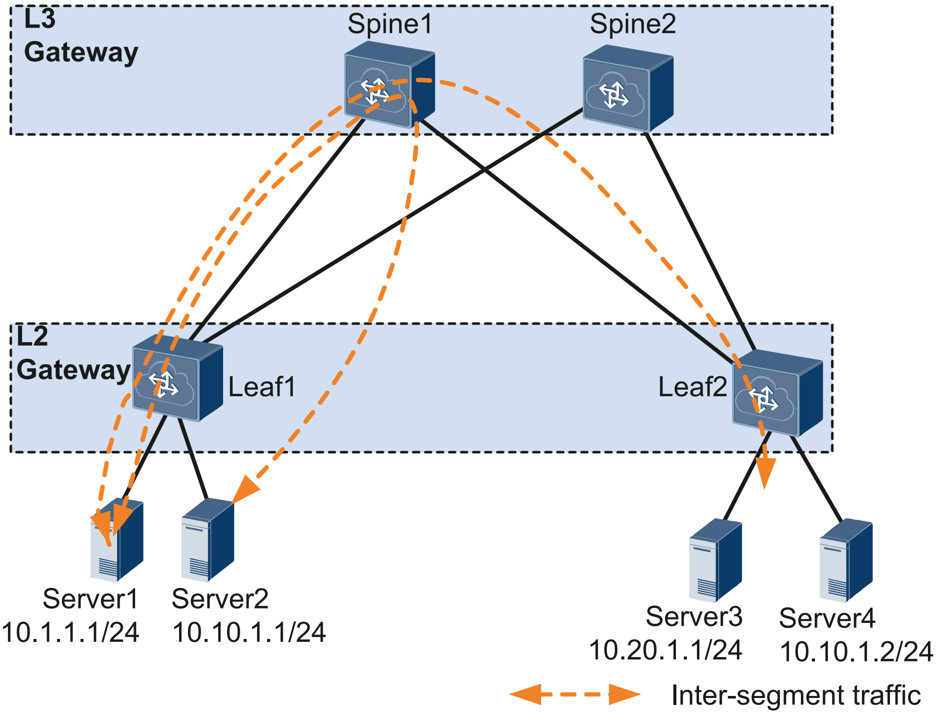
10 Configuring VXLAN in Single-Node, Distributed Gateway, and BGP EVPN Mode

About This Chapter

Distributed VXLAN gateways can be configured to address problems that occur in legacy centralized VXLAN gateway networking, for example, forwarding paths are not optimal, and the ARP entry specification is a bottleneck.

Usage Scenario

Figure 10-1 Centralized VXLAN gateway networking

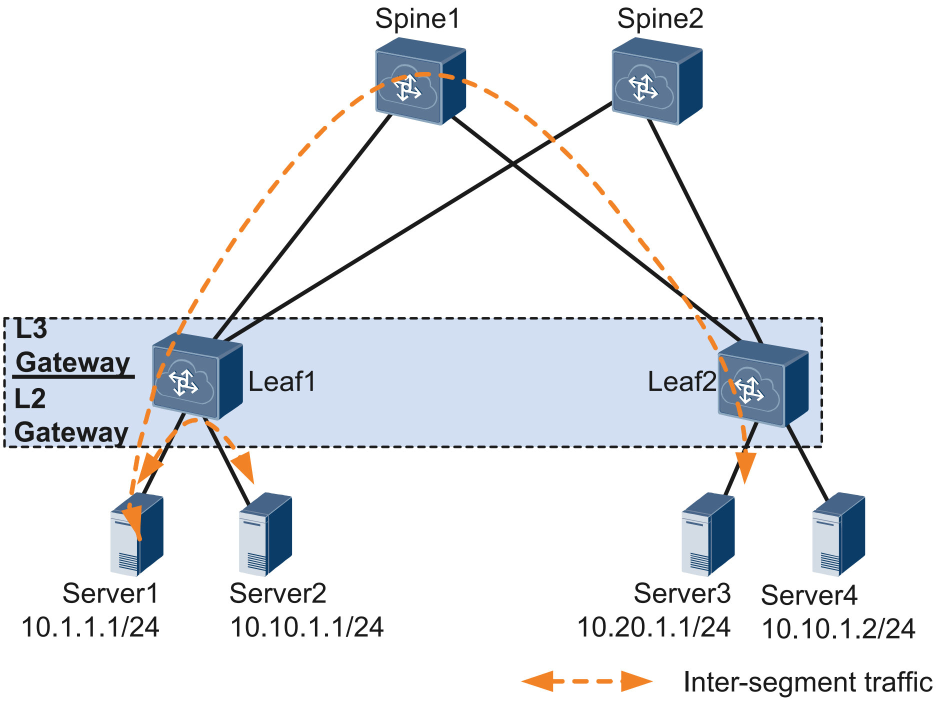


In legacy networking, a centralized Layer 3 gateway is deployed on a spine node. On the network shown in **Figure 9-1**, packets across different networks must be forwarded through a centralized Layer 3 gateway, resulting in the following problems:

- Forwarding paths are not optimal. All Layer 3 traffic must be transmitted to the centralized Layer 3 gateway for forwarding.
- The ARP entry specification is a bottleneck. ARP entries must be generated for tenants on the Layer 3 gateway. However, only a limited number of ARP entries can be configured for the Layer 3 gateway, impeding data center network expansion.

To address these problems, distributed VXLAN gateways can be configured. On the network shown in **Figure 9-2**, Server 1 and Server 2 on different network segments both connect to Leaf 1. When Server 1 and Server 2 communicate, traffic is forwarded only through Leaf 1, not through any spine node.

Figure 10-2 Distributed VXLAN gateway networking



Distributed VXLAN gateway networking has the following characteristics:

- Flexible deployment. A leaf node can function as both Layer 2 and Layer 3 VXLAN gateways.
- Improved network expansion capabilities. A leaf node only needs to learn the ARP entries of servers attached to it. A centralized Layer 3 gateway in the same scenario, however, has to learn the ARP entries of all servers on the network. Therefore, the ARP entry specification is no longer a bottleneck on a distributed VXLAN gateway.
- Enhanced network performance: Changing broadcast packets to unicast packets, leaf nodes can determine whether to broadcast ARP request packets received from tenants or servers. This reduces the number of ARP broadcast packets and improves network performance.

Pre-configuration Tasks

Before configuring VXLAN in distributed gateway mode, ensure that reachable routes are available.

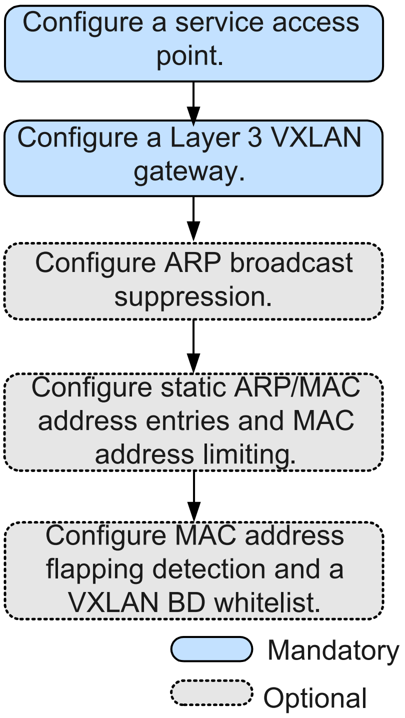
Configuration Procedures

NOTE

The procedure for configuring intra-segment communication through distributed VXLAN gateways is the same as that for configuring intra-segment communication through centralized VXLAN gateways. For details, see **8 Configuring VXLAN in Single-Node, Centralized Gateway, and BGP EVPN Mode**. This chapter describes the procedure for configuring inter-segment communication through distributed VXLAN gateways.

Figure 10-3 shows the flowchart for configuring inter-segment communication through distributed VXLAN gateways.

Figure 10-3 Flowchart for configuring inter-segment communication through distributed VXLAN gateways



Configure a service access point.
Configure a Layer 3 VXLAN gateway.
Configure ARP broadcast suppression.
Configure static ARP/MAC address entries and MAC address limiting.
Configure MAC address flapping detection and a VXLAN BD whitelist.
Mandatory
Optional

10.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function

10.2 Configuring a VXLAN Service Access Point
On VXLANs, Layer 2 sub-interfaces are used as service access points. These Layer 2 sub-interfaces can have different encapsulation types configured to transmit various types of data packets. A BD is a broadcast domain. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD.

10.3 Configuring a VXLAN Tunnel and a Layer 3 VXLAN Gateway
To configure a Layer 3 VXLAN gateway using EVPN, configure EVPN as the VXLAN control plane, establish a BGP EVPN peer relationship, configure an EVPN instance,

configure a VPN instance, configure ingress replication, configure the type of route to be advertised between VXLAN gateways,and bind the VPN instance to a Layer 3 VXLAN gateway.

10.4 (Optional) Configuring ARP Broadcast Suppression
When tenants communicate with each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks and may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on Layer 2 VXLAN gateways.

10.5 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages

10.6 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting
Static ARP entries or MAC address entries can be configured for traffic forwarding, and MAC address limiting can be configured to improve VXLAN security.

10.7 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection
In specific VXLAN applications, when a device connects to a load balancing server equipped with two network interface cards, the server's MAC address may be learned by two interfaces on the device. This is a normal situation where MAC address flapping detection is not needed. In this case, configure a VXLAN BD whitelist for MAC address flapping detection.

10.8 (Optional) Configuring IP Address Conflict Detection Parameters
On VXLANs, IP address conflicts of terminal users will prevent these users from going online. Therefore, IP address conflicts must be detected.

10.9 (Optional) Optimizing ACL Resource Usage on the VXLAN

10.10 (Optional) Optimizing Load Balancing on the VXLAN Network

10.11 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode

10.12 Checking the Configurations
After configuring VXLAN in distributed gateway mode using BGP EVPN, check the configurations, and you can find that VXLAN tunnels are dynamically established and are in the Up state.

10.1 Configuring the VXLAN Tunnel Mode and Enabling the VXLAN ACL Extension Function

Context

- Configuring a tunnel mode: You need to set the tunnel mode to VXLAN when configuring the VXLAN feature; otherwise, the configurations do not take effect.
- Enabling the VXLAN ACL extension function: By default, the VXLAN ACL extension function is disabled on the device. If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, and BD traffic statistics collection, on the device deployed with VXLAN services, there is high probability that the other services fail to be configured. You can enable the VXLAN ACL extension function to lower the configuration failure probability.

To ensure normal forwarding of VXLAN packets, the VXLAN tunnel mode must have been configured and the VXLAN ACL extension function must have been enabled on Layer 2 and Layer 3 VXLAN gateways.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
ip tunnel mode vxlan
```

The tunnel mode is set to VXLAN.

By default, the tunnel mode is VXLAN.

NOTE

Only the Admin-VS in port mode supports this command, and all VSs in group mode support this command.

Step 3 Run:

```
assign forward nvo3 acl extend enable
```

The VXLAN ACL extension function is enabled.

By default, the VXLAN ACL extension function is disabled.

NOTE

You can run this command on the admin-VS only. The configuration takes effect for all the VSs.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

Follow-up Procedure

After configuring the VXLAN tunnel mode and enabling the VXLAN ACL extension function, you need to save the configuration and restart the switch to make the configuration take effect.

10.2 Configuring a VXLAN Service Access Point

On VXLANs, Layer 2 sub-interfaces are used as service access points. These Layer 2 sub-interfaces can have different encapsulation types configured to transmit various types of data packets. A BD is a broadcast domain. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD.

Context

When a Layer 2 sub-interface is used as a service access point, different encapsulation types can be configured for the sub-interface to transmit various types of data packets. After a Layer 2 sub-interface is added to a BD, the sub-interface can transmit data packets through this BD. **Table 10-1** describes the different encapsulation types.

Table 10-1 Traffic encapsulation types

Traffic Encapsulation Type	Description
dot1q	If a Dot1q sub-interface receives a single-tagged VLAN packet, the sub-interface forwards only the packet with a specifie VLAN ID. If a Dot1q sub-interface receives a double-tagged VLAN packet, the sub-interface forwards only the packet with a specified outer VLAN ID. ● When performing VXLAN encapsulation on packets, a Dot1q Layer 2 sub-interface removes the outer tags of the packets. ● When performing VXLAN decapsulation on packets, a Dot1q Layer 2 sub-interface replaces the VLAN tags with specified VLAN tags if the inner packets carry VLAN tags, or adds specified VLAN tags to the packets if the inner packets do not carry VXLAN tags. When setting the encapsulation type to **dot1q** for a Layer 2 sub-interface, note the following: ● The VLAN IDs specified for the Layer 2 sub-interface cannot be the same as either the VLAN IDs of packets allowed to pass through the corresponding Layer 2 interfaces or the MUX VLAN IDs. ● Layer 2 and Layer 3 sub-interfaces cannot have the same VLAN IDs specified.

Traffic Encapsulation Type	Description
untag	An untagged Layer 2 sub-interface receives only packets that do not carry VLAN tags. ● When performing VXLAN encapsulation on packets, an untagged Layer 2 sub-interface does not add any VLAN tag to the packets. ● When performing VXLAN decapsulation on packets, an untagged Layer 2 sub-interface removes the VLAN tags of single-tagged inner packets or the outer VLAN tags of double-tagged inner packets. When setting the encapsulation type to **untag** for a Layer 2 sub-interface, note the following: ● Ensure that the corresponding physical interface of the sub-interface does not have any configuration, and is removed from the default VLAN. ● Untagged Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● An interface can have only one untagged Layer 2 sub-interface configured.
qinq	A QinQ sub-interface receives only tagged packets with specified inner and outer VLAN tags. ● When performing VXLAN encapsulation on packets, a QinQ sub-interface removes two VLAN tags from packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintains the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. ● When performing VXLAN decapsulation on packets, a QinQ sub-interface adds two specific VLAN tags to packets if the action of the Layer 2 sub-interface is set to removing two VLAN tags and maintain the VLAN tags of packets if the action of the Layer 2 sub-interface is not set to removing two VLAN tags. **NOTE** The traffic behavior for QinQ interfaces bound to the same BD must be the same. QinQ interfaces do not support DHCP Snooping or VBDIF and cannot be bound to the same BD as Dot1q sub-interfaces. A QinQ interface can have only one outer VLAN tag and one inner VLAN tag.

Traffic Encapsulation Type	Description
default	A default Layer 2 sub-interface receives all packets, irrespective of whether the packets carry VLAN tags. When performing VXLAN encapsulation and decapsulation on packets, a default Layer 2 sub-interface does not process VLAN tags of the packets. When setting the encapsulation type to **default** for a Layer 2 sub-interface, note the following: ● Ensure that the interface for the Layer 2 sub-interface is not added to any VLAN. ● Default Layer 2 sub-interfaces can be configured only for Layer 2 physical interfaces and Eth-Trunk interfaces. ● If a default Layer 2 sub-interface is created for an interface, the interface cannot have other types of Layer 2 sub-interfaces configured.

NOTE

When a sub-interface that is configured with dot1q and QinQ receives double-tagged VLAN packets, the QinQ sub-interface preferentially processes the packets. For example, if a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of 20, the QinQ sub-interface preferentially processes the packet. If a dot1q and QinQ sub-interface carries the VLAN ID of 10 for dot1q and outer VLAN ID of 10 and inner VLAN ID of 20 for QinQ and receives a packet with the outer VLAN ID of 10 and inner VLAN ID of non-20, the dot1q sub-interface preferentially processes the packet.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
bridge-domain bd-id
```

A BD is created, and the BD view is displayed.

By default, no BD is created.

Step 3 (Optional) Run:

```
description description
```

A description is configured for the BD.

By default, no description is configured for a BD.

Step 4 Run:

```
quit
```

Return to the system view.

Step 5 Run:

```
interface interface-type interface-number.subnum mode l2
```

A Layer 2 sub-interface is created, and the sub-interface view is displayed.

By default, no Layer 2 sub-interface is created.

Before running this command, ensure that the Layer 2 interface for which a Layer 2 sub-interface is created does not have the **port link-type dot1q-tunnel** command configuration. If this configuration exists, run the **undo port link-type** command to delete the configuration.

Step 6 Run:

```
encapsulation { dot1q [ vid ce-vid ] | default | untag  | qinq [ vid pe-vid ce-
vid ce-vid ] }
```

An encapsulation type is configured for the Layer 2 sub-interface.

By default, no encapsulation type is configured for Layer 2 sub-interfaces.

Step 7 (Optional) Run:

```
rewrite pop double
```

The sub-interface is enabled to remove double VLAN tags from received packets if the encapsulation type of the sub-interface is set to QinQ in **Step 6**.

By default, a Layer 2 sub-interface with the encapsulation type being QinQ is enabled to transparently transmit received packets.

Step 8 Run:

```
bridge-domain bd-id
```

The Layer 2 sub-interface is added to a BD so that the sub-interface can transmit data packets through this BD.

By default, the Layer 2 sub-interface is not added to a BD.

Step 9 Run:

```
commit
```

The configuration is committed.

----End

10.3 Configuring a VXLAN Tunnel and a Layer 3 VXLAN Gateway

To configure a Layer 3 VXLAN gateway using EVPN, configure EVPN as the VXLAN control plane, establish a BGP EVPN peer relationship, configure an EVPN instance, configure a VPN instance, configure ingress replication, configure the type of route to be advertised between VXLAN gateways,and bind the VPN instance to a Layer 3 VXLAN gateway.

Context

VXLAN packets are transmitted through VXLAN tunnels. In distributed VXLAN gateway scenarios, perform the following steps on a VXLAN gateway to use EVPN for establishing VXLAN tunnels:

1. Configure EVPN as the VXLAN control plane. Subsequent EVPN configurations can then be performed.
2. Configure a BGP EVPN peer relationship. Configure VXLAN gateways to establish BGP EVPN peer relationships so that they can exchange EVPN routes. If an RR has been deployed, each VXLAN gateway only needs to establish a BGP EVPN peer relationship with the RR.
3. (Optional) Configure an RR. The deployment of RRs reduces the number of BGP EVPN peer relationships to be established, simplifying configuration. A live-network device can be used as an RR, or a standalone RR can be deployed. Spine nodes are generally used as RRs, and leaf nodes as RR clients.
4. Configure an EVPN instance. EVPN instances are used to receive and advertise EVPN routes.
5. Configure ingress replication. After ingress replication is configured for a VNI, the system uses BGP EVPN to construct a list of remote VTEPs. After a VXLAN gateway receives BUM packets, its sends a copy of the BUM packets to every VXLAN gateway in the list.
6. Configure a VPN instance whose routes can be installed into the routing table of the EVPN instance. This VPN instance is used to store host routes or network segment routes.
7. Configure the type of route to be advertised between VXLAN gateways. VXLAN gateways can send different information through different types of routes. If an RR is deployed on the network, only the type of route to be advertised between the RR and VXLAN gateways needs to be configured.
8. Bind the VPN instance to a Layer 3 VXLAN gateway, enable distributed gateway, and configure host route advertisement.

NOTE

If tenants on the same network segment connect to different Layer 3 VXLAN gateways, the Layer 3 VXLAN gateways must have the same IP address and MAC address configured. When tenants are moved to a different location, the tenants can retain Layer 3 gateway configurations, reducing maintenance workload.

BUM packet forwarding is implemented only using ingress replication. To establish a VXLAN tunnel between a Huawei device and a non-Huawei device, ensure that the non-Huawei device also has ingress replication configured. Otherwise, communication fails.

Procedure

Step 1 Configure EVPN as the VXLAN control plane.

1. Run:
   ```
   system-view
   ```
   The system view is displayed.
2. Run:
   ```
   evpn-overlay enable
   ```
   EVPN is configured as the VXLAN control plane.

   By default, EVPN is not configured as the VXLAN control plane.
3. Run:
   ```
   commit
   ```
   The configuration is committed.

