Configuration Guide Manual VXLAN

QSW-6900

Оглавление

1. CONFIGURING VXLAN

1.1. Overview

Virtual Extensible Local Area Network (VXLAN) is a virtual Ethernet based on the physical IP (overlay) network. It is a technology that encapsulates layer 2 (L2) Ethernet frames within layer 3 User Datagram Protocol (UDP) packets.

VXLAN has a 24-bit VXLAN network identifier (VNI). It allows users to create up to 16,000,000 isolated virtual networks to meet the requirements of multi-tenant environments and scale expansion, far surpassing the widely used Virtual Local Area Network (VLAN) technology that is limited to 4,000 isolated networks. VXLAN uses the IP multicast method to encapsulate multicast, broadcast, and unknown unicast packets, effectively controlling the broadcast domain in multi-tenant environments.

With the transformation of data centers, more and more virtual machines are deployed. In addition, as virtual machines must be migrated in L2 environments, scales of L2 networks increase. VXLAN can extend L2 networks over layer 3 (L3) networks, so that virtual machines can be moved to L3 networks interconnected to L2 networks without changing the IP addresses and MAC addresses, thereby ensuring service continuity.

Protocols and Standards

- RFC7348: Virtual eXtensible Local Area Network (VXLAN) -- A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks

1.2. Applications

Application	Description
EVPN-based Multi-tenant Centralized Deployment	Applicable to the centralized deployment scenario with Ethernet virtual private network (EVPN) enabled.
EVPN-based Multi-tenant Distributed Deployment	Applicable to the distributed deployment scenario with EVPN enabled.
EVPN-based Single-tenant VXLAN Routing Deployment	Applicable to the VXLAN routing deployment scenario with a single tenant.
EVPN-based Multi-tenant VXLAN Route Deployment	Applicable to the VXLAN route deployment scenario with multiple tenants.
SDN Controller–based Centralized All-active Anycast Gateway Deployment	Applicable to the scenario for deploying all-active anycast gateways based on the software-defined networking (SDN) controller in a centralized manner data centers.
Deployment of an EVPN Distributed Network to Be	Applicable to the deployment scenario in which dynamic and

Application	Description
Compatible with Non-EVPN VTEP Devices	static tunnels coexist.
Deployment of L2 Subinterfaces to Access a VXLAN	Applicable to the deployment scenario in which hosts access VXLANs through L2 subinterfaces.
VNI Mapping–based Data Center Interconnection Deployment	Applicable to the scenario in which VXLANs across different data centers are interconnected using the VNI mapping technology.

1.2.1. EVPN-based Multi-tenant Centralized Deployment

Scenario

VPN routing and forwarding (VRF) networks are usually allocated to different tenants to support the multi-tenant application in a data center. Multiple VXLANs can be assigned to each tenant. VXLANs of the same tenant can be mutually accessed through the L3 router, while VXLANs of different tenants cannot be mutually accessed, as shown in Figure 2-1.