Step 2 Configure a BGP EVPN peer relationship. If an RR has been deployed, each VXLAN gateway only needs to establish a BGP EVPN peer relationship with the RR. If the spine node and gateway reside in different ASs, the gateway must establish an EBGP EVPN peer relationship with the spine node.

1. Run:

   ```
   bgp as-number [ instance instance-name ]
   ```

   BGP is enabled, and the BGP or BGP multi-instance view is displayed.

   By default, the BGP is disabled. If an RR has been deployed, each VXLAN gateway only needs to establish a BGP EVPN peer relationship with the RR.

2. (Optional) Run:

   ```
   router-id ipv4-address
   ```

   A router ID is set.

   By default, no BGP Router ID is configured, and the Router ID configured for the route management module through the **router id** command is used.

3. Run:

   ```
   peer ipv4-address as-number as-number
   ```

   The peer device is configured as a BGP peer.

   By default, no BGP peer is configured, and no AS number is specified for a peer or peer group.

4. (Optional) Run:

   ```
   peer ipv4-address connect-interface interface-type interface-number [ ipv4-
   source-address ]
   ```

   A source interface and a source address are specified to set up a TCP connection with the BGP peer.

   By default, the outbound interface of a BGP packet serves as the source interface of a BGP packet.

   NOTE

   When loopback interfaces are used to establish a BGP connection, running the **peer connect-interface** command on both ends is recommended to ensure the connectivity. If this command is run on only one end, the BGP connection may fail to be established.

5. (Optional) Run:

   ```
   peer ipv4-address ebgp-max-hop [ hop-count ]
   ```

   The maximum number of hops is set for an EBGP EVPN connection.

   The default value of *hop-count* is 255.

   In most cases, a directly connected physical link must be available between EBGP EVPN peers. If you want to establish EBGP EVPN peer relationships between indirectly connected peers, run the **peer ebgp-max-hop** command. The command also can configure the maximum number of hops for an EBGP EVPN connection.

   NOTE

   When the IP address of loopback interface to establish an EBGP EVPN peer relationship, run the **peer ebgp-max-hop** (of which the value of hop-count is not less than 2) command. Otherwise, the peer relationship fails to be established.

6. Run:

```
l2vpn-family evpn
```

The BGP-EVPN address family view or BGP multi-instance EVPN view is displayed.

By default, the BGP-EVPN address family or BGP multi-instance EVPN view is disabled.

7. Run:

```
peer { ipv4-address | group-name } enable
```

The device is enabled to exchange EVPN routes with a specified peer or peer group.

By default, only the peer in the BGP IPv4 unicast address family view is automatically enabled.

8. (Optional) Run:

```
peer { group-name | ipv4-address } route-policy route-policy-name { import |
export }
```

A routing policy is specified for routes received from or to be advertised to a BGP EVPN peer or peer group.

After the routing policy is applied, the routes received from or to be advertised to a specified BGP EVPN peer or peer group will be filtered, ensuring that only desired routes are imported or advertised. This configuration helps manage routes and reduce required routing entries and system resources.

9. (Optional) Run:

```
peer { ipv4-address | group-name } next-hop-invariable
```

The device is prevented from changing the next hop address of a route when advertising the route to an EBGP peer. If the spine node and gateway have established an EBGP EVPN peer relationship, run the **peer next-hop-invariable** command to ensure that the next hops of routes received by the gateway point to other gateways.

By default, a BGP EVPN speaker changes the next hops of routes to the interface that it uses to establish EBGP EVPN peer relationships before advertising these routes to EBGP EVPN peers.

10. (Optional) Run:

```
peer { group-name | ipv4-address } mac-limit number [ percentage ] [ alert-
only | idle-forever | idle-timeout times ]
```

The maximum number of MAC advertisement routes that can be received from each peer is configured.

If an EVPN instance may import many invalid MAC advertisement routes from peers and these routes occupy a large proportion of the total MAC advertisement routes. If the received MAC advertisement routes exceed the specified maximum number, the system displays an alarm, instructing users to check the validity of the MAC advertisement routes received in the EVPN instance.

11. Run:

```
quit
```

Exit from the BGP-EVPN address family view or BGP multi-instance EVPN view.

12. Run:

```
quit
```

Exit from the BGP or BGP multi-instance view.

13. Run:

```
commit
```

The configuration is committed.

Step 3 (Optional) Configure an RR. If an RR is configured, each VXLAN gateway only needs to **establish a BGP EVPN peer relationship** with the RR, reducing the number of BGP EVPN peer relationships to be established and simplifying configuration.

1. Run:

   ```
   bgp as-number [ instance instance-name ]
   ```

   The BGP or BGP multi-instance view is displayed.

2. Run:

   ```
   l2vpn-family evpn
   ```

   The BGP-EVPN address family view or BGP multi-instance EVPN view is displayed.

3. Run:

   ```
   peer { ipv4-address | group-name } reflect-client
   ```

   The device is configured as an RR and an RR client is specified.

   By default, the route reflector and its client are not configured.

4. Run:

   ```
   undo policy vpn-target
   ```

   The function to filter received EVPN routes based on VPN targets is disabled. If you do not perform this step, the RR will fail to receive and reflect the routes sent by clients.

5. Run:

   ```
   quit
   ```

   Exit from the BGP-EVPN address family view or BGP multi-instance EVPN view.

6. Run:

   ```
   quit
   ```

   Exit from the BGP or BGP multi-instance view.

7. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 4 Configure an EVPN instance.

1. Run:

   ```
   bridge-domain bd-id
   ```

   The BD view is displayed.

   By default, no bridge domain is created.

2. Run:

   ```
   vxlan vni vni-id
   ```

   A VNI is created and mapped to the BD.

   By default, no VNI is created.

3. Run:

   ```
   evpn
   ```

   An EVPN instance is created.

By default, no EVPN instance is created for VXLANs.

4. Run:

   ```
   route-distinguisher { route-distinguisher | auto }
   ```

   An RD is configured for the EVPN instance. The two ends of a VXLAN tunnel can share an RD or use different RDs.

   By default, no RD is configured for BD EVPN instances.

5. Run:

   ```
   vpn-target { vpn-target &<1-8> | auto } [ both | export-extcommunity | import-
   extcommunity ]
   ```

   VPN targets are configured for the EVPN instance. The export VPN target of the local end must be the same as the import VPN target of the remote end, and the import VPN target of the local end must be the same as the export VPN target of the remote end.

   By default, no VPN target is configured for BD EVPN instances.

6. (Optional) Run:

   ```
   import route-policy policy-name
   ```

   The current EVPN instance is associated with an import routing policy.

   By default, an EVPN instance matches the export VPN targets of received routes against its import VPN targets to determine whether to import these routes. To control route import more precisely, perform this step to associate the EVPN instance with an import routing policy and set attributes for eligible routes.

7. (Optional) Run:

   ```
   export route-policy policy-name
   ```

   The current EVPN instance is associated with an export routing policy.

   By default, an EVPN instance adds all VPN targets in the export VPN target list to EVPN routes to be advertised to its peers. To control route export more precisely, perform this step to associate the EVPN instance with an export routing policy and set attributes for eligible routes.

8. Run:

   ```
   quit
   ```

   The EVPN instance view is exited.

9. Run:

   ```
   quit
   ```

   Return to the system view.

10. Run:

    ```
    commit
    ```

    The configuration is committed.

Step 5 Configure an ingress replication list.

1. Run:

   ```
   interface nve nve-number
   ```

   An NVE interface is created, and the NVE interface view is displayed.

2. Run:

   ```
   source ip-address
   ```

An IP address is configured for the source VTEP.

By default, no IP address is configured for any source VTEP.

3. Run:

```
vni vni-id head-end peer-list protocol bgp
```

An ingress replication list is configured.

By default, no ingress replication list is configured for any VNI.

NOTE

BUM packet forwarding is implemented only using ingress replication. To establish a VXLAN tunnel between a Huawei device and a non-Huawei device, ensure that the non-Huawei device also has ingress replication configured. Otherwise, communication fails.

4. Run:

```
quit
```

Return to the system view.

5. Run:

```
commit
```

The configuration is committed.

Step 6 (Optional) Configure a MAC address for an NVE interface.

When BGP EVPN is deployed between distributed VXLAN gateways, you need to configure the same VTEP MAC address for the two devices that provide dual-active VXLAN access. In this way, gateways on the VXLAN network can forward traffic properly.

1. Run:

```
interface nve nve-number
```

The NVE interface view is displayed.

2. Run:

```
mac-address mac-address
```

A MAC address is configured for the NVE interface.

By default, the MAC address of an NVE interface is the system MAC address.

3. Run:

```
quit
```

Return to the system view.

4. Run:

```
commit
```

The configuration is committed.

Step 7 Configure a VPN instance whose routes can be installed into the routing table of the EVPN instance.

1. Run:

```
ip vpn-instance vpn-instance-name
```

A VPN instance is created, and the VPN instance view is displayed.

By default, no VPN instance is created.

2. Run:

   ```
   vxlan vni vni-id
   ```

   A VNI is created and mapped to the VPN instance.

   By default, a VNI is not bound to any VPN instance.

3. Run:

   ```
   ipv4-family
   ```

   The IPv4 address family is enabled for the VPN instance, and the VPN instance IPv4 address family view is displayed.

   By default, the IPv4 address family is disabled for a VPN instance.

4. Run:

   ```
   route-distinguisher route-distinguisher
   ```

   An RD is configured for the VPN instance IPv4 address family.

   By default, no RD is configured for the VPN instance IPv4 address family.

5. Run:

   ```
   vpn-target vpn-target &<1-8> [ both | export-extcommunity | import-
   extcommunity ]
   ```

   VPN targets are configured for the VPN instance IPv4 address family.

   By default, no VPN target is configured for a VPN instance IPv4 address family.

   A VPN target is the extended community attribute of BGP. It controls reception and advertisement of VPN routes. A maximum of eight VPN targets can be configured each time the **vpn-target** command is run. To configure more VPN targets for the VPN instance IPv4 address family, run the **vpn-target** command several times.

6. Run:

   ```
   vpn-target vpn-target &<1-8> [ both | export-extcommunity | import-
   extcommunity ] evpn
   ```

   VPN targets are configured for the VPN instance IPv4 address family for route installation into the EVPN instance. *vpn-target* specified here must be the same as the RT configured for the EVPN instance in the BD view. This implementation ensures that routes in the VPN instance can be installed into the routing table of the specified EVPN instance.

7. (Optional) Run:

   ```
   import route-policy policy-name evpn
   ```

   The VPN instance IPv4 address family of the current VPN instance is associated with an import routing policy to filter routes imported from the EVPN instance.

   By default, the VPN instance IPv4 address family of a VPN instance matches the export VPN targets of received routes against its import VPN targets to determine whether to import these routes. To control route import more precisely, perform this step to associate the VPN IPv4 address family with an import routing policy and set attributes for eligible routes.

8. (Optional) Run:

   ```
   export route-policy policy-name evpn
   ```

   The VPN instance IPv4 address family of the current VPN instance is associated with an export routing policy to filter routes to be advertised to the EVPN instance.

By default, the VPN IPv4 address family adds all VPN targets in the export VPN target list to routes to be advertised to the EVPN instance. To control route export more precisely, perform this step to associate the VPN IPv4 address family with an export routing policy and set attributes for eligible routes.

9. Run:

```
quit
```

The VPN instance IPv4 address family view is exited.

10. Run:

```
quit
```

The VPN instance view is exited.

11. Run:

```
commit
```

The configuration is committed.

Step 8 Configure the type of route to be advertised between VXLAN gateways. If an RR is deployed on the network, only the type of route to be advertised between the RR and VXLAN gateways needs to be configured.

VXLAN gateways can advertise the forwarding types of routes. **Table 10-2** lists the scenario and deployment for each type of route. By default, the type of route to be advertised is not specified.

Table 10-2 Configuring the type of route to be advertised between VXLAN gateways

Route Type	Usage Scenario	Configuration Procedure
Configure IRB route advertisement.	In scenarios where ARP broadcast suppression is needed, configure IRB route advertisement before running the **arp broadcast-suppress enable** command. VXLAN gateways can also use IRB routes to advertise host routes. NOTE After configuring IRB route advertisement, you must run the **arp collect host enable** command for host route advertisement.	1. Run the **bgp** *as-number* [**instance** *instance-name*] command to enter the BGP or BGP multi-instance view. 2. Run the **l2vpn-family evpn** command to enter the BGP-EVPN address family view or BGP multi-instance EVPN view. 3. Run the **peer** { *ipv4-address* | *group-name* } **advertise irb** command to configure IRB route advertisement. 4. Run the **quit** command to exit the BGP-EVPN address family view or BGP multi-instance EVPN view. 5. Run the **quit** command to exit the BGP or BGP multi-instance view. 6. Run the **commit** command to commit the configuration.

Route Type	Usage Scenario	Configuration Procedure
Configure IP prefix route advertisement.	Though IP prefix route advertisement cannot implement ARP broadcast suppression, it allows advertisement of network segment routes for host routes between VXLAN gateways. If a large number of specific host routes are available, you can configure IP prefix route advertisement so that VXLAN gateways do not have to store all these routes. **NOTE** ● A VXLAN gateway can advertise network segment routes only if the network segments attached to the gateway are unique network-wide. ● After configuring IP prefix route advertisement, you must run the **arp direct-route enable** command for host route advertisement. Then, host migration will be affected. To avoid this problem, configure IRB route advertisement.	1. Run the **bgp** *as-number* [**instance** *instance-name*] command to enter the BGP or BGP multi-instance view. 2. Run the **ipv4-family vpn-instance** *vpn-instance-name* command to enter the BGP-VPN instance IPv4 address family view. 3. Run the **advertise l2vpn evpn** command to configure IP prefix route advertisement. 4. Run the **quit** command to exit the BGP-VPN instance IPv4 address family view. 5. Run the **quit** command to exit the BGP or BGP multi-instance view. 6. Run the **commit** command to commit the configuration.

Step 9 Bind the VPN instance to a Layer 3 gateway, enable distributed gateway, and configure host route advertisement.

1. Run:

   ```
   interface vbdif bd-id
   ```

   A VBDIF interface is created, and the VBDIF interface view is displayed.

   By default, no VBDIF interface is created.

2. Run:

   ```
   ip binding vpn-instance vpn-instance-name
   ```

   A VPN instance is bound to the VBDIF interface.

3. Run:

   ```
   ip address ip-address { mask | mask-length } [ sub ]
   ```

   An IP address is configured for the VBDIF interface to implement Layer 3 interworking.

   By default, no IP address is configured for interfaces.

4. (Optional) Run:

   ```
   mac-address mac-address
   ```

A MAC address is configured for the VBDIF interface.

By default, the MAC address of a VBDIF interface is the system MAC address.

5. Run:

   ```
   arp distribute-gateway enable
   ```

   Distributed gateway is enabled.

   By default, distributed gateway is disabled.

   NOTE

   After distributed gateway is enabled on a Layer 3 gateway, the Layer 3 gateway discards network-side ARP messages and learns only user-side ARP messages.

6. Perform either of the following steps to configure host route advertisement:
   - If VXLAN gateways advertise IRB routes to each other, run the **arp collect host enable** command for host route advertisement.
   - If VXLAN gateways advertise IP prefix routes to each other, run the **arp direct-route enable** [**route-policy** *route-policy-name*] command for host route advertisement.

7. Run:

   ```
   quit
   ```

   Return to the system view.

8. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

10.4 (Optional) Configuring ARP Broadcast Suppression

When tenants communicate with each other for the first time, they send ARP requests. These ARP requests are broadcast on Layer 2 networks and may cause a broadcast storm. To prevent this problem, ARP broadcast suppression can be enabled on Layer 2 VXLAN gateways.

Context

After you enable ARP broadcast suppression on a Layer 2 VXLAN gateway, configure Border Gateway Protocol Ethernet Virtual Private Network (BGP EVPN) on Layer 2 and Layer 3 VXLAN gateways to allow ARP broadcast suppression to take effect. BGP EVPN can then generate host information based on learned ARP entries and advertise the host information to Layer 2 VXLAN gateways. After the Layer 2 VXLAN gateways receive ARP broadcast packets, they convert the ARP broadcast packets into unicast packets based on the learned host information before forwarding the packets out. This decreases the number of broadcast packets in a BD, improving network performance.

Configuring BGP RRs is recommended to simplify BGP configuration. Layer 3 VXLAN gateways are generally used as RRs, and Layer 2 VXLAN gateways as RR clients.

Procedure

Step 1 Configure BGP EVPN on Layer 2 and Layer 3 VXLAN gateways to advertise host information learned by Layer 2 VXLAN gateways.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   bgp as-number [ instance instance-name ]
   ```

   The BGP or BGP multi-instance view is displayed.

3. Run:

   ```
   l2vpn-family evpn
   ```

   The BGP-EVPN address family view or BGP multi-instance EVPN view is displayed.

   By default, the BGP-EVPN address family or BGP multi-instance EVPN view is disabled.

4. Configure advertisement of ARP or IRB routes to implement ARP broadcast suppression. The following two configurations cannot coexist.

   - To configure ARP route advertisement, run the **peer** { *ipv4-address* | *group-name* } **advertise arp** command.
   - To configure IRB route advertisement, run the **peer** { *ipv4-address* | *group-name* } **advertise irb** command.

5. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Enable BGP EVPN on a Layer 3 VXLAN gateway to collect host information.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   interface vbdif bd-id
   ```

   The VBDIF interface view is displayed.

3. Run:

   ```
   arp collect host enable
   ```

   BGP EVPN is enabled to collect host information.

   By default, BGP EVPN is disabled from collecting host information.

4. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 3 Enable ARP broadcast suppression on a Layer 2 VXLAN gateway.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

```
bridge-domain bd-id
```

The BD view is displayed.

3. Run:

```
arp broadcast-suppress enable
```

ARP broadcast suppression is enabled.

By default, ARP broadcast suppression is disabled.

4. Run:

```
commit
```

The configuration is committed.

----End

10.5 (Optional) Disabling a VBDIF Interface from Sending ARP Miss Messages

Context

When the device wants to communicate with another device in the same network segment, it queries ARP entries to direct packet forwarding. If the device fails to find the corresponding ARP entry from the forwarding plane, it sends an ARP Miss message to the CPU. The ARP Miss message will trigger the device to send an ARP broadcast packet to start ARP learning. In some cases, customers may want to limit the number of broadcast packets on the VXLAN network. You can then disable a VBDIF interface from sending ARP Miss messages to achieve this purpose.

After a VBDIF interface is disabled from sending ARP Miss messages, the device cannot learn ARP entries from this VBDIF interface, so ARP entries must be manually configured on it.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface vbdif bd-id
```

A VBDIF interface is created and the VBDIF interface view is displayed.

Step 3 Run:

```
arp miss disable
```

The VBDIF interface is disabled from sending ARP Miss messages.

By default, a VBDIF interface can send ARP Miss messages.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.6 (Optional) Configuring Static ARP/MAC Address Entries and MAC Address Limiting

Static ARP entries or MAC address entries can be configured for traffic forwarding, and MAC address limiting can be configured to improve VXLAN security.

Context

- Static ARP entries are manually configured and maintained. They can be neither aged nor overwritten by dynamic ARP entries. Therefore, configuring static ARP entries on Layer 3 VXLAN gateways enhances communication security. If a static ARP entry is configured on a device, the device can communicate with a peer device that has a specified IP address using only the specified MAC address. Network attackers cannot modify the mapping between the IP and MAC addresses, which ensures communication between the two devices.
- After the source NVE on a VXLAN tunnel receives broadcast, unknown unicast, and multicast (BUM) packets, the local VTEP sends a copy of the BUM packets to every VTEP in the ingress replication list. Configuring static MAC address entries helps reduce broadcast traffic and prevent unauthorized data access from bogus users.
- The maximum number of MAC addresses that a device can learn can be configured to limit the number of access users and prevent against attacks on MAC address tables. If the device has learned the maximum number of MAC addresses allowed, no more addresses can be learned. The device can also be configured to discard packets after learning the maximum allowed number of MAC addresses, improving network security.
- If Layer 3 VXLAN gateway does not need to learn MAC addresses of packets in a BD, MAC address learning can be disabled from the BD to conserve MAC address entry resources. If the network topology of a VXLAN becomes stable and MAC address entry learning is complete, MAC address learning can also be disabled.

Configuring static MAC address entries and MAC address limiting applies to Layer 2 VXLAN gateways; configuring static ARP entries applies to Layer 3 VXLAN gateways; disabling MAC address limiting applies to both Layer 2 and Layer 3 VXLAN gateways.

Procedure

- Configure a static ARP entry.
  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  arp static ip-address mac-address vni vni-id source-ip source-ip peer-ip
  peer-ip
  ```

  A static ARP entry is configured.

  By default, no static ARP entry is configured.

  NOTE

  ip-address must belong to the same network segment as the Layer 3 gateway's IP address.

  c. Run:

  ```
  commit
  ```

The configuration is committed.

- Configure a static MAC address entry.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  mac-address static mac-address bridge-domain bd-id source source-ip-address peer peer-ip vni vni-id
  ```

  A static MAC address entry is configured.

  By default, no static MAC address entry is configured.

  c. Run:

  ```
  commit
  ```

  The configuration is committed.

- Configure MAC address limiting.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  bridge-domain bd-id
  ```

  The BD view is displayed.

  c. Run:

  ```
  mac-address limit { action { discard | forward } | maximum max | alarm { disable | enable } } *
  ```

  MAC address limiting is configured.

  By default, MAC address limiting is not configured.

  d. Run:

  ```
  commit
  ```

  The configuration is committed.

- Disable MAC address learning.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  bridge-domain bd-id
  ```

  The BD view is displayed.

  c. Run:

  ```
  mac-address learning disable
  ```

  MAC address learning is disabled.

  By default, MAC address learning is enabled for a BD.

  d. Run:

  ```
  commit
  ```

  The configuration is committed.

----End

10.7 (Optional) Configuring a VXLAN BD Whitelist for MAC Address Flapping Detection

In specific VXLAN applications, when a device connects to a load balancing server equipped with two network interface cards, the server's MAC address may be learned by two interfaces on the device. This is a normal situation where MAC address flapping detection is not needed. In this case, configure a VXLAN BD whitelist for MAC address flapping detection.

Context

By default, MAC address flapping detection is enabled globally. After a BD is added to a MAC address flapping detection whitelist, detection is not performed for this BD. Even if MAC address flapping occurs in the BD, the occurrence generates neither an alarm nor a record.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
mac-address flapping detection exclude bridge-domain bd-id1 [ to bd-id2 ]
```

A VXLAN BD whitelist for MAC address flapping detection is configured.

By default, no VXLAN BD whitelist for MAC address flapping detection is configured.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

10.8 (Optional) Configuring IP Address Conflict Detection Parameters

On VXLANs, IP address conflicts of terminal users will prevent these users from going online. Therefore, IP address conflicts must be detected.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
arp broadcast-suppress enable
```

ARP broadcast suppression is enabled.

Step 3 Run:

```
host ip-conflict-check period period-value retry-times times-value
```

An interval at which IP address conflicts of terminal users are detected and the IP address conflict threshold are configured.

By default, IP address conflicts of terminal users are detected at an interval of 180s, and the IP address conflict threshold is 5. If the number of detected IP address conflicts exceeds the configured threshold within the configured detection interval, the device generates an alarm.

Step 4 Run:

```
commit
```

The configuration is committed.

----End

10.9 (Optional) Optimizing ACL Resource Usage on the VXLAN

Context

If you configure other ACL resource-consuming services, such as MQC, simplified ACL, traffic policing, BD traffic statistics collection, and DHCP on a switch deployed with VXLAN services, there is high probability that the other services fail to be configured. In this case, use either of the following methods to reduce the probability of service deployment failures.

- Enabling the VXLAN service extension function:

  This function takes effect only on VXLAN-enabled cards except the CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA series.

  NOTE

  After this function is enabled, only CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA can forward packets with 230 to 294 bytes received from the VXLAN tunnels to cards of the same series.

- Disabling an Eth-Trunk from load balancing VXLAN packets in optimized mode:
  - You are advised to disable load balancing of VXLAN packets in optimized mode when neither the user side nor the network side of a VXLAN tunnel is an Eth-Trunk.
  - You can disable load balancing of VXLAN packets in optimized mode when either the user side or the network side of a VXLAN tunnel is an Eth-Trunk. After this function is disabled, an Eth-Trunk does not evenly load balance VXLAN packets or does not perform load balancing.

  NOTE

  This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.

Procedure

Step 1 Enable the VXLAN service extension function.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 service extend enable
   ```

   The VXLAN service extension function is enabled.

   By default, the VXLAN service extension function is disabled.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Disable an Eth-Trunk from load balancing VXLAN packets in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 eth-trunk hash disable
   ```

   An Eth-Trunk is disabled from load balancing VXLAN packets in optimized mode.

   By default, an Eth-Trunk load balances VXLAN packets in optimized mode.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

10.10 (Optional) Optimizing Load Balancing on the VXLAN Network

Context

On a VXLAN network, VXLAN packets can be load balanced through ECMP or Eth-Trunks. To enable load balancing or improve the load balancing effect, enable either of the following functions:

- Enable load balancing of VXLAN packets through ECMP in optimized mode.
- Enable an Eth-Trunk to load balance VXLAN packets in optimized mode.

NOTE

This function is not supported if the card works in non-enhanced mode. Run the **display forward capability** command to check the working mode of the card.

Procedure

Step 1 Enable load balancing of VXLAN packets through ECMP in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 ecmp hash enable
   ```

   Load balancing of VXLAN packets through ECMP in optimized mode is enabled.

   By default, load balancing of VXLAN packets through ECMP in optimized mode is disabled.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

Step 2 Enable an Eth-Trunk to load balance VXLAN packets in optimized mode.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   assign forward nvo3 eth-trunk hash enable
   ```

   An Eth-Trunk is enabled to load balance VXLAN packets in optimized mode.

   By default, an Eth-Trunk is disabled from load balancing VXLAN packets in optimized mode.

3. Run:

   ```
   commit
   ```

   The configuration is committed.

----End

10.11 (Optional) Configuring the Layer 3 VXLAN Gateway to Work in Non-loopback Mode

Context

By default, a Layer 3 VXLAN gateway works in loopback mode. That is, the Layer 3 VXLAN gateway first loops back VXLAN-encapsulated packets, and then forwards the packets. In this situation, traffic transmitted over the VXLAN tunnel cannot be forwarded at line speed when the traffic volume is heavy.

To solve the preceding problem, configure the Layer 3 VXLAN gateway to work in non-loopback mode.

- Layer 2 non-loopback mode: The device determines whether to loop back the received packets according to the destination MAC address in the inner Ethernet header of VXLAN packets where the received packets are encapsulated. If the destination MAC address is not the MAC address of the VXLAN Layer 3 gateway, the device directly forwards the packets at line speed.
- Layer 3 non-loopback mode: The device determines whether to loop back the received packets according to the destination MAC address in the inner Ethernet header of VXLAN packets where the received packets are encapsulated. If the destination MAC address is the MAC address of the VXLAN Layer 3 gateway, the device directly forwards the packets at line speed.

NOTE

- GE interface cards do not support the Layer 2 non-loopback mode. Among non-GE interface cards, EA series cards support only the Layer 2 non-loopback mode, and other series cards support Layer 2 and Layer 3 non-loopback modes.
- Traffic will be interrupted when you change the enhanced mode of a Layer 3 VXLAN gateway. Therefore, you are advised to change the enhanced mode of a Layer 3 VXLAN gateway when there is no or little traffic on the network.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
assign forward nvo3-gateway enhanced { l2 | l3 }
```

The Layer 3 VXLAN gateway is configured to work in non-loopback mode.

By default, the enhanced mode of the Layer 3 VXLAN gateway is not set and the switch works in loopback mode.

NOTE

When you set the enhanced mode of the VXLAN gateway to Layer 3 non-loopback, you also need to run the **assign forward nvo3 service extend enable** command to enable the VXLAN service extension function. In this case, ensure that the switch does not have CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA cards installed, or the CE-L48GT-EA, CE-L48GT-EC, CE-L48GS-EA, CE-L48GS-EC, CE-L24XS-BA, CE-L24XS-EA, CE-L48XS-BA, CE-L48XS-EA and CE-L24LQ-EA cards do not transmit the VXLAN service.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

10.12 Checking the Configurations

After configuring VXLAN in distributed gateway mode using BGP EVPN, check the configurations, and you can find that VXLAN tunnels are dynamically established and are in the Up state.

Prerequisites

VXLAN in distributed gateway mode has been configured using BGP EVPN.

Procedure

- Run the **display bridge-domain** [*bd-id* [**brief** | **verbose**]] command to check BD configurations.
- Run the **display interface nve** [*nve-number* | **main**] command to check NVE interface information.

- Run the **display evpn vpn-instance** [*vpn-instance-name*] command to check EVPN instance information.
- Run the **display bgp** [**instance** *instance-name*] **evpn peer** [[*ipv4-address*] **verbose**] command to check BGP EVPN peer information.
- Run the **display vxlan peer** [**vni** *vni-id*] command to check ingress replication lists of a VNI or all VNIs.
- Run the **display vxlan tunnel** [*tunnel-id*] [**verbose**] command to check VXLAN tunnel information.
- Run the **display vxlan vni** [*vni-id* [**verbose**]] command to check VNI information.
- Run the **display interface vbdif** [*bd-id*] command to check VBDIF interface information and statistics.
- Run the **display dfs-group** *dfs-group-id* **active-active-gateway** command to check information of all-active gateways in a DFS group.
- Run the **display arp broadcast-suppress user bridge-domain** *bd-id* command to check the ARP broadcast suppression table of a BD.
- Run the **display arp** [**network** *network-address* [*network-mask* | *mask-length*]] **static** command to check static ARP entries.
- Run the **display mac-address static bridge-domain** *bd-id* command to check static MAC address entries in a BD.
- Run the **display mac-address limit bridge-domain** *bd-id* command to check MAC address limiting configurations of a BD.
- Run the **display mac-address flapping** command to check the MAC address flapping detection configuration.
- Run the **display bgp** [**instance** *instance-name*] **evpn all routing-table** command to check EVPN route information.
- Run the **display mac-address inactive** [**evn**] [**slot** *slot-id*] command to check the MAC address entries that fail to be delivered.
- Run the **display mac-address total-number evn** [**vlan** *vlan-id*] command to check the number of EVN MAC address entries.
- Run the **display mac-address evn** [**vlan** *vlan-id*] command to check EVN MAC address entries.

----End

11 Maintaining VXLAN

About This Chapter

During VXLAN maintenance, traffic statistics and MAC address entries in a BD can be viewed in real time to monitor the VXLAN operating status. These statistics can be cleared when required.

11.1 Configuring the VXLAN Alarm Function
To learn about the VXLAN operating status in time, configure the VXLAN alarm function so that the NMS will be notified of the VXLAN status changes. This facilitates O&M.

11.2 Collecting and Checking VXLAN Packet Statistics
To check the network status or locate network faults, you can enable the BD or VXLAN-based traffic statistics function to view VXLAN packet statistics.

11.3 Clearing VXLAN Packet Statistics
Before you collect VXLAN packet statistics within a certain period, clear the existing statistics on the device to ensure statistics accuracy.

11.4 Checking Statistics about MAC Address Entries in a BD
Statistics about MAC address entries in a BD can be viewed to monitor the VXLAN operating status.

11.5 Clearing Statistics about Dynamic MAC Address Entries in a BD
To view dynamic MAC address entries in a BD within a specified period of time, clear existing dynamic MAC address entry information before starting statistics collection to ensure information accuracy.

11.6 Configuring BGP EVPN Soft Reset
BGP EVPN soft reset allows a device to receive EVPN routes from BGP EVPN peers again.

11.7 Resetting BGP EVPN Connections
Resetting a BGP EVPN connection will interrupt peer relationships.

11.8 Monitoring the VXLAN Operating Status
During VXLAN operating status monitoring, you can check causes for fault locating if a VXLAN tunnel goes Down or fails to be dynamically created.

11.1 Configuring the VXLAN Alarm Function

To learn about the VXLAN operating status in time, configure the VXLAN alarm function so that the NMS will be notified of the VXLAN status changes. This facilitates O&M.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
snmp-agent trap enable feature-name nvo3 [ trap-name { hwnvo3vxlantnldown |
hwnvo3vxlantnlup } ]
```

The VXLAN alarm function is enabled.

By default, the VXLAN alarm function is disabled.

Step 3 Run:

```
commit
```

The configuration is committed.

----End

Checking the Configurations

After the VXLAN alarm function is enabled, check the VXLAN alarm status.

Run the **display snmp-agent trap feature-name nvo3 all** command to check configurations of all alarm functions of the VXLAN module.

11.2 Collecting and Checking VXLAN Packet Statistics

To check the network status or locate network faults, you can enable the BD or VXLAN-based traffic statistics function to view VXLAN packet statistics.

Procedure

- Enable BD-based packet statistics collection.

  a. Run:

  ```
  system-view
  ```

  The system view is displayed.

  b. Run:

  ```
  bridge-domain bd-id
  ```

  A BD is created, and the BD view is displayed.

  By default, no BD is created.

  c. Run:

  ```
  statistics enable
  ```

  Traffic statistics collection is enabled for the BD.

By default, traffic statistics collection is disabled in BDs.

d. Run:

```
commit
```

The configuration is committed.

- Enable VXLAN-based packet statistics collection.

a. Run:

```
system-view
```

The system view is displayed.

b. Run:

```
interface nve nve-number
```

An NVE interface is created, and the NVE interface view is displayed.

By default, no NVE interface is created.

c. Run:

```
vxlan statistics peer peer-ip-address [ vni vni-id ] enable
```

Traffic statistics collection is enabled for the VXLAN tunnel.

By default, statistics collection of VXLAN tunnel packets is disabled.

NOTE

The functions of collecting packet statistics based on the VXLAN tunnel and VNI and based on the VXLAN tunnel only are mutually exclusive. For example, the **vxlan statistics peer 10.1.1.1 vni 10000 enable** and **vxlan statistics peer 10.1.1.1 enable** commands cannot be configured simultaneously.

d. Run:

```
commit
```

The configuration is committed.

Postrequisite

- Run the **display bridge-domain** *bd-id* **statistics** command to check traffic statistics of a BD.
- Run the **display vxlan statistics source** *source-ip-address* **peer** *peer-ip-address* [**vni** *vni-id*] command to view VXLAN tunnel packet statistics.

11.3 Clearing VXLAN Packet Statistics

Before you collect VXLAN packet statistics within a certain period, clear the existing statistics on the device to ensure statistics accuracy.

Context

NOTE

The VXLAN packet statistics cannot be restored after being cleared. Confirm your operation before clearing the VXLAN packet statistics.

Procedure

- Run the **reset bridge-domain** *bd-id* **statistics** command in the user view to clear packet statistics of a BD.

- Run the **reset vxlan statistics source** *source-ip-address* **peer** *peer-ip-address* [**vni** *vni-id*] command in the user view to clear VXLAN tunnel packet statistics.

----End

11.4 Checking Statistics about MAC Address Entries in a BD

Statistics about MAC address entries in a BD can be viewed to monitor the VXLAN operating status.

Context

In routine maintenance, run the following commands in any view to check the VXLAN operating status.

Procedure

- Run the **display mac-address** [*mac-address*] **bridge-domain** *bd-id* command to check statistics about all MAC address entries in a BD.
- Run the **display mac-address total-number** [**static**] **bridge-domain** *bd-id* command to check the number of MAC address entries in a BD.

----End

11.5 Clearing Statistics about Dynamic MAC Address Entries in a BD

To view dynamic MAC address entries in a BD within a specified period of time, clear existing dynamic MAC address entry information before starting statistics collection to ensure information accuracy.

Context

NOTE

Statistics about dynamic MAC address entries in a BD cannot be restored after they are cleared. Exercise caution when running the reset command.

Procedure

- Run the **reset mac-address bridge-domain** *bd-id* command in the user view to clear statistics about dynamic MAC address entries in a BD.

----End

11.6 Configuring BGP EVPN Soft Reset

BGP EVPN soft reset allows a device to receive EVPN routes from BGP EVPN peers again.

Usage Scenario

BGP EVPN soft reset performs a soft reset on BGP EVPN connections, which triggers BGP EVPN peers to send EVPN routes to a local device without tearing down the BGP EVPN connections and refresh the BGP EVPN routing table.

Procedure

- In the user view, run:

  ```
  refresh bgp evpn { all | peer-address | group group-name } { export | import }
  ```

  BGP EVPN soft reset is configured.

----End

11.7 Resetting BGP EVPN Connections

Resetting a BGP EVPN connection will interrupt peer relationships.

Context

NOTICE

The BGP EVPN peer relationship between switchs is interrupted after you reset BGP EVPN connections with the **reset bgp** command. Therefore, exercise caution when running the command.

To reset BGP EVPN connections, run the following reset commands in the user view:

Procedure

- To reset all BGP EVPN connections, run the **reset bgp** [**instance** *instance-name*] **evpn all** command.
- To reset BGP EVPN connections with a specified AS, run the **reset bgp**[**instance** *instance-name*] **evpn** *as-number* command.
- To reset BGP EVPN connections with a specified peer, run the **reset bgp** [**instance** *instance-name*] **evpn** *ipv4-address* command.
- To reset BGP EVPN connections with a specified peer group, run the **reset bgp** [**instance** *instance-name*] **evpn group** *group-name* command.

----End

11.8 Monitoring the VXLAN Operating Status

During VXLAN operating status monitoring, you can check causes for fault locating if a VXLAN tunnel goes Down or fails to be dynamically created.

Procedure

Step 1 Run the **display vxlan troubleshooting** command to check causes for the VXLAN tunnel Down events and dynamic VXLAN tunnel establishment failures.

This command can display causes for the recent five VXLAN tunnel Down events and dynamic VXLAN tunnel establishment failures at most.

----End

12 Configuration Examples (Single-Node Mode)

About This Chapter

This section provides several configuration examples of VXLAN. In each configuration example, the networking requirements, configuration roadmap, configuration procedures, and configuration files are provided.

This section only provides configuration examples for each feature. For details about multi-feature configuration cases, feature-specific configuration cases, interconnection cases, protocol or hardware replacement cases, and industry application cases, see the *Typical Configuration Cases*.

12.1 Example for Configuring VXLAN in Centralized Gateway Mode for Static Tunnel Establishment
This section provides an example for configuring VXLAN in centralized gateway mode for static tunnel establishment.

12.2 Example for Configuring VXLAN in Centralized Gateway Mode Using BGP EVPN
This section provides an example for configuring VXLAN in centralized gateway mode for dynamic tunnel establishment so that users on the same network segment or different network segments can communicate.

12.3 Example for Configuring VXLAN in Distributed Gateway Mode Using MP-BGP
This section provides an example for configuring VXLAN in distributed gateway mode using MP-BGP.

12.4 Example for Configuring VXLAN in Distributed Gateway Mode Using BGP EVPN
This section provides an example for configuring VXLAN in distributed gateway mode using BGP EVPN.

12.5 Example for Configuring All-Active VXLAN Gateways

12.6 Example for Configuring Dual-Active VXLAN Access

12.1 Example for Configuring VXLAN in Centralized Gateway Mode for Static Tunnel Establishment

This section provides an example for configuring VXLAN in centralized gateway mode for static tunnel establishment.

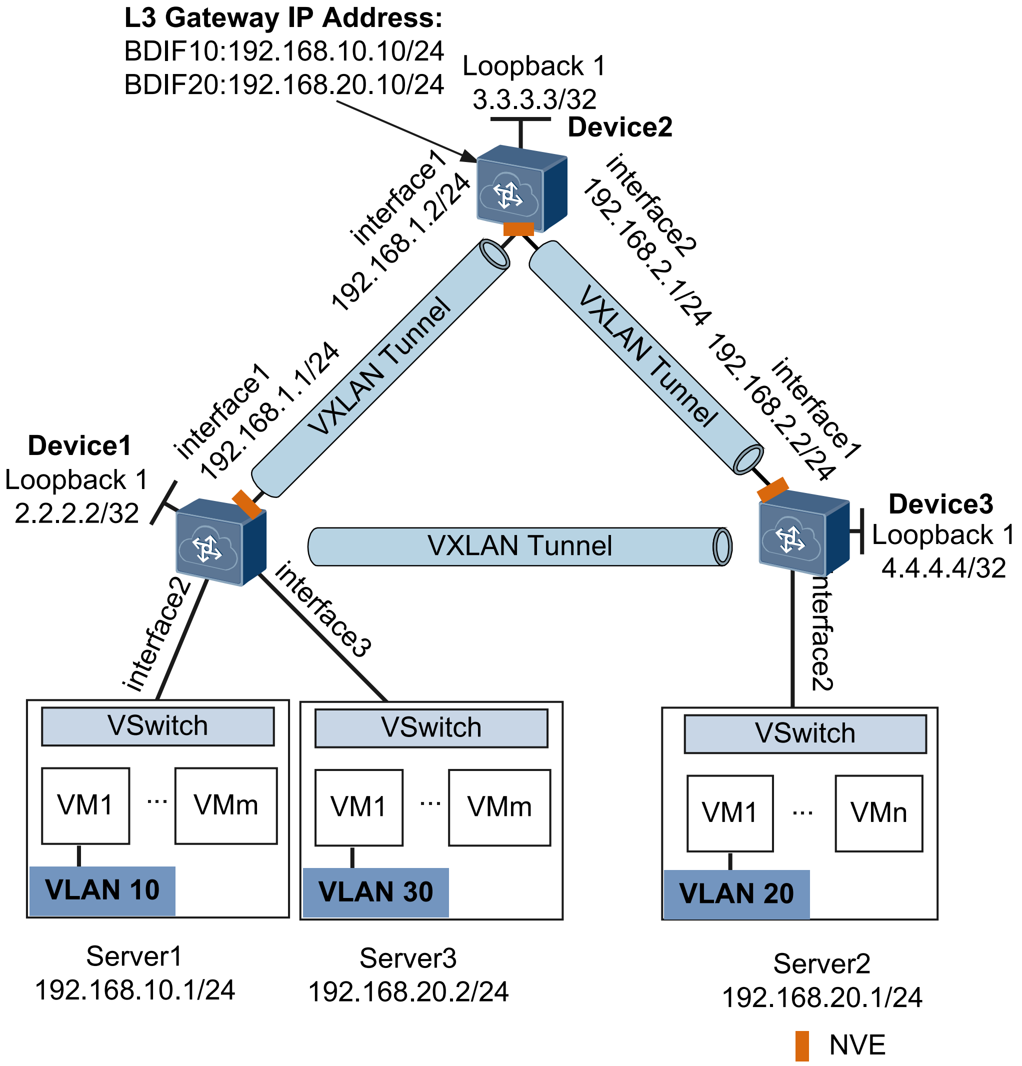
Networking Requirements

On the network shown in **Figure 12-1**, an enterprise has VMs deployed in different data centers. VM 1 on Server 1 belongs to VLAN 10, VM 1 on Server 2 belongs to VLAN 20, and VM 1 on Server 3 belongs to VLAN 30. Server 1 and Server 2 reside on different network segments, whereas Server 2 and Server 3 reside on the same network segment. To allow VM1s in different data centers to communicate with each other, configure a centralized VXLAN gateway.

Figure 12-1 VXLAN in centralized gateway mode for static tunnel establishment

NOTE

Interface 1, Interface 2, and Interface 3 represent 10GE 1/0/1, 10GE 1/0/2, and 10GE 1/0/3, respectively.



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on Device 1, Device 2, and Device 3 to allow them to communicate at Layer 3.
2. Configure a service access point on Device 1 and Device 3 to differentiate service traffic.
3. Configure a VXLAN tunnel on Device 1, Device 2, and Device 3 to forward service traffic.
4. Configure Device 2 as a Layer 3 VXLAN gateway to allow users on different network segments to communicate.

Data Preparation

To complete the configuration, you need the following data.

- VMs' VLAN IDs (10, 20, and 30)
- IP addresses of interfaces connecting devices
- Interior Gateway Protocol (IGP) running between devices (OSPF in this example)
- BD IDs (10 and 20)
- VNI IDs (5010 and 5020)

Procedure

Step 1 Configure a routing protocol.

# Configure Device 1. Repeat this step for Device 2 and Device 3. Configure the devices to advertise the 32-bit IP addresses of loopback interfaces.

```
<HUAWEI> system-view
[~HUAWEI] sysname Device1
[*HUAWEI] commit
[~Device1] interface loopback 1
[*Device1-LoopBack1] ip address 2.2.2.2 32
[*Device1-LoopBack1] quit
[*Device1] interface 10ge 1/0/1
[*Device1-10GE1/0/1] undo portswitch
[*Device1-10GE1/0/1] ip address 192.168.1.1 24
[*Device1-10GE1/0/1] quit
[*Device1] ospf
[*Device1-ospf-1] area 0
[*Device1-ospf-1-area-0.0.0.0] network 2.2.2.2 0.0.0.0
[*Device1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[*Device1-ospf-1-area-0.0.0.0] quit
[*Device1-ospf-1] quit
[*Device1] commit
```

# After OSPF is configured, the devices can use OSPF to learn the IP addresses of each other's loopback interfaces and successfully ping each other. The following example shows the command output on Device 1 after it pings Device 3:

```
[~Device1] ping 4.4.4.4
  PING 4.4.4.4: 56  data bytes, press CTRL_C to break
    Reply from 4.4.4.4: bytes=56 Sequence=1 ttl=254 time=5 ms
    Reply from 4.4.4.4: bytes=56 Sequence=2 ttl=254 time=2 ms
    Reply from 4.4.4.4: bytes=56 Sequence=3 ttl=254 time=2 ms
    Reply from 4.4.4.4: bytes=56 Sequence=4 ttl=254 time=3 ms
    Reply from 4.4.4.4: bytes=56 Sequence=5 ttl=254 time=3 ms
```

```
 --- 4.4.4.4 ping statistics ---
   5 packet(s) transmitted
   5 packet(s) received
   0.00% packet loss
   round-trip min/avg/max = 2/3/5 ms
```

Step 2 Configure the VXLAN tunnel mode and enable the VXLAN ACL extension function.

# Configure Device1. The configurations on Device2 and Device3 are similar to that on Device1, and are not mentioned here.

```
[~Device1] ip tunnel mode vxlan
[*Device1] assign forward nvo3 acl extend enable
[*Device1] commit
```

NOTE

After modifying the VXLAN tunnel mode or enabling the VXLAN ACL extension function, you need to save the configuration and restart the device to make the configuration take effect. You can restart the device immediately or after completing all the configurations.

Step 3 Configure a service access point on Device 1 and Device 3.

# Configure Device 1. Repeat this step for Device 3.

```
[~Device1] bridge-domain 10
[*Device1-bd10] quit
[*Device1] interface 10ge 1/0/2.1 mode l2
[*Device1-10GE1/0/2.1] encapsulation dot1q vid 10
[*Device1-10GE1/0/2.1] bridge-domain 10
[*Device1-10GE1/0/2.1] quit
[*Device1] bridge-domain 20
[*Device1-bd20] quit
[*Device1] interface 10ge 1/0/3.1 mode l2
[*Device1-10GE1/0/3.1] encapsulation dot1q vid 30
[*Device1-10GE1/0/3.1] bridge-domain 20
[*Device1-10GE1/0/3.1] quit
[*Device1] commit
```

Step 4 Configure a VXLAN tunnel on Device 1, Device 2, and Device 3.

# Configure Device 1.

```
[~Device1] bridge-domain 10
[~Device1-bd10] vxlan vni 5010
[*Device1-bd10] quit
[*Device1] interface nve 1
[*Device1-Nve1] source 2.2.2.2
[*Device1-Nve1] vni 5010 head-end peer-list 3.3.3.3
[*Device1-Nve1] quit
[*Device1] bridge-domain 20
[*Device1-bd20] vxlan vni 5020
[*Device1-bd20] quit
[*Device1] interface nve 1
[*Device1-Nve1] vni 5020 head-end peer-list 4.4.4.4
[*Device1-Nve1] quit
[*Device1] commit
```

# Configure Device 2.

```
[~Device2] bridge-domain 10
[*Device2-bd10] vxlan vni 5010
[*Device2-bd10] quit
[*Device2] interface nve 1
[*Device2-Nve1] source 3.3.3.3
[*Device2-Nve1] vni 5010 head-end peer-list 2.2.2.2
[*Device2-Nve1] quit
[*Device2] bridge-domain 20
[*Device2-bd20] vxlan vni 5020
```

```
[*Device2-bd20] quit
[*Device2] interface nve 1
[*Device2-Nve1] source 3.3.3.3
[*Device2-Nve1] vni 5020 head-end peer-list 4.4.4.4
[*Device2-Nve1] quit
[*Device2] commit
```

# Configure Device 3.

```
[~Device3] bridge-domain 20
[~Device3-bd20] vxlan vni 5020
[*Device3-bd20] quit
[*Device3] interface nve 1
[*Device3-Nve1] source 4.4.4.4
[*Device3-Nve1] vni 5020 head-end peer-list 3.3.3.3
[*Device3-Nve1] vni 5020 head-end peer-list 2.2.2.2
[*Device3-Nve1] quit
[*Device3] commit
```

Step 5 Configure Device 2 as a Layer 3 VXLAN gateway.

```
[~Device2] interface vbdif 10
[*Device2-Vbdif10] ip address 192.168.10.10 24
[*Device2-Vbdif10] quit
[*Device2] interface vbdif 20
[*Device2-Vbdif20] ip address 192.168.20.10 24
[*Device2-Vbdif20] quit
[*Device2] commit
```

Step 6 Verify the configuration.

After completing the configurations, run the **display vxlan tunnel** and **display vxlan vni** commands on Device 1, Device 2, and Device 3 to check the VXLAN tunnel and VNI information, respectively. The VNIs are Up on Device 1, Device 2, and Device 3. The following example shows the command output on Device 2:

```
[~Device2] display vxlan tunnel
Number of vxlan tunnel : 2
Tunnel ID   Source           Destination        State  Type
--------------------------------------------------------------
40265318411  3.3.3.3          2.2.2.2            up     static
40265318412  3.3.3.3          4.4.4.4            up     static
[~Device2] display vxlan vni
Number of vxlan vni : 2
VNI            BD-ID            State
---------------------------------------
5010           10               up
5020           20               up
```

VM1s on different servers can communicate.

----End

Configuration Files

- Device 1 configuration file

```
#
sysname Device1
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
bridge-domain 10
 vxlan vni 5010
#
bridge-domain 20
 vxlan vni 5020
```

```
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.1.1 255.255.255.0
#
interface 10GE1/0/2.1 mode l2
 encapsulation dot1q vid 10
 bridge-domain 10
#
interface 10GE1/0/3.1 mode l2
 encapsulation dot1q vid 30
 bridge-domain 20
#
interface LoopBack1
 ip address 2.2.2.2 255.255.255.255
#
interface Nve1
 source 2.2.2.2
 vni 5010 head-end peer-list 3.3.3.3
 vni 5020 head-end peer-list 4.4.4.4
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Device 2 configuration file

```
#
sysname Device2
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
bridge-domain 10
 vxlan vni 5010
#
bridge-domain 20
 vxlan vni 5020
#
interface Vbdif10
 ip address 192.168.10.10 255.255.255.0
#
interface Vbdif20
 ip address 192.168.20.10 255.255.255.0
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 192.168.2.1 255.255.255.0
#
interface LoopBack1
 ip address 3.3.3.3 255.255.255.255
#
interface Nve1
 source 3.3.3.3
 vni 5010 head-end peer-list 2.2.2.2
 vni 5020 head-end peer-list 4.4.4.4
#
ospf 1
 area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 192.168.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
```

```
#
return
```

- Device 3 configuration file

```
#
sysname Device3
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
bridge-domain 20
 vxlan vni 5020
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.2.2 255.255.255.0
#
interface 10GE1/0/2.1 mode l2
 encapsulation dot1q vid 20
 bridge-domain 20
#
interface LoopBack1
 ip address 4.4.4.4 255.255.255.255
#
interface Nve1
 source 4.4.4.4
 vni 5020 head-end peer-list 2.2.2.2
 vni 5020 head-end peer-list 3.3.3.3
#
ospf 1
 area 0.0.0.0
  network 4.4.4.4 0.0.0.0
  network 192.168.2.0 0.0.0.255
#
return
```

12.2 Example for Configuring VXLAN in Centralized Gateway Mode Using BGP EVPN

This section provides an example for configuring VXLAN in centralized gateway mode for dynamic tunnel establishment so that users on the same network segment or different network segments can communicate.

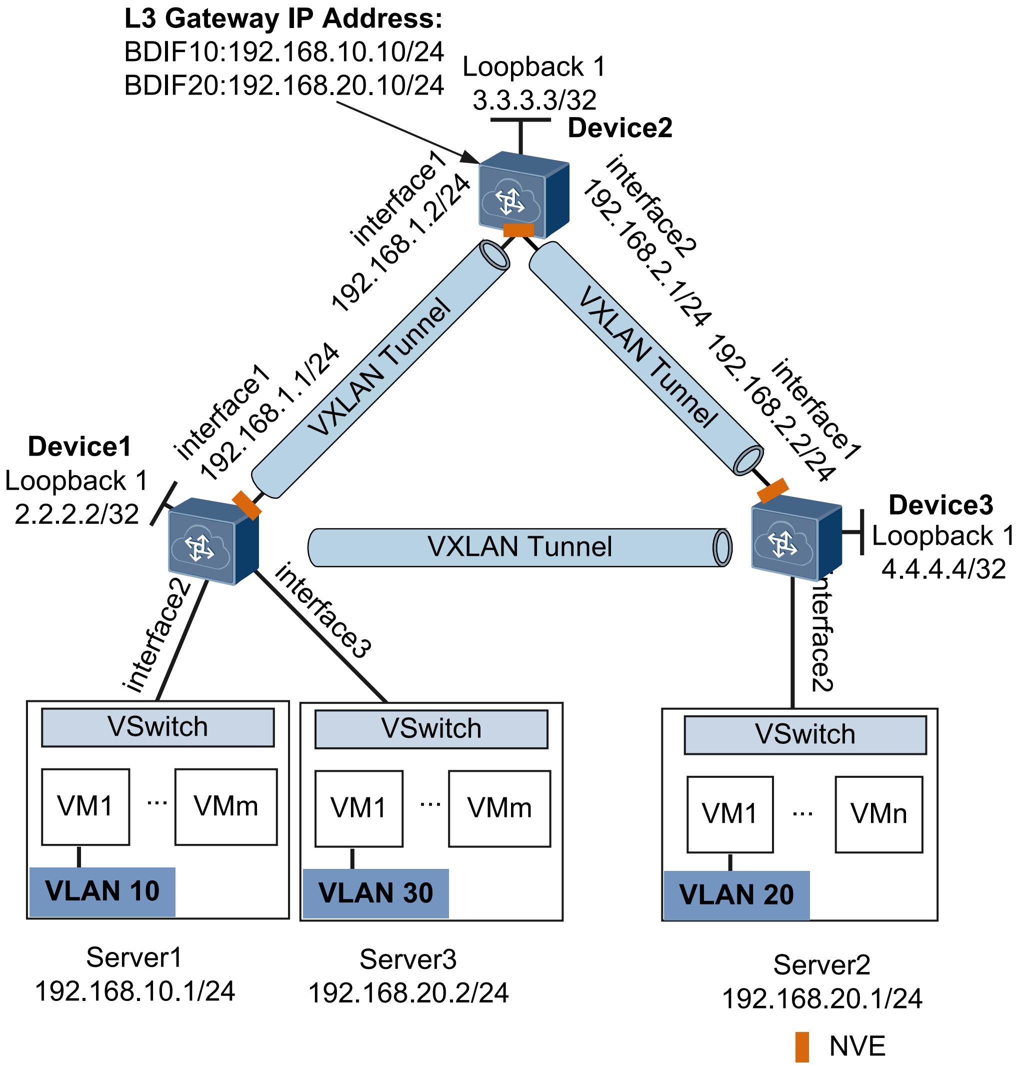
Networking Requirements

On the network shown in **Figure 12-2**, an enterprise has VMs deployed in different data centers. VM 1 on Server 1 belongs to VLAN 10, VM 1 on Server 2 belongs to VLAN 20, and VM 1 on Server 3 belongs to VLAN 30. Server 1 and Server 2 reside on different network segments, whereas Server 2 and Server 3 reside on the same network segment. To allow VM1s in different data centers to communicate with each other, configure a Layer 3 VXLAN gateway.

Figure 12-2 VXLAN in centralized gateway mode for dynamic tunnel establishment

NOTE

Interface 1, Interface 2, and Interface 3 represent 10GE 1/0/1, 10GE 1/0/2, and 10GE 1/0/3, respectively.



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on Device 1, Device 2, and Device 3 to allow them to communicate at Layer 3.
2. Configure a service access point on Device 1 and Device 3 to differentiate service traffic.
3. Configure EVPN as the VXLAN control plane.
4. Configure an BGP EVPN peer relationship.
5. Configure EVPN instances.
6. Configure an ingress replication list.
7. Configure Device 2 as a Layer 3 VXLAN gateway.

Data Preparation

To complete the configuration, you need the following data.

- VMs' VLAN IDs (10, 20, and 30)

- IP addresses of interfaces connecting devices
- Interior Gateway Protocol (IGP) running between devices (OSPF in this example)
- BD IDs (10 and 20)
- VNI IDs (5010 and 5020)
- EVPN instances' RDs (11:1, 12:1, 21:1, 23:1, and 31:2) and RTs (1:1 and 2:2)

Procedure

Step 1 Configure a routing protocol.

# Configure Device 1. Configure the devices to advertise the 32-bit IP addresses of loopback interfaces.

```
<HUAWEI> system-view
[~HUAWEI] sysname Device1
[*HUAWEI] commit
[~Device1] interface loopback 1
[*Device1-LoopBack1] ip address 2.2.2.2 32
[*Device1-LoopBack1] quit
[*Device1] interface 10ge 1/0/1
[*Device1-10GE1/0/1] undo portswitch
[*Device1-10GE1/0/1] ip address 192.168.1.1 24
[*Device1-10GE1/0/1] quit
[*Device1] ospf
[*Device1-ospf-1] area 0
[*Device1-ospf-1-area-0.0.0.0] network 2.2.2.2 0.0.0.0
[*Device1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[*Device1-ospf-1-area-0.0.0.0] quit
[*Device1-ospf-1] quit
[*Device1] commit
```

# After OSPF is configured, the devices can use OSPF to learn the IP addresses of each other's loopback interfaces and successfully ping each other. The following example shows the command output on Device 1 after it pings Device 3:

```
[~Device1] ping 4.4.4.4
  PING 4.4.4.4: 56  data bytes, press CTRL_C to break
    Reply from 4.4.4.4: bytes=56 Sequence=1 ttl=254 time=5 ms
    Reply from 4.4.4.4: bytes=56 Sequence=2 ttl=254 time=2 ms
    Reply from 4.4.4.4: bytes=56 Sequence=3 ttl=254 time=2 ms
    Reply from 4.4.4.4: bytes=56 Sequence=4 ttl=254 time=3 ms
    Reply from 4.4.4.4: bytes=56 Sequence=5 ttl=254 time=3 ms

  --- 4.4.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 2/3/5 ms
```

Step 2 Configure the VXLAN tunnel mode and enable the VXLAN ACL extension function.

# Configure Device1. The configurations on Device2 and Device3 are similar to that on Device1, and are not mentioned here.

```
[~Device1] ip tunnel mode vxlan
[*Device1] assign forward nvo3 acl extend enable
[*Device1] commit
```

NOTE

After modifying the VXLAN tunnel mode or enabling the VXLAN ACL extension function, you need to save the configuration and restart the device to make the configuration take effect. You can restart the device immediately or after completing all the configurations.

Step 3 Configure a service access point on Device 1 and Device 3.

# Configure Device 1. Repeat this step for Device 3.

```
[~Device1] bridge-domain 10
[*Device1-bd10] quit
[*Device1] interface 10ge 1/0/2.1 mode l2
[*Device1-10GE1/0/2.1] encapsulation dot1q vid 10
[*Device1-10GE1/0/2.1] bridge-domain 10
[*Device1-10GE1/0/2.1] quit
[*Device1] bridge-domain 20
[*Device1-bd20] quit
[*Device1] interface 10ge 1/0/3.1 mode l2
[*Device1-10GE1/0/3.1] encapsulation dot1q vid 30
[*Device1-10GE1/0/3.1] bridge-domain 20
[*Device1-10GE1/0/3.1] quit
[*Device1] commit
```

Step 4 Configure EVPN as the VXLAN control plane on Device 1, Device 2, and Device 3.

# Configure Device 1. Repeat this step for Device 2 and Device 3.

```
[~Device1] evpn-overlay enable
[*Device1] commit
```

Step 5 Configure an BGP EVPN peer relationship.

# Configure Device 1. Repeat this step for Device 2 and Device 3.

```
[~Device1] bgp 100 instance evpn1
[*Device1-bgp-instance-evpn1] peer 3.3.3.3 as-number 100
[*Device1-bgp-instance-evpn1] peer 3.3.3.3 connect-interface LoopBack1
[*Device1-bgp-instance-evpn1] peer 4.4.4.4 as-number 100
[*Device1-bgp-instance-evpn1] peer 4.4.4.4 connect-interface LoopBack1
[*Device1-bgp-instance-evpn1] l2vpn-family evpn
[*Device1-bgp-instance-evpn1-af-evpn] peer 3.3.3.3 enable
[*Device1-bgp-instance-evpn1-af-evpn] peer 4.4.4.4 enable
[*Device1-bgp-instance-evpn1-af-evpn] quit
[*Device1-bgp-instance-evpn1] quit
[*Device1] commit
```

Step 6 Configure an EVPN instance on Device 1, Device 2, and Device 3.

# Configure Device 1. Repeat this step for Device 2 and Device 3.

```
[~Device1] bridge-domain 10
[~Device1-bd10] vxlan vni 5010
[*Device1-bd10] evpn
[*Device1-bd10-evpn] route-distinguisher 11:1
[*Device1-bd10-evpn] vpn-target 1:1
[*Device1-bd10-evpn] quit
[*Device1-bd10] quit
[*Device1] bridge-domain 20
[*Device1-bd20] vxlan vni 5020
[*Device1-bd20] evpn
[*Device1-bd20-evpn] route-distinguisher 12:1
[*Device1-bd20-evpn] vpn-target 2:2
[*Device1-bd20-evpn] quit
[*Device1-bd20] quit
[*Device1] commit
```

Step 7 Configure an ingress replication list.

# Configure Device 1. Repeat this step for Device 2 and Device 3.

```
[~Device1] interface nve 1
[*Device1-Nve1] source 2.2.2.2
[*Device1-Nve1] vni 5010 head-end peer-list protocol bgp
[*Device1-Nve1] vni 5020 head-end peer-list protocol bgp
```

```
[*Device1-Nve1] quit
[*Device1] commit
```

Step 8 Configure Device 2 as a Layer 3 VXLAN gateway.

```
[~Device2] interface vbdif 10
[*Device2-Vbdif10] ip address 192.168.10.10 24
[*Device2-Vbdif10] quit
[*Device2] interface vbdif 20
[*Device2-Vbdif20] ip address 192.168.20.10 24
[*Device2-Vbdif20] quit
[*Device2] commit
```

Step 9 Verify the configuration.

After completing the configurations, run the **display vxlan tunnel** and **display vxlan vni** commands on Device 1, Device 2, and Device 3 to check the VXLAN tunnel and VNI information, respectively. The VNIs are Up. The following example shows the command output on Device 1.

```
[~Device1] display vxlan tunnel
Number of vxlan tunnel : 2
Tunnel ID   Source           Destination      State  Type
--------------------------------------------------------------
4026531843  2.2.2.2          3.3.3.3          up     dynamic
4026531844  2.2.2.2          4.4.4.4          up     dynamic
[~Device1] display vxlan vni
Number of vxlan vni : 2
VNI            BD-ID            State
---------------------------------------
5010           10               up
5020           20               up
```

VM1s on different servers can communicate.

----End

Configuration Files

- Device 1 configuration file

```
#
sysname Device1
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
evpn-overlay enable
#
bridge-domain 10
 vxlan vni 5010
 evpn
  route-distinguisher 11:1
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
bridge-domain 20
 vxlan vni 5020
 evpn
  route-distinguisher 12:1
  vpn-target 2:2 export-extcommunity
  vpn-target 2:2 import-extcommunity
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.1.1 255.255.255.0
#
interface 10GE1/0/2.1 mode l2
```

```
 encapsulation dot1q vid 10
 bridge-domain 10
#
interface 10GE1/0/3.1 mode l2
 encapsulation dot1q vid 30
 bridge-domain 20
#
interface LoopBack1
 ip address 2.2.2.2 255.255.255.255
#
interface Nve1
 source 2.2.2.2
 vni 5010 head-end peer-list protocol bgp
 vni 5020 head-end peer-list protocol bgp
#
bgp 100 instance evpn1
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack1
 peer 4.4.4.4 as-number 100
 peer 4.4.4.4 connect-interface LoopBack1
 #
 l2vpn-family evpn
  policy vpn-target
  peer 3.3.3.3 enable
  peer 4.4.4.4 enable
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Device 2 configuration file

```
#
sysname Device2
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
evpn-overlay enable
#
bridge-domain 10
 vxlan vni 5010
 evpn
  route-distinguisher 21:1
  vpn-target 1:1 export-extcommunity
  vpn-target 1:1 import-extcommunity
#
bridge-domain 20
 vxlan vni 5020
 evpn
  route-distinguisher 23:1
  vpn-target 2:2 export-extcommunity
  vpn-target 2:2 import-extcommunity
#
interface Vbdif10
 ip address 192.168.10.10 255.255.255.0
#
interface Vbdif20
 ip address 192.168.20.10 255.255.255.0
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 192.168.2.1 255.255.255.0
```

```
#
interface LoopBack1
 ip address 3.3.3.3 255.255.255.255
#
interface Nve1
 source 3.3.3.3
 vni 5010 head-end peer-list protocol bgp
 vni 5020 head-end peer-list protocol bgp
#
bgp 100 instance evpn1
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack1
 peer 4.4.4.4 as-number 100
 peer 4.4.4.4 connect-interface LoopBack1
 #
 l2vpn-family evpn
  policy vpn-target
  peer 2.2.2.2 enable
  peer 4.4.4.4 enable
#
ospf 1
 area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 192.168.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
#
return
```

- Device 3 configuration file

```
#
sysname Device3
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
evpn-overlay enable
#
bridge-domain 20
 vxlan vni 5020
 evpn
  route-distinguisher 31:2
  vpn-target 2:2 export-extcommunity
  vpn-target 1:1 export-extcommunity
  vpn-target 2:2 import-extcommunity
  vpn-target 1:1 import-extcommunity
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.2.2 255.255.255.0
#
interface 10GE1/0/2.1 mode l2
 encapsulation dot1q vid 20
 bridge-domain 20
#
interface LoopBack1
 ip address 4.4.4.4 255.255.255.255
#
interface Nve1
 source 4.4.4.4
 vni 5020 head-end peer-list protocol bgp
#
bgp 100 instance evpn1
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack1
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack1
 #
 l2vpn-family evpn
  policy vpn-target
  peer 2.2.2.2 enable
  peer 3.3.3.3 enable
#
ospf 1
 area 0.0.0.0
  network 4.4.4.4 0.0.0.0
  network 192.168.2.0 0.0.0.255
#
return
```

12.3 Example for Configuring VXLAN in Distributed Gateway Mode Using MP-BGP

This section provides an example for configuring VXLAN in distributed gateway mode using MP-BGP.

Networking Requirements

Distributed VXLAN gateways can be configured to address problems that occur in legacy centralized VXLAN gateway networking, for example, forwarding paths are not optimal, and the ARP entry specification is a bottleneck.

On the network shown in **Figure 12-3**, an enterprise has VMs deployed in different data centers. VM 1 on Server 1 belongs to VLAN 10, and VM 1 on Server 2 belongs to VLAN 20. VM 1 on Server 1 and VM 1 on Server 2 reside on different network segments. To allow VM1s in different data centers to communicate with each other, configure distributed VXLAN gateways.

Figure 12-3 VXLAN in distributed gateway mode using MP-BGP

NOTE

Interface 1, Interface 2, Interface 3, and Interface 4 represent 10GE 1/0/0, 10GE 1/0/1, 10GE 1/0/2, 10GE 1/0/3, respectively.

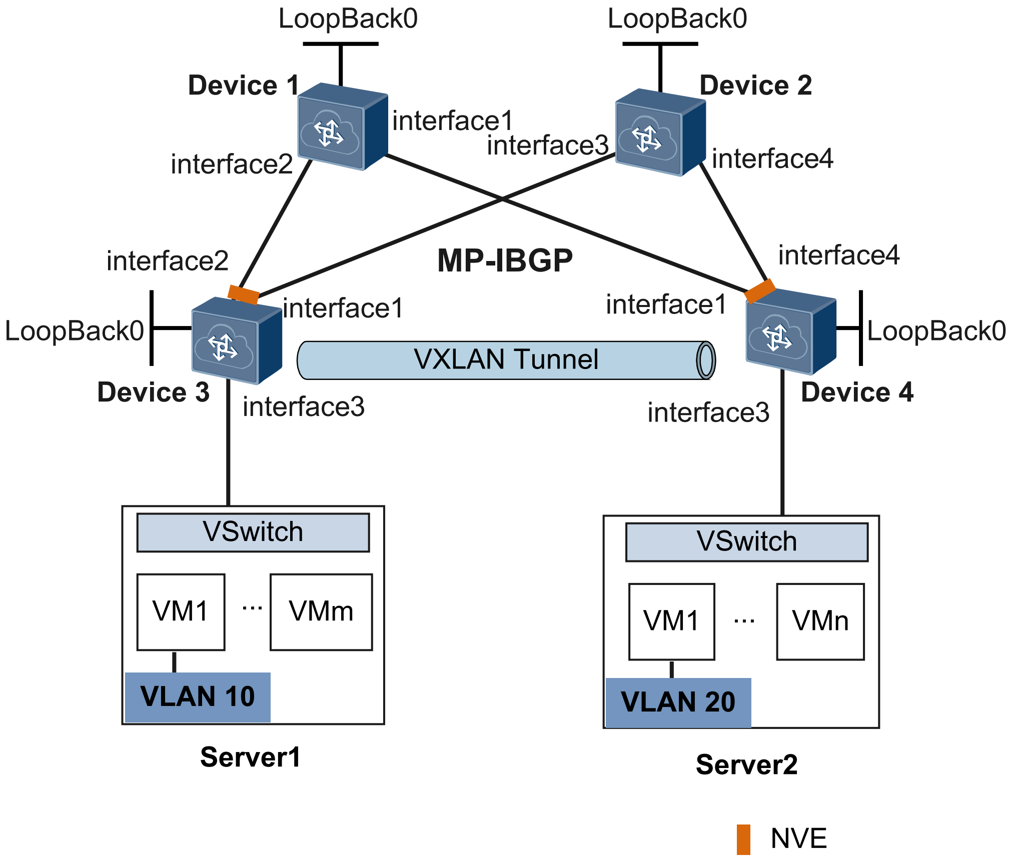


Table 12-1 Interface IP addresses

Device	Interface	IP Address
Device 1	10GE 1/0/0	192.168.2.1/24
	10GE 1/0/1	192.168.1.1/24
	LoopBack0	1.1.1.1/32
Device 2	10GE 1/0/2	192.168.3.1/24
	10GE 1/0/3	192.168.4.1/24
	LoopBack0	2.2.2.2/32
Device 3	10GE 1/0/0	192.168.3.2/24
	10GE 1/0/1	192.168.1.2/24
	LoopBack0	3.3.3.3/32
Device 4	10GE 1/0/0	192.168.2.2/24
	10GE 1/0/3	192.168.4.2/24
	LoopBack0	4.4.4.4/32

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure an IGP on Device 1, Device 2, Device 3, and Device 4; configure BGP and BGP/MPLS IP VPN to ensure Layer 3 networking.
2. Configure a service access point on Device 3 and Device 4 to differentiate service traffic.
3. Configure a Layer 3 VXLAN tunnel between Device 3 and Device 4 to forward service traffic.
4. Configure Device 3 and Device 4 as Layer 3 VXLAN gateways to allow users on different network segments to communicate.
5. Configure BGP on Device 1, Device 2, Device 3, and Device 4 to advertise remote-nexthop attributes (tunnel address, L3VPN VNIs, and MAC addresses) to IBGP peers for Layer 3 VXLAN tunnel connectivity.

Data Preparation

To complete the configuration, you need the following data.

- VMs' VLAN IDs (10 and 20)
- IP addresses of interfaces connecting devices
- IGP running between devices (OSPF in this example)
- RRs (Device 1 and Device 2) and RR clients (Device 3 and Device 4)
- BD IDs (10 and 20)
- VNI ID (10)

Procedure

Step 1 Configure a Layer 3 network.

1. Assign an IP address to each interface and configure OSPF.

   # Configure Device 1. Repeat this step for Device 2, Device 3, and Device 4. Configure the devices to advertise the 32-bit IP addresses of loopback interfaces.

```
<HUAWEI> system-view
[~HUAWEI] sysname Device1
[*HUAWEI] commit
[~Device1] interface loopback 0
[*Device1-LoopBack0] ip address 1.1.1.1 32
[*Device1-LoopBack0] quit
[*Device1] interface 10ge 1/0/0
[*Device1-10GE1/0/0] undo portswitch
[*Device1-10GE1/0/0] ip address 192.168.2.1 24
[*Device1-10GE1/0/0] quit
[*Device1] interface 10ge 1/0/1
[*Device1-10GE1/0/1] undo portswitch
[*Device1-10GE1/0/1] ip address 192.168.1.1 24
[*Device1-10GE1/0/1] quit
[*Device1] ospf
[*Device1-ospf-1] area 0
[*Device1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[*Device1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[*Device1-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[*Device1-ospf-1-area-0.0.0.0] quit
[*Device1-ospf-1] quit
[*Device1] commit
```

# After OSPF is configured, the devices can use OSPF to learn the IP addresses of each other's loopback interfaces and successfully ping each other. The following example shows the command output on Device 1 after it pings Device 4:

```
[~Device1] ping 4.4.4.4
  PING 4.4.4.4: 56  data bytes, press CTRL_C to break
    Reply from 4.4.4.4: bytes=56 Sequence=1 ttl=253 time=55 ms
    Reply from 4.4.4.4: bytes=56 Sequence=2 ttl=253 time=3 ms
    Reply from 4.4.4.4: bytes=56 Sequence=3 ttl=253 time=4 ms
    Reply from 4.4.4.4: bytes=56 Sequence=4 ttl=253 time=3 ms
    Reply from 4.4.4.4: bytes=56 Sequence=5 ttl=253 time=3 ms

  --- 4.4.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 3/13/55 ms
```

2. Configure BGP and BGP/MPLS IP VPN. Configure Device 1 and Device 2 as RRs and Device 3 and Device 4 as RR clients.

   # Configure Device 1. Repeat this step for Device 2.

```
[~Device1] bgp 100
[*Device1-bgp] router-id 1.1.1.1
[*Device1-bgp] peer 3.3.3.3 as-number 100
[*Device1-bgp] peer 3.3.3.3 connect-interface LoopBack0
[*Device1-bgp] peer 4.4.4.4 as-number 100
[*Device1-bgp] peer 4.4.4.4 connect-interface LoopBack0
[*Device1-bgp] ipv4-family vpnv4
[*Device1-bgp-af-vpnv4] undo policy vpn-target
[*Device1-bgp-af-vpnv4] peer 3.3.3.3 enable
[*Device1-bgp-af-vpnv4] peer 3.3.3.3 reflect-client
[*Device1-bgp-af-vpnv4] peer 4.4.4.4 enable
[*Device1-bgp-af-vpnv4] peer 4.4.4.4 reflect-client
[*Device1-bgp-af-vpnv4] quit
[*Device1-bgp] quit
[*Device1] commit
```

   # Configure Device 3. Repeat this step for Device 4.

```
[~Device3] ip vpn-instance vrf1
[*Device3-vpn-instance-vrf1] ipv4-family
[*Device3-vpn-instance-vrf1-af-ipv4] route-distinguisher 100:1
[*Device3-vpn-instance-vrf1-af-ipv4] vpn-target 100:1 export-extcommunity
[*Device3-vpn-instance-vrf1-af-ipv4] vpn-target 100:1 import-extcommunity
[*Device3-vpn-instance-vrf1-af-ipv4] quit
[*Device3-vpn-instance-vrf1] vxlan vni 10
[*Device3-vpn-instance-vrf1] quit
[*Device3] bgp 100
[*Device3-bgp] router-id 3.3.3.3
[*Device3-bgp] peer 1.1.1.1 as-number 100
[*Device3-bgp] peer 1.1.1.1 connect-interface LoopBack0
[*Device3-bgp] peer 2.2.2.2 as-number 100
[*Device3-bgp] peer 2.2.2.2 connect-interface LoopBack0
[*Device3-bgp] ipv4-family vpnv4
[*Device3-bgp-af-vpnv4] peer 1.1.1.1 enable
[*Device3-bgp-af-vpnv4] peer 2.2.2.2 enable
[*Device3-bgp-af-vpnv4] quit
[*Device3-bgp] ipv4-family vpn-instance vrf1
[*Device3-bgp-vrf1] import-route direct
[*Device3-bgp-vrf1] quit
[*Device3-bgp] quit
[*Device3] commit
```

Step 2 Configure the VXLAN tunnel mode and enable the VXLAN ACL extension function.

# Configure Device3. The configurations on Device4 is similar to that on Device3, and are not mentioned here.

```
[~Device3] ip tunnel mode vxlan
[*Device3] assign forward nvo3 acl extend enable
[*Device3] commit
```

NOTE

After modifying the VXLAN tunnel mode or enabling the VXLAN ACL extension function, you need to save the configuration and restart the device to make the configuration take effect. You can restart the device immediately or after completing all the configurations.

Step 3 Configure a service access point on Device 3 and Device 4.

# Configure Device 3. Repeat this step for Device 4.

```
[~Device3] bridge-domain 10
[*Device3-bd10] quit
[*Device3] interface 10ge 1/0/2.1 mode l2
[*Device3-10GE1/0/2.1] encapsulation dot1q vid 10
[*Device3-10GE1/0/2.1] bridge-domain 10
[*Device3-10GE1/0/2.1] quit
[*Device3] commit
```

Step 4 Establish a Layer 3 VXLAN tunnel between Device 3 and Device 4.

# Configure Device 3. Repeat this step for Device 4.

```
[~Device3] interface Nve1
[*Device3-Nve1] mode l3
[*Device3-Nve1] source 3.3.3.3
[*Device3-Nve1] quit
[*Device3] commit
```

Step 5 Configure Device 3 and Device 4 as Layer 3 VXLAN gateways.

# Configure a Layer 3 VXLAN gateway on Device 3. The configuration on Device 4 is similar to that on Device 3, and is not mentioned here. The IP addresses of BDIF interfaces on Device 3 and Device 4 must be on different network segments.

```
[~Device3] interface Vbdif10
[*Device3-Vbdif10] ip binding vpn-instance vrf1
[*Device3-Vbdif10] ip address 10.1.1.1 255.255.255.0
[*Device3-Vbdif10] quit
[*Device3] commit
```

Step 6 Configure BGP on Device 1, Device 2, Device 3, and Device 4 to advertise the remote-nexthop attribute to IBGP peers.

# Configure Device 1. Repeat this step for Device 2, Device 3, and Device 4.

```
[~Device1] bgp 100
[~Device1-bgp] ipv4-family vpnv4
[~Device1-bgp-af-vpnv4] peer 3.3.3.3 advertise remote-nexthop
[*Device1-bgp-af-vpnv4] peer 4.4.4.4 advertise remote-nexthop
[*Device1-bgp-af-vpnv4] quit
[*Device1-bgp] quit
[*Device1] commit
```

Step 7 Verify the configuration.

After completing the configurations, run the **display vxlan tunnel** command on Device 3 and Device 4 to check VXLAN tunnel information. The following example uses the command output on Device 4.

```
[~Device4] display vxlan tunnel
Number of vxlan tunnel : 1
Tunnel ID   Source           Destination      State  Type
----------------------------------------------------------------
4026531841  4.4.4.4          3.3.3.3          up     dynamic
```

VM1s on different servers can communicate.

----End

Configuration Files

- Device 1 configuration file

```
#
sysname Device1
#
interface 10GE1/0/0
 undo portswitch
 ip address 192.168.2.1 255.255.255.0
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.1.1 255.255.255.0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 router-id 1.1.1.1
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack0
 peer 4.4.4.4 as-number 100
 peer 4.4.4.4 connect-interface LoopBack0
 #
 ipv4-family unicast
  peer 3.3.3.3 enable
  peer 4.4.4.4 enable
 #
 ipv4-family vpnv4
  undo policy vpn-target
  peer 3.3.3.3 enable
  peer 3.3.3.3 reflect-client
  peer 3.3.3.3 advertise remote-nexthop
  peer 4.4.4.4 enable
  peer 4.4.4.4 reflect-client
  peer 4.4.4.4 advertise remote-nexthop
#
ospf 1
 area 0.0.0.0
  network 1.1.1.1 0.0.0.0
  network 192.168.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
#
return
```

- Device 2 configuration file

```
#
sysname Device2
#
interface 10GE1/0/2
 undo portswitch
 ip address 192.168.3.1 255.255.255.0
#
interface 10GE1/0/3
 undo portswitch
 ip address 192.168.4.1 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 100
 router-id 2.2.2.2
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack0
 peer 4.4.4.4 as-number 100
 peer 4.4.4.4 connect-interface LoopBack0
 #
 ipv4-family unicast
  peer 3.3.3.3 enable
  peer 4.4.4.4 enable
 #
 ipv4-family vpnv4
  undo policy vpn-target
  peer 3.3.3.3 enable
  peer 3.3.3.3 reflect-client
  peer 3.3.3.3 advertise remote-nexthop
  peer 4.4.4.4 enable
  peer 4.4.4.4 reflect-client
  peer 4.4.4.4 advertise remote-nexthop
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 192.168.3.0 0.0.0.255
  network 192.168.4.0 0.0.0.255
#
return
```

- Device 3 configuration file

```
#
sysname Device3
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
ip vpn-instance vrf1
 ipv4-family
  route-distinguisher 100:1
  vpn-target 100:1 export-extcommunity
  vpn-target 100:1 import-extcommunity
 vxlan vni 10
#
bridge-domain 10
#
interface Vbdif10
 ip binding vpn-instance vrf1
 ip address 10.1.1.1 255.255.255.0
#
interface 10GE1/0/0
 undo portswitch
 ip address 192.168.3.2 255.255.255.0
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/2.1 mode l2
 encapsulation dot1q vid 10
 bridge-domain 10
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
interface Nve1
 mode l3
 source 3.3.3.3
#
bgp 100
 router-id 3.3.3.3
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack0
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack0
 #
 ipv4-family unicast
  peer 1.1.1.1 enable
  peer 2.2.2.2 enable
 #
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.1 enable
  peer 1.1.1.1 advertise remote-nexthop
  peer 2.2.2.2 enable
  peer 2.2.2.2 advertise remote-nexthop
 #
 ipv4-family vpn-instance vrf1
  import-route direct
#
ospf 1
 area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 192.168.1.0 0.0.0.255
  network 192.168.3.0 0.0.0.255
#
return
```

- Device 4 configuration file

```
#
sysname Device4
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
ip vpn-instance vrf1
 ipv4-family
  route-distinguisher 100:1
  vpn-target 100:1 export-extcommunity
  vpn-target 100:1 import-extcommunity
 vxlan vni 10
#
bridge-domain 20
#
interface Vbdif20
 ip binding vpn-instance vrf1
 ip address 20.1.1.1 255.255.255.0
#
interface 10GE1/0/0
 undo portswitch
 ip address 192.168.2.2 255.255.255.0
#
interface 10GE1/0/2.1 mode l2
 encapsulation dot1q vid 20
 bridge-domain 20
#
interface 10GE1/0/3
 undo portswitch
 ip address 192.168.4.2 255.255.255.0
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
interface Nve2
 mode l3
 source 4.4.4.4
#
bgp 100
 router-id 4.4.4.4
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack0
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack0
 #
 ipv4-family unicast
  peer 1.1.1.1 enable
  peer 2.2.2.2 enable
 #
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.1 enable
  peer 1.1.1.1 advertise remote-nexthop
  peer 2.2.2.2 enable
  peer 2.2.2.2 advertise remote-nexthop
 #
 ipv4-family vpn-instance vrf1
  import-route direct
#
ospf 1
 area 0.0.0.0
  network 4.4.4.4 0.0.0.0
  network 192.168.2.0 0.0.0.255
  network 192.168.4.0 0.0.0.255
#
return
```

12.4 Example for Configuring VXLAN in Distributed Gateway Mode Using BGP EVPN

This section provides an example for configuring VXLAN in distributed gateway mode using BGP EVPN.

Networking Requirements

Distributed VXLAN gateways can be configured to address problems that occur in legacy centralized VXLAN gateway networking, for example, forwarding paths are not optimal, and the ARP entry specification is a bottleneck.

On the network shown in **Figure 12-4**, an enterprise has VMs deployed in different data centers. VM 1 on Server 1 belongs to VLAN 10, and VM 1 on Server 2 belongs to VLAN 20. VM 1 on Server 1 and VM 1 on Server 2 reside on different network segments. To allow VM1s in different data centers to communicate with each other, configure distributed VXLAN gateways. Device 1 is deployed in AS 100, Device 2 is deployed in AS 200, and Device 3 is deployed in AS300. Device 1, Device 2 and Device 3 use AS 100 for BGP EVPN.

Figure 12-4 VXLAN in distributed gateway mode using BGP EVPN

NOTE

Interface 1 and Interface 2 represent 10GE 1/0/0 and 10GE 1/0/1, respectively.

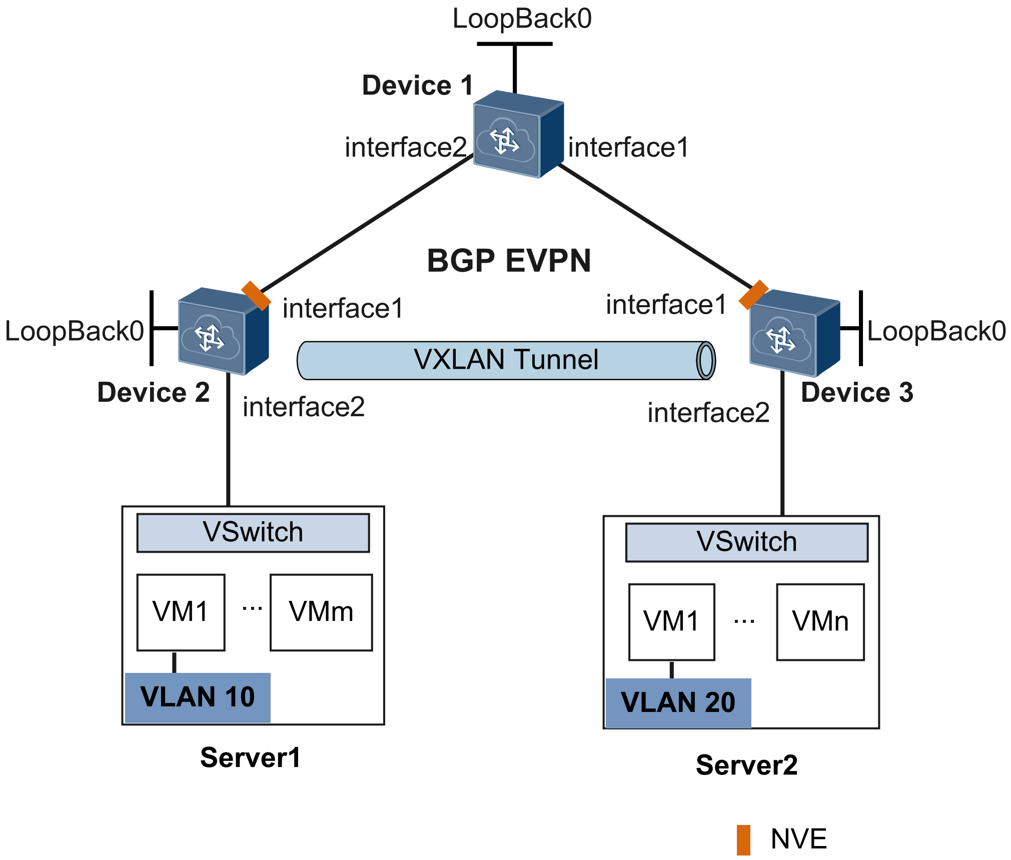


Table 12-2 Interface IP addresses

Device	Interface	IP Address
Device 1	10GE1/0/0	192.168.3.2/24
	10GE1/0/1	192.168.2.2/24
	LoopBack0	1.1.1.1/32
Device 2	10GE1/0/0	192.168.2.1/24
	LoopBack0	2.2.2.2/32
Device 3	10GE1/0/0	192.168.3.1/24
	LoopBack0	3.3.3.3/32

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure EBGP to run between Device 1 and Device 2 and between Device 1 and Device 3.
2. Configure a service access point on Device 2 and Device 3 to differentiate service traffic.
3. Configure EVPN as the VXLAN control plane.

4. Specify Device 1 as an BGP EVPN peer for Device 2 and Device 3.
5. Specify Device 2 and Device 3 as BGP EVPN peers for Device 1 and configure Device 2 and Device 3 as RR clients.
6. Configure VPN and EVPN instances on Device 2 and Device 3.
7. Configure an ingress replication list on Device 2 and Device 3.
8. Configure Device 2 and Device 3 as Layer 3 VXLAN gateways.
9. Configure BGP to advertise IRB routes between Device 1 and Device 2 and between Device 1 and Device 3.

Data Preparation

To complete the configuration, you need the following data.

- VMs' VLAN IDs (10 and 20)
- IP addresses of interfaces connecting devices
- BD IDs (10 and 20)
- VNI IDs (10 and 20)
- VNI ID (5010)

Procedure

Step 1 Configure a routing protocol.

# Configure Device 1. Repeat this step for Device 2 and Device 3.

```
<HUAWEI> system-view
[~HUAWEI] sysname Device1
[*HUAWEI] commit
[~Device1] interface loopback 0
[*Device1-LoopBack0] ip address 1.1.1.1 32
[*Device1-LoopBack0] quit
[*Device1] interface 10ge 1/0/0
[*Device1-10GE1/0/0] undo portswitch
[*Device1-10GE1/0/0] ip address 192.168.3.2 24
[*Device1-10GE1/0/0] quit
[*Device1] interface 10ge 1/0/1
[*Device1-10GE1/0/1] undo portswitch
[*Device1-10GE1/0/1] ip address 192.168.2.2 24
[*Device1-10GE1/0/1] quit
[*Device1] bgp 100
[*Device1-bgp] peer 192.168.2.1 as-number 200
[*Device1-bgp] peer 192.168.3.1 as-number 300
[*Device1-bgp] network 1.1.1.1 32
[*Device1-bgp] quit
[*Device1] commit
```

Step 2 Configure the VXLAN tunnel mode and enable the VXLAN ACL extension function.

# Configure Device2. The configurations on Device3 is similar to that on Device2, and are not mentioned here.

```
[~Device2] ip tunnel mode vxlan
[*Device2] assign forward nvo3 acl extend enable
[*Device2] commit
```

NOTE

After modifying the VXLAN tunnel mode or enabling the VXLAN ACL extension function, you need to save the configuration and restart the device to make the configuration take effect. You can restart the device immediately or after completing all the configurations.

Step 3 Configure a service access point on Device 2 and Device 3.

# Configure Device 2. Repeat this step for Device 3.

```
[~Device2] bridge-domain 10
[*Device2-bd10] quit
[*Device2] interface 10ge 1/0/1.1 mode l2
[*Device2-10GE1/0/1.1] encapsulation dot1q vid 10
[*Device2-10GE1/0/1.1] bridge-domain 10
[*Device2-10GE1/0/1.1] quit
[*Device2] commit
```

Step 4 Configure EVPN as the VXLAN control plane.

# Configure Device 1. Repeat this step for Device 2 and Device 3.

```
[~Device1] evpn-overlay enable
[*Device1] commit
```

Step 5 Specify Device 1 as an BGP EVPN peer for Device 2 and Device 3.

# Specify Device 1 as an BGP EVPN peer for Device 2. Repeat this step for Device 3.

```
[~Device2] bgp 100 instance evpn1
[*Device2-bgp-instance-evpn1] peer 1.1.1.1 as-number 100
[*Device2-bgp-instance-evpn1] peer 1.1.1.1 connect-interface LoopBack0
[*Device2-bgp-instance-evpn1] l2vpn-family evpn
[*Device2-bgp-instance-evpn1-af-evpn] peer 1.1.1.1 enable
[*Device2-bgp-instance-evpn1-af-evpn] quit
[*Device2-bgp-instance-evpn1] quit
[*Device2] commit
```

Step 6 Specify Device 2 and Device 3 as BGP EVPN peers for Device 1 and configure them as RR clients.

# Specify BGP EVPN peers for Device 1.

```
[~Device1] bgp 100 instance evpn1
[*Device1-bgp-instance-evpn1] peer 2.2.2.2 as-number 100
[*Device1-bgp-instance-evpn1] peer 2.2.2.2 connect-interface LoopBack0
[*Device1-bgp-instance-evpn1] peer 3.3.3.3 as-number 100
[*Device1-bgp-instance-evpn1] peer 3.3.3.3 connect-interface LoopBack0
[*Device1-bgp-instance-evpn1] l2vpn-family evpn
[*Device1-bgp-instance-evpn1-af-evpn] peer 2.2.2.2 enable
[*Device1-bgp-instance-evpn1-af-evpn] peer 2.2.2.2 reflect-client
[*Device1-bgp-instance-evpn1-af-evpn] peer 3.3.3.3 enable
[*Device1-bgp-instance-evpn1-af-evpn] peer 3.3.3.3 reflect-client
[*Device1-bgp-instance-evpn1-af-evpn] undo policy vpn-target
[*Device1-bgp-instance-evpn1-af-evpn] quit
[*Device1-bgp-instance-evpn1] quit
[*Device1] commit
```

Step 7 Configure VPN and EVPN instances on Device 2 and Device 3.

# Configure VPN and EVPN instances on Device 2. Repeat this step for Device 3.

```
[~Device2] ip vpn-instance vpn1
[*Device2-vpn-instance-vpn1] vxlan vni 5010
[*Device2-vpn-instance-vpn1] ipv4-family
[*Device2-vpn-instance-vpn1-af-ipv4] route-distinguisher 11:11
[*Device2-vpn-instance-vpn1-af-ipv4] vpn-target 1:1
[*Device2-vpn-instance-vpn1-af-ipv4] vpn-target 11:1 evpn
[*Device2-vpn-instance-vpn1-af-ipv4] quit
[*Device2-vpn-instance-vpn1] quit
[*Device2] bridge-domain 10
[*Device2-bd10] vxlan vni 10
[*Device2-bd10] evpn
[*Device2-bd10-evpn] route-distinguisher 10:1
[*Device2-bd10-evpn] vpn-target 11:1
[*Device2-bd10-evpn] quit
[*Device2-bd10] quit
[*Device2] commit
```

Step 8 Configure an ingress replication list on Device 2 and Device 3.

# Configure Device 2. Repeat this step for Device 3.

```
[~Device2] interface nve 1
[*Device2-Nve1] source 2.2.2.2
[*Device2-Nve1] vni 10 head-end peer-list protocol bgp
[*Device2-Nve1] quit
[*Device2] commit
```

Step 9 Configure Device 2 and Device 3 as Layer 3 VXLAN gateways.

# Configure a Layer 3 VXLAN gateway on Device 2. The configuration on Device 3 is similar to that on Device 2, and is not mentioned here. The IP addresses of BDIF interfaces on Device 2 and Device 3 must be on different network segments.

```
[~Device2] interface Vbdif10
[*Device2-Vbdif10] ip binding vpn-instance vpn1
[*Device2-Vbdif10] ip address 10.1.1.1 255.255.255.0
[*Device2-Vbdif10] arp distribute-gateway enable
[*Device2-Vbdif10] arp collect host enable
[*Device2-Vbdif10] quit
[*Device2] commit
```

Step 10 Configure BGP to advertise IRB routes between Device 1 and Device 2 and between Device 1 and Device 3.

# Configure Device 1. The configurations of Device 2 and Device 3 are similar to the configuration of Device1, and are not mentioned here.

```
[~Device1] bgp 100 instance evpn1
[~Device1-bgp-instance-evpn1] l2vpn-family evpn
[~Device1-bgp-instance-evpn1-af-evpn] peer 2.2.2.2 advertise irb
[*Device1-bgp-instance-evpn1-af-evpn] peer 3.3.3.3 advertise irb
[*Device1-bgp-instance-evpn1-af-evpn] quit
[*Device1-bgp-instance-evpn1] quit
[*Device1] commit
```

Step 11 Verify the configuration.

After completing the configurations, run the **display vxlan tunnel** command on Device 2 and Device 3 to check VXLAN tunnel information. The following example shows the command output on Device 2.

```
[~Device2] display vxlan tunnel
Number of vxlan tunnel : 1
Tunnel ID   Source           Destination      State  Type
-------------------------------------------------------------
4026531841  2.2.2.2          3.3.3.3          up     dynamic
```

VM1s on different servers can communicate.

----End

Configuration Files

- Device 1 configuration file

```
#
sysname Device1
#
evpn-overlay enable
#
interface 10GE1/0/0
 undo portswitch
 ip address 192.168.3.2 255.255.255.0
#
interface 10GE1/0/1
```

```
 undo portswitch
 ip address 192.168.2.2 255.255.255.0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 peer 192.168.2.1 as-number 200
 peer 192.168.3.1 as-number 300
 #
 ipv4-family unicast
  network 1.1.1.1 255.255.255.255
  peer 192.168.2.1 enable
  peer 192.168.3.1 enable
#
bgp 100 instance evpn1
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack0
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack0
 #
 l2vpn-family evpn
  undo policy vpn-target
  peer 2.2.2.2 enable
  peer 2.2.2.2 advertise irb
  peer 2.2.2.2 reflect-client
  peer 3.3.3.3 enable
  peer 3.3.3.3 advertise irb
  peer 3.3.3.3 reflect-client
#
return
```

- Device 2 configuration file

```
#
sysname Device2
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
evpn-overlay enable
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 11:11
  vpn-target 1:1 export-extcommunity
  vpn-target 11:1 export-extcommunity evpn
  vpn-target 1:1 import-extcommunity
  vpn-target 11:1 import-extcommunity evpn
 vxlan vni 5010
#
bridge-domain 10
 vxlan vni 10
 evpn
  route-distinguisher 10:1
  vpn-target 11:1 export-extcommunity
  vpn-target 11:1 import-extcommunity
#
interface Vbdif10
 ip binding vpn-instance vpn1
 ip address 10.1.1.1 255.255.255.0
 arp distribute-gateway enable
 arp collect host enable
#
interface 10GE1/0/0
 undo portswitch
 ip address 192.168.2.1 255.255.255.0
#
interface 10GE1/0/1.1 mode l2
 encapsulation dot1q vid 10
 bridge-domain 10
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
interface Nve1
 source 2.2.2.2
 vni 10 head-end peer-list protocol bgp
#
bgp 200
 peer 192.168.2.2 as-number 100
 #
 ipv4-family unicast
  network 2.2.2.2 255.255.255.255
  peer 192.168.2.2 enable
#
bgp 100 instance evpn1
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack0
 #
 l2vpn-family evpn
  policy vpn-target
  peer 1.1.1.1 enable
  peer 1.1.1.1 advertise irb
#
return
```

- Device 3 configuration file

```
#
sysname Device3
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
evpn-overlay enable
#
ip vpn-instance vpn1
 ipv4-family
  route-distinguisher 22:22
  vpn-target 2:2 export-extcommunity
  vpn-target 11:1 export-extcommunity evpn
  vpn-target 2:2 import-extcommunity
  vpn-target 11:1 import-extcommunity evpn
 vxlan vni 5010
#
bridge-domain 20
 vxlan vni 20
 evpn
  route-distinguisher 20:1
  vpn-target 11:1 export-extcommunity
  vpn-target 11:1 import-extcommunity
#
interface Vbdif20
 ip binding vpn-instance vpn1
 ip address 20.1.1.1 255.255.255.0
 arp distribute-gateway enable
 arp collect host enable
#
interface 10GE1/0/0
 undo portswitch
 ip address 192.168.3.1 255.255.255.0
#
interface 10GE1/0/1.1 mode l2
 encapsulation dot1q vid 20
 bridge-domain 20
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
```

```
interface Nve1
 source 3.3.3.3
 vni 20 head-end peer-list protocol bgp
#
bgp 300
 peer 192.168.3.2 as-number 100
 #
 ipv4-family unicast
  network 3.3.3.3 255.255.255.255
  peer 192.168.3.2 enable
#
bgp 100 instance evpn1
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack0
 #
 l2vpn-family evpn
  policy vpn-target
  peer 1.1.1.1 enable
  peer 1.1.1.1 advertise irb
#
return
```

12.5 Example for Configuring All-Active VXLAN Gateways

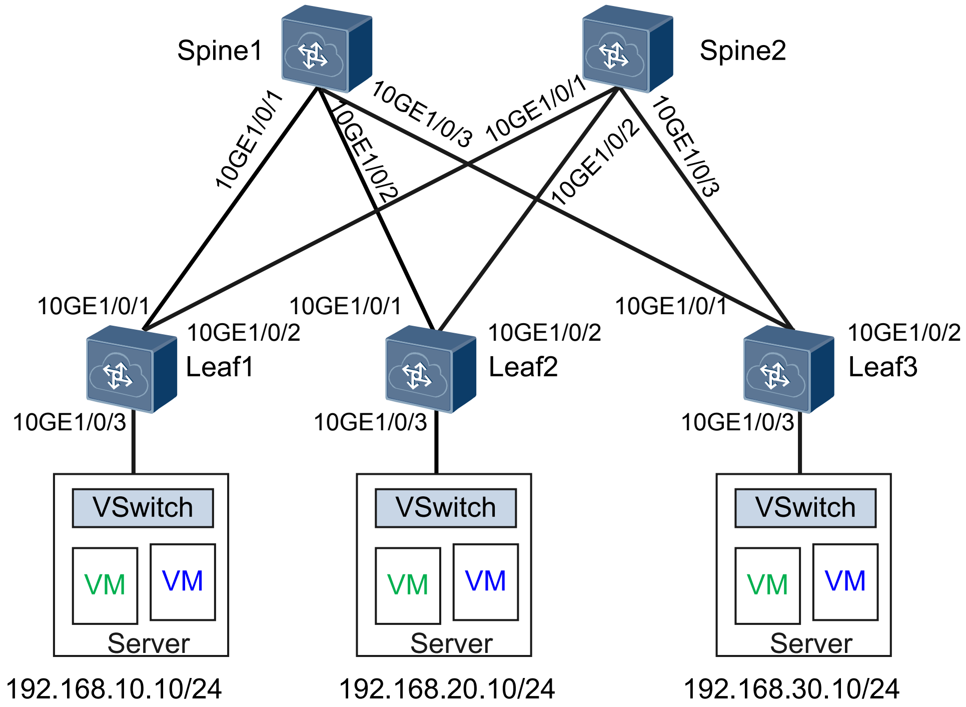
Networking Requirements

As shown in **Figure 12-5**, the intranet of an enterprise data center uses the two Spine-Leaf structure.

- Spine1 and Spine2 work as backbone nodes at the aggregation layer of the network.
- Leaf1 to Leaf3 work as leaf nodes at the access layer of the network.
- The Leaf devices are fully connected to the Spine devices. They constitute equal-cost multipath (ECMP) to achieve network high availability (HA). The spine devices are not connected to each other, so are the leaf devices.

Major traffic in the data center is the east-west traffic caused by VM migration. Therefore, the customer wants to construct a large Layer 2 network over the network and wants the spine devices to function as gateways simultaneously, load balancing packets from the leaf devices.

Figure 12-5 Configuring all-active VXLAN gateways



Precautions

To ensure users on different network segments to communicate with each other, the default gateway address must be the IP address of the VBDIF interface on the Layer 3 gateway.

Configuration Roadmap

The configuration roadmap is as follows:

1. Configure OSPF on Leaf1 to Leaf3 as well as Spine1 and Spine2 to ensure Layer 3 network connectivity.
2. Configure VXLAN on Leaf1 to Leaf3 as well as Spine1 and Spine2 to construct a large Layer 2 VXLAN network over the basic Layer 3 network.
3. Configure service access points on Leaf1 to Leaf3 to distinguish traffic from servers and forward the traffic to the VXLAN network.
4. Configure VXLAN Layer 3 gateways on Spine1 and Spine2 to implement communication between VXLAN networks on different network segments and between VXLAN and non-VXLAN networks.

NOTE

If the link through which Spine1 (or Spine2) is uplink connected to the network fails, Spine1 (or Spine2) discards all received user traffic because no uplink outbound interface is available. You can configure a monitor-link to associate the uplink and downlink interfaces of Spine1 (or Spine2). When the uplink outbound interface of Spine1 (or Spine2) becomes Down, the downlink interface also becomes Down. Then user traffic will not be forwarded or discarded by Spine1 (or Spine2). For details about the monitor-link configuration, see Configuring the Uplink and Downlink Interfaces in a Monitor Link Group.

Data Plan

To complete the configuration, you need the following data:

- Interface IP addresses for device interconnection
- Routing protocol: OSPF
- VLAN IDs to which VMs belong: VLAN 10, VLAN 20, and VLAN 30
- BD IDs: BD 10, BD 20, and BD 30
- VNI IDs: VNI 5000, VNI 5001, and VNI 5002

Procedure

Step 1 Configure a routing protocol.

Configure an IP address for each interface on Leaf1 to Leaf3 as well as Spine1 and Spine2. When OSPF is used, the devices advertise the 32-bit loopback IP addresses.

# Configure Leaf1. The configurations on Leaf2 and Leaf3 are similar to that on Leaf1, and are not mentioned here.

```
<HUAWEI> system-view
[~HUAWEI] sysname Leaf1
[*HUAWEI] commit
[~Leaf1] interface loopback 1
[*Leaf1-LoopBack1] ip address 10.10.10.3 32
[*Leaf1-LoopBack1] quit
[*Leaf1] interface 10ge 1/0/1
[*Leaf1-10GE1/0/1] undo portswitch
[*Leaf1-10GE1/0/1] ip address 10.1.1.2 24
[*Leaf1-10GE1/0/1] quit
[*Leaf1] interface 10ge 1/0/2
[*Leaf1-10GE1/0/2] undo portswitch
[*Leaf1-10GE1/0/2] ip address 10.2.1.2 24
[*Leaf1-10GE1/0/2] quit
[*Leaf1] ospf
[*Leaf1-ospf-1] area 0
[*Leaf1-ospf-1-area-0.0.0.0] network 10.10.10.3 0.0.0.0
[*Leaf1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[*Leaf1-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[*Leaf1-ospf-1-area-0.0.0.0] quit
[*Leaf1-ospf-1] quit
[*Leaf1] commit
```

# Configure Spine1. The configuration on Spine2 is similar to that on Spine1, and is not mentioned here.

```
<HUAWEI> system-view
[~HUAWEI] sysname Spine1
[*HUAWEI] commit
[~Spine1] interface loopback 1
[*Spine1-LoopBack1] ip address 10.10.10.1 32
[*Spine1-LoopBack1] quit
[*Spine1] interface loopback 2
[*Spine1-LoopBack2] ip address 10.10.10.10 32
[*Spine1-LoopBack2] quit
[*Spine1] interface 10ge 1/0/1
[*Spine1-10GE1/0/1] undo portswitch
[*Spine1-10GE1/0/1] ip address 10.1.1.1 24
[*Spine1-10GE1/0/1] quit
[*Spine1] interface 10ge 1/0/2
[*Spine1-10GE1/0/2] undo portswitch
[*Spine1-10GE1/0/2] ip address 10.3.1.1 24
[*Spine1-10GE1/0/2] quit
[*Spine1] interface 10ge 1/0/3
[*Spine1-10GE1/0/3] undo portswitch
[*Spine1-10GE1/0/3] ip address 10.5.1.1 24
[*Spine1-10GE1/0/3] quit
[*Spine1] ospf
```

```
[*Spine1-ospf-1] area 0
[*Spine1-ospf-1-area-0.0.0.0] network 10.10.10.1 0.0.0.0
[*Spine1-ospf-1-area-0.0.0.0] network 10.10.10.10 0.0.0.0
[*Spine1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] network 10.3.1.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] network 10.5.1.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] quit
[*Spine1-ospf-1] quit
[*Spine1] commit
```

After the configuration is complete, run the **display ospf peer** command. The command output shows that OSPF neighbor relationships have been set up between the devices, and the neighbor status is Full. Run the **display ip routing-table** command, and you can see that the devices have learned the routes to Loopback of each other.

Step 2 Configure the VXLAN tunnel mode and enable the VXLAN ACL extension function.

# Configure Spine1. The configurations on Spine2, Leaf1, Leaf2, and Leaf3 are similar to that on Spine1, and are not mentioned here.

```
[~Spine1] ip tunnel mode vxlan
[*Spine1] assign forward nvo3 acl extend enable
[*Spine1] commit
```

NOTE

After modifying the VXLAN tunnel mode or enabling the VXLAN ACL extension function, you need to save the configuration and restart the device to make the configuration take effect. You can restart the device immediately or after completing all the configurations.

Step 3 Configure VXLAN on Leaf1 to Leaf3 as well as Spine1 and Spine2 to construct a large Layer 2 VXLAN network.

# Configure Leaf1. The configurations on Leaf2 and Leaf3 are similar to that on Leaf1, and are not mentioned here.

```
[~Leaf1] bridge-domain 10
[*Leaf1-bd10] vxlan vni 5000
[*Leaf1-bd10] quit
[*Leaf1] interface nve 1
[*Leaf1-Nve1] source 10.10.10.3
[*Leaf1-Nve1] vni 5000 head-end peer-list 10.10.10.1
[*Leaf1-Nve1] quit
[*Leaf1] commit
```

# Configure Spine1. The configuration on Spine2 is similar to that on Spine1, and is not mentioned here.

```
[~Spine1] bridge-domain 10
[*Spine1-bd10] vxlan vni 5000
[*Spine1-bd10] quit
[*Spine1] bridge-domain 20
[*Spine1-bd20] vxlan vni 5001
[*Spine1-bd20] quit
[*Spine1] bridge-domain 30
[*Spine1-bd30] vxlan vni 5002
[*Spine1-bd30] quit
[*Spine1] interface nve 1
[*Spine1-Nve1] source 10.10.10.1
[*Spine1-Nve1] vni 5000 head-end peer-list 10.10.10.3
[*Spine1-Nve1] vni 5001 head-end peer-list 10.10.10.4
[*Spine1-Nve1] vni 5002 head-end peer-list 10.10.10.5
[*Spine1-Nve1] quit
[*Spine1] commit
```

After the configuration is complete, run the **display vxlan vni** command on Leaf1 to Leaf3 as well as Spine1 and Spine2. The command output shows that the VNI status is **up**. Run the

display vxlan tunnel command, and you can see VXLAN tunnel information. The display on Spine1 is used as an example.

```
[~Spine1] display vxlan vni
Number of vxlan vni : 3
VNI            BD-ID            State
---------------------------------------
5000           10               up
5001           20               up
5002           30               up
[~Spine1] display vxlan tunnel
Number of vxlan tunnel : 3
Tunnel ID   Source           Destination        State  Type
---------------------------------------------------------------
4026531842  10.10.10.1       10.10.10.3         up     static
4026531843  10.10.10.1       10.10.10.4         up     static
4026531844  10.10.10.1       10.10.10.5         up     static
```

Step 4 Configure service access points on Leaf1 to Leaf3.

# Configure Leaf1. The configurations on Leaf2 and Leaf3 are similar to that on Leaf1, and are not mentioned here.

```
[~Leaf1] vlan 10
[*Leaf1-vlan10] quit
[*Leaf1] bridge-domain 10
[*Leaf1-bd10] l2 binding vlan 10
[*Leaf1-bd10] quit
[*Leaf1] interface 10ge 1/0/3
[*Leaf1-10GE1/0/3] port link-type trunk
[*Leaf1-10GE1/0/3] undo port trunk allow-pass vlan 1
[*Leaf1-10GE1/0/3] port trunk allow-pass vlan 10
[*Leaf1-10GE1/0/3] quit
[*Leaf1] commit
```

Step 5 Configure VXLAN Layer 3 gateways on Spine1 and Spine2.

# Configure a Layer 3 VXLAN gateway on Spine1. The configuration on Spine2 is similar to that on Spine1, and is not mentioned here.

```
[~Spine1] interface vbdif 10
[*Spine1-Vbdif10] ip address 192.168.10.1 24
[*Spine1-Vbdif10] mac-address 0000-5e00-0101
[*Spine1-Vbdif10] quit
[*Spine1] interface vbdif 20
[*Spine1-Vbdif20] ip address 192.168.20.1 24
[*Spine1-Vbdif20] mac-address 0000-5e00-0102
[*Spine1-Vbdif20] quit
[*Spine1] interface vbdif 30
[*Spine1-Vbdif30] ip address 192.168.30.1 24
[*Spine1-Vbdif30] mac-address 0000-5e00-0103
[*Spine1-Vbdif30] quit
[*Spine1] commit
```

NOTE

Because Spine1 and Spine2 work as all-active gateways, you need to ensure that the IP addresses of NVE interfaces, as well as the IP addresses and MAC addresses of VBDIF interfaces on the two devices are the same.

Step 6 Configure all-active gateways on Spine1 and Spine2.

# Configure Spine1. The configuration on Spine2 is similar to that on Spine1, and is not mentioned here.

```
[~Spine1] dfs-group 1
[*Spine1-dfs-group-1] source ip 10.10.10.10
[*Spine1-dfs-group-1] active-active-gateway
[*Spine1-dfs-group-1-active-active-gateway] peer 10.10.10.20
```

```
[*Spine1-dfs-group-1-active-active-gateway] quit
[*Spine1-dfs-group-1] quit
[*Spine1] commit
```

Step 7 Verify the configuration.

After the configuration is complete, run the **display dfs-group 1 active-active-gateway** command on Spine1 and Spine2. The command output shows information about the all-active gateways in the DFS group. The display on Spine1 is used as an example.

```
[~Spine1] display dfs-group 1 active-active-gateway
A:Active     I:Inactive
-------------------------------------------------------------------
Peer             System name         State         Duration
10.10.10.20      Spine2              A             0:0:8
```

----End

Configuration Files

- Configuration file of Leaf1

```
#
sysname Leaf1
#
assign forward nvo3 acl extend enable
#
vlan batch 10
#
ip tunnel mode vxlan
#
bridge-domain 10
 l2 binding vlan 10
 vxlan vni 5000
#
interface 10GE1/0/1
 undo portswitch
 ip address 10.1.1.2 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 10.2.1.2 255.255.255.0
#
interface 10GE1/0/3
 port link-type trunk
 undo port trunk allow-pass vlan 1
 port trunk allow-pass vlan 10
#
interface LoopBack1
 ip address 10.10.10.3 255.255.255.255
#
interface Nve1
 source 10.10.10.3
 vni 5000 head-end peer-list 10.10.10.1
#
ospf 1
 area 0.0.0.0
  network 10.1.1.0 0.0.0.255
  network 10.2.1.0 0.0.0.255
  network 10.10.10.3 0.0.0.0
#
return
```

- Configuration file of Leaf2

```
#
sysname Leaf2
#
assign forward nvo3 acl extend enable
```

```
#
vlan batch 20
#
ip tunnel mode vxlan
#
bridge-domain 20
 l2 binding vlan 20
 vxlan vni 5001
#
interface 10GE1/0/1
 undo portswitch
 ip address 10.3.1.2 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 10.4.1.2 255.255.255.0
#
interface 10GE1/0/3
 port link-type trunk
 undo port trunk allow-pass vlan 1
 port trunk allow-pass vlan 20
#
interface LoopBack1
 ip address 10.10.10.4 255.255.255.255
#
interface Nve1
 source 10.10.10.4
 vni 5001 head-end peer-list 10.10.10.1
#
ospf 1
 area 0.0.0.0
  network 10.3.1.0 0.0.0.255
  network 10.4.1.0 0.0.0.255
  network 10.10.10.4 0.0.0.0
#
return
```

- Configuration file of Leaf3

```
#
sysname Leaf3
#
assign forward nvo3 acl extend enable
#
vlan batch 30
#
ip tunnel mode vxlan
#
bridge-domain 30
 l2 binding vlan 30
 vxlan vni 5002
#
interface 10GE1/0/1
 undo portswitch
 ip address 10.5.1.2 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 10.6.1.2 255.255.255.0
#
interface 10GE1/0/3
 port link-type trunk
 undo port trunk allow-pass vlan 1
 port trunk allow-pass vlan 30
#
interface LoopBack1
 ip address 10.10.10.5 255.255.255.255
#
interface Nve1
 source 10.10.10.5
 vni 5002 head-end peer-list 10.10.10.1
```

```
#
ospf 1
 area 0.0.0.0
  network 10.5.1.0 0.0.0.255
  network 10.6.1.0 0.0.0.255
  network 10.10.10.5 0.0.0.0
#
return
```

- Configuration file of Spine1

```
#
sysname Spine1
#
assign forward nvo3 acl extend enable
#
dfs-group 1
 source ip 10.10.10.10
 #
 active-active-gateway
  peer 10.10.10.20
#
ip tunnel mode vxlan
#
bridge-domain 10
 vxlan vni 5000
#
bridge-domain 20
 vxlan vni 5001
#
bridge-domain 30
 vxlan vni 5002
#
interface Vbdif10
 ip address 192.168.10.1 255.255.255.0
 mac-address 0000-5e00-0101
#
interface Vbdif20
 ip address 192.168.20.1 255.255.255.0
 mac-address 0000-5e00-0102
#
interface Vbdif30
 ip address 192.168.30.1 255.255.255.0
 mac-address 0000-5e00-0103
#
interface 10GE1/0/1
 undo portswitch
 ip address 10.1.1.1 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 10.3.1.1 255.255.255.0
#
interface 10GE1/0/3
 undo portswitch
 ip address 10.5.1.1 255.255.255.0
#
interface LoopBack1
 ip address 10.10.10.1 255.255.255.255
#
interface LoopBack2
 ip address 10.10.10.10 255.255.255.255
#
interface Nve1
 source 10.10.10.1
 vni 5000 head-end peer-list 10.10.10.3
 vni 5001 head-end peer-list 10.10.10.4
 vni 5002 head-end peer-list 10.10.10.5
#
ospf 1
 area 0.0.0.0
```

```
  network 10.1.1.0 0.0.0.255
  network 10.3.1.0 0.0.0.255
  network 10.5.1.0 0.0.0.255
  network 10.10.10.1 0.0.0.0
  network 10.10.10.10 0.0.0.0
#
return
```

- Configuration file of Spine2

```
#
sysname Spine2
#
assign forward nvo3 acl extend enable
#
dfs-group 1
 source ip 10.10.10.20
 #
 active-active-gateway
  peer 10.10.10.10
#
ip tunnel mode vxlan
#
bridge-domain 10
 vxlan vni 5000
#
bridge-domain 20
 vxlan vni 5001
#
bridge-domain 30
 vxlan vni 5002
#
interface Vbdif10
 ip address 192.168.10.1 255.255.255.0
 mac-address 0000-5e00-0101
#
interface Vbdif20
 ip address 192.168.20.1 255.255.255.0
 mac-address 0000-5e00-0102
#
interface Vbdif30
 ip address 192.168.30.1 255.255.255.0
 mac-address 0000-5e00-0103
#
interface 10GE1/0/1
 undo portswitch
 ip address 10.2.1.1 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 10.4.1.1 255.255.255.0
#
interface 10GE1/0/3
 undo portswitch
 ip address 10.6.1.1 255.255.255.0
#
interface LoopBack1
 ip address 10.10.10.1 255.255.255.255
#
interface LoopBack2
 ip address 10.10.10.20 255.255.255.255
#
interface Nve1
 source 10.10.10.1
 vni 5000 head-end peer-list 10.10.10.3
 vni 5001 head-end peer-list 10.10.10.4
 vni 5002 head-end peer-list 10.10.10.5
#
ospf 1
 area 0.0.0.0
  network 10.2.1.0 0.0.0.255
  network 10.4.1.0 0.0.0.255
  network 10.6.1.0 0.0.0.255
  network 10.10.10.1 0.0.0.0
  network 10.10.10.20 0.0.0.0
#
return
```

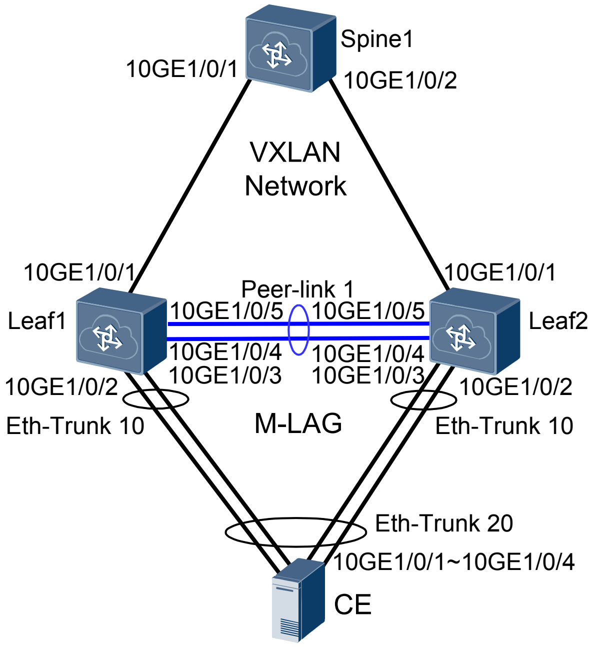
12.6 Example for Configuring Dual-Active VXLAN Access

Networking Requirements

The network shown in **Figure 12-6** has the following requirements when servers access the VXLAN network:

- To ensure high reliability, the server is dual-homed to two leaf devices. When one access link fails, traffic can be rapidly switched to the other link.
- To improve bandwidth utilization, two links are in active state simultaneously to load balance traffic.

Figure 12-6 VXLAN dual-active access networking



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure a routing protocol on Leaf1, Leaf2, and Spine1 to ensure Layer 3 network connectivity.
2. Configure basic VXLAN functions on Leaf1, Leaf2, and Spine1 to ensure VXLAN network connectivity.
3. Create Eth-Trunks.
4. Configure Leaf1 and Leaf2 as root bridges and configure the same bridge ID for them.
5. Configure M-LAG.

- Associate a DFS group with VXLAN on Leaf1 and Leaf2.
- Configure a link between Leaf1 and Leaf2 as the peer link.
- Bind the user-side Eth-Trunk to the DFS group on Leaf1 and Leaf2.

NOTE

If the link through which Leaf1 is uplink connected to the VXLAN network fails, Leaf1 discards all received user traffic because no uplink outbound interface is available. You can configure a monitor-link to associate the uplink and downlink interfaces of Leaf1. When the uplink outbound interface of Leaf1 becomes Down, the downlink interface also becomes Down. Then user traffic will not be forwarded or discarded by Leaf1. For details about the monitor-link configuration, see Configuring the Uplink and Downlink Interfaces in a Monitor Link Group.

Data Plan

To complete the configuration, you need the following data:

- Interface IP addresses for device interconnection
- Routing protocol: OSPF
- VLAN ID to which VMs belong: VLAN 10
- BD ID: BD 10
- VNI ID: VNI 5010

Procedure

Step 1 Configure a routing protocol.

# Configure Leaf1. The configurations on Spine1 and Leaf2 are similar to that on Leaf1, and are not mentioned here. When OSPF is used, the devices advertise the 32-bit loopback IP addresses.

```
<HUAWEI> system-view
[~HUAWEI] sysname Leaf1
[*HUAWEI] commit
[~Leaf1] interface loopback 1
[*Leaf1-LoopBack1] ip address 10.2.2.2 32
[*Leaf1-LoopBack1] quit
[*Leaf1] interface loopback 2
[*Leaf1-LoopBack2] ip address 10.3.3.3 32
[*Leaf1-LoopBack2] quit
[*Leaf1] interface 10ge 1/0/1
[*Leaf1-10GE1/0/1] undo portswitch
[*Leaf1-10GE1/0/1] ip address 192.168.1.1 24
[*Leaf1-10GE1/0/1] quit
[*Leaf1] ospf
[*Leaf1-ospf-1] area 0
[*Leaf1-ospf-1-area-0.0.0.0] network 10.2.2.2 0.0.0.0
[*Leaf1-ospf-1-area-0.0.0.0] network 10.3.3.3 0.0.0.0
[*Leaf1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[*Leaf1-ospf-1-area-0.0.0.0] quit
[*Leaf1-ospf-1] quit
[*Leaf1] commit
```

After OSPF is configured, the devices can learn the loopback IP address of each other and successfully ping each other. The following shows the ping result from Leaf1 to Spine1.

```
[~Leaf1] ping 10.1.1.1
  PING 10.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=254 time=5 ms
    Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=254 time=2 ms
    Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=254 time=2 ms
    Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=254 time=3 ms
    Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=254 time=3 ms

  --- 10.1.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 2/3/5 ms
```

Step 2 Configure the VXLAN tunnel mode and enable the VXLAN ACL extension function.

# Configure Leaf1. The configurations on Leaf2 and Spine1 are similar to that on Leaf1, and are not mentioned here.

```
[~Leaf1] ip tunnel mode vxlan
[*Leaf1] assign forward nvo3 acl extend enable
[*Leaf1] commit
```

NOTE

After modifying the VXLAN tunnel mode or enabling the VXLAN ACL extension function, you need to save the configuration and restart the device to make the configuration take effect. You can restart the device immediately or after completing all the configurations.

Step 3 Configure VXLAN tunnels between Leaf1, Leaf2, and Spine1.

# Configure Leaf1. The configuration on Leaf2 is similar to that on Leaf1, and is not mentioned here.

```
[~Leaf1] bridge-domain 10
[*Leaf1-bd10] vxlan vni 5010
[*Leaf1-bd10] quit
[*Leaf1] interface nve1
[*Leaf1-Nve1] source 10.2.2.2
[*Leaf1-Nve1] vni 5010 head-end peer-list 10.1.1.1
[*Leaf1-Nve1] quit
[*Leaf1] commit
```

NOTE

Because Leaf1 and Leaf2 work as dual-active access, you need to ensure that the IP addresses of NVE interfaces on the two devices are the same.

# Configure Spine1.

```
[~Spine1] bridge-domain 10
[*Spine1-bd10] vxlan vni 5010
[*Spine1-bd10] quit
[*Spine1] interface nve1
[*Spine1-Nve1] source 10.1.1.1
[*Spine1-Nve1] vni 5010 head-end peer-list 10.2.2.2
[*Spine1-Nve1] quit
[*Spine1] commit
```

After the configuration is complete, run the **display vxlan vni** command on Spine1. The command output shows that the VNI status is **up**. Run the **display vxlan tunnel** command, and you can see VXLAN tunnel information.

```
[~Spine1] display vxlan vni
Number of vxlan vni : 1
VNI            BD-ID            State
---------------------------------------
5010           10               up
[~Spine1] display vxlan tunnel
Number of vxlan tunnel : 1
Tunnel ID   Source           Destination        State  Type
--------------------------------------------------------------
4026531841  10.1.1.1          10.2.2.2          up     static
```

Step 4 Create an Eth-Trunk and add physical Ethernet interfaces to the Eth-Trunk.

An uplink interface of a server connected to a switch needs to be bound to an aggregated link and the link aggregation mode of the server needs to be consistent with that of the switch.

# Create an Eth-Trunk in LACP mode on Leaf1 and add physical Ethernet interfaces to the Eth-Trunk. The configuration on Leaf2 is similar to that on Leaf1, and is not mentioned here.

```
[~Leaf1] interface eth-trunk 1
[*Leaf1-Eth-Trunk1] mode lacp-static
[*Leaf1-Eth-Trunk1] trunkport 10ge 1/0/4 to 1/0/5
[*Leaf1-Eth-Trunk1] quit
[*Leaf1] interface eth-trunk 10
[*Leaf1-Eth-Trunk10] mode lacp-dynamic
[*Leaf1-Eth-Trunk10] trunkport 10ge 1/0/2 to 1/0/3
[*Leaf1-Eth-Trunk10] quit
[*Leaf1] commit
```

Step 5 Configure Leaf1 and Leaf2 as root bridges and configure the same bridge ID for them.

NOTE

If the two devices that constitute an M-LAG connect to downstream switching devices, you must configure root protection.

# Configure Leaf1.

```
[~Leaf1] stp root primary
[*Leaf1] stp bridge-address 39-39-39
[*Leaf1] interface eth-trunk 10
[*Leaf1-Eth-Trunk10] stp edged-port enable
[*Leaf1-Eth-Trunk10] commit
[~Leaf1-Eth-Trunk10] quit
```

# Configure Leaf2.

```
[~Leaf2] stp root primary
[*Leaf2] stp bridge-address 39-39-39
[*Leaf2] interface eth-trunk 10
[*Leaf2-Eth-Trunk10] stp edged-port enable
[*Leaf2-Eth-Trunk10] commit
[~Leaf2-Eth-Trunk10] quit
```

Step 6 Configure a DFS group on Leaf1 and Leaf2 respectively.

# Configure Leaf1. The configuration on Leaf2 is similar to that on Leaf1, and is not mentioned here.

```
[~Leaf1] dfs-group 1
[*Leaf1-dfs-group-1] source ip 10.3.3.3
[*Leaf1-dfs-group-1] quit
[*Leaf1] commit
```

Step 7 Configure a link between Leaf1 and Leaf2 as the peer link.

# Configure Leaf1. The configuration on Leaf2 is similar to that on Leaf1, and is not mentioned here.

```
[~Leaf1] interface eth-trunk 1
[~Leaf1-Eth-Trunk1] undo stp enable
[*Leaf1-Eth-Trunk1] peer-link 1
[*Leaf1-Eth-Trunk1] quit
[*Leaf1] commit
```

Step 8 Bind the user-side Eth-Trunk to the DFS group on Leaf1 and Leaf2.

# Configure Leaf1. The configuration on Leaf2 is similar to that on Leaf1, and is not mentioned here.

```
[~Leaf1] interface eth-trunk 10
[~Leaf1-Eth-Trunk10] dfs-group 1 m-lag 1
[*Leaf1-Eth-Trunk10] quit
[*Leaf1] commit
```

Step 9 Configure service access points on Leaf1 and Leaf2.

# Configure Leaf1. The configuration on Leaf2 is similar to that on Leaf1, and is not mentioned here.

```
[~Leaf1] vlan 10
[*Leaf1-vlan10] quit
[*Leaf1] bridge-domain 10
[*Leaf1-bd10] l2 binding vlan 10
[*Leaf1-bd10] quit
[*Leaf1] interface eth-trunk 10
[*Leaf1-Eth-Trunk10] port link-type trunk
[*Leaf1-Eth-Trunk10] undo port trunk allow-pass vlan 1
[*Leaf1-Eth-Trunk10] port trunk allow-pass vlan 10
[*Leaf1-Eth-Trunk10] quit
[*Leaf1] commit
```

Step 10 Verify the configuration.

Run the **display dfs-group 1 m-lag** command to check M-LAG information.

```
[~Leaf1] display dfs-group 1 m-lag
*                : Local node
Heart beat state : OK
Node 1 *
  Dfs-Group ID   : 1
  Priority       : 100
  Address        : ip address 10.3.3.3
  State          : Master
  Causation      : -
  System ID      : 0025-9e95-7c11
  SysName        : Leaf1
  Version        : V100R006C00
  Device Type    : CE12800
Node 2
  Dfs-Group ID   : 1
  Priority       : 100
  Address        : ip address 10.4.4.4
  State          : Backup
  Causation      : -
  System ID      : 0025-9e95-7c31
  SysName        : Leaf2
  Version        : V100R006C00
  Device Type    : CE12800
```

Check M-LAG information on Leaf1.

```
[~Leaf1] display dfs-group 1 node 1 m-lag brief
* - Local node

M-Lag ID     Interface      Port State    Status
       1     Eth-Trunk 10   Up            active(*)-active
```

Check M-LAG information on Leaf2.

```
[~Leaf2] display dfs-group 1 node 2 m-lag brief
* - Local node

M-Lag ID     Interface      Port State    Status
       1     Eth-Trunk 10   Up            active-active(*)
```

Run the **display bridge-domain 10 verbose** command on Leaf1 and Leaf2 to view BD information. The display on Leaf1 is taken as an example.

```
[~Leaf1] display bridge-domain 10 verbose
  Bridge-domain ID        : 10
  Description             :
  State                   : Up
  MAC Learning            : Enable
  Statistics              : Disable
  Broadcast               : Forward
  Unknown-unicast         : Forward
  Unknown-multicast       : Forward
  Split-horizon           : Disable

  ----------------
Interface                                State
  Eth-Trunk1.5010                        up
  10GE1/0/3.1                            up
```

----End

Configuration Files

- Configuration file of Leaf1

```
#
sysname Leaf1
#
assign forward nvo3 acl extend enable
#
dfs-group 1
 source ip 10.3.3.3
#
vlan batch 10
#
ip tunnel mode vxlan
#
stp bridge-address 0039-0039-0039
stp instance 0 root primary
#
bridge-domain 10
 l2 binding vlan 10
 vxlan vni 5010
#
interface Eth-Trunk1
 stp disable
 mode lacp-static
 peer-link 1
#
interface Eth-Trunk10
 stp edged-port enable
 mode lacp-dynamic
 dfs-group 1 m-lag 1
#
interface Eth-Trunk10
 port link-type trunk
 undo port trunk allow-pass vlan 1
 port trunk allow-pass vlan 10
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.1.1 255.255.255.0
#
interface 10GE1/0/2
 eth-trunk 10
#
interface 10GE1/0/3
 eth-trunk 10
#
interface 10GE1/0/4
 eth-trunk 1
#
```

```
interface 10GE1/0/5
 eth-trunk 1
#
interface LoopBack1
 ip address 10.2.2.2 255.255.255.255
#
interface LoopBack2
 ip address 10.3.3.3 255.255.255.255
#
interface Nve1
 source 10.2.2.2
 vni 5010 head-end peer-list 10.1.1.1
#
ospf 1
 area 0.0.0.0
  network 10.2.2.2 0.0.0.0
  network 10.3.3.3 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

- Configuration file of Leaf2

```
#
sysname Leaf2
#
assign forward nvo3 acl extend enable
#
dfs-group 1
 source ip 10.4.4.4
#
vlan batch 10
#
ip tunnel mode vxlan
#
stp bridge-address 0039-0039-0039
stp instance 0 root primary
#
bridge-domain 10
 l2 binding vlan 10
 vxlan vni 5010
#
interface Eth-Trunk1
 stp disable
 mode lacp-static
 peer-link 1
#
interface Eth-Trunk10
 stp edged-port enable
 mode lacp-dynamic
 dfs-group 1 m-lag 1
#
interface Eth-Trunk10
 port link-type trunk
 undo port trunk allow-pass vlan 1
 port trunk allow-pass vlan 10
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.2.1 255.255.255.0
#
interface 10GE1/0/2
 eth-trunk 10
#
interface 10GE1/0/3
 eth-trunk 10
#
interface 10GE1/0/4
 eth-trunk 1
#
interface 10GE1/0/5
```

```
 eth-trunk 1
#
interface LoopBack1
 ip address 10.2.2.2 255.255.255.255
#
interface LoopBack2
 ip address 10.4.4.4 255.255.255.255
#
interface Nve1
 source 10.2.2.2
 vni 5010 head-end peer-list 10.1.1.1
#
ospf 1
 area 0.0.0.0
  network 10.2.2.2 0.0.0.0
  network 10.4.4.4 0.0.0.0
  network 192.168.2.0 0.0.0.255
#
return
```

- Configuration file of Spine1

```
#
sysname Spine1
#
assign forward nvo3 acl extend enable
#
ip tunnel mode vxlan
#
bridge-domain 10
 vxlan vni 5010
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 192.168.2.2 255.255.255.0
#
interface LoopBack1
 ip address 10.1.1.1 255.255.255.255
#
interface Nve1
 source 10.1.1.1
 vni 5010 head-end peer-list 10.2.2.2
#
ospf 1
 area 0.0.0.0
  network 10.1.1.1 0.0.0.0
  network 192.168.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
#
return
```

13 References

The following table lists the references.

Document No.	Document Name	Remarks
draft-ietf-bess-evpn-prefix-advertisement-01	IP Prefix Advertisement in EVPN	-
draft-ietf-bess-evpn-inter-subnet-forwarding-00	Integrated Routing and Bridging in EVPN	-
draft-ietf-NVo3-framework-04	Framework for DC Network Virtualization	-
draft-ietf-NVo3-dataplane-requirements-02	NVo3 Data Plane Requirements	-
draft-mahalingam-dutt-dcops-vxlan-06	A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks	-
RFC 7432	BGP MPLS-Based Ethernet VPN	-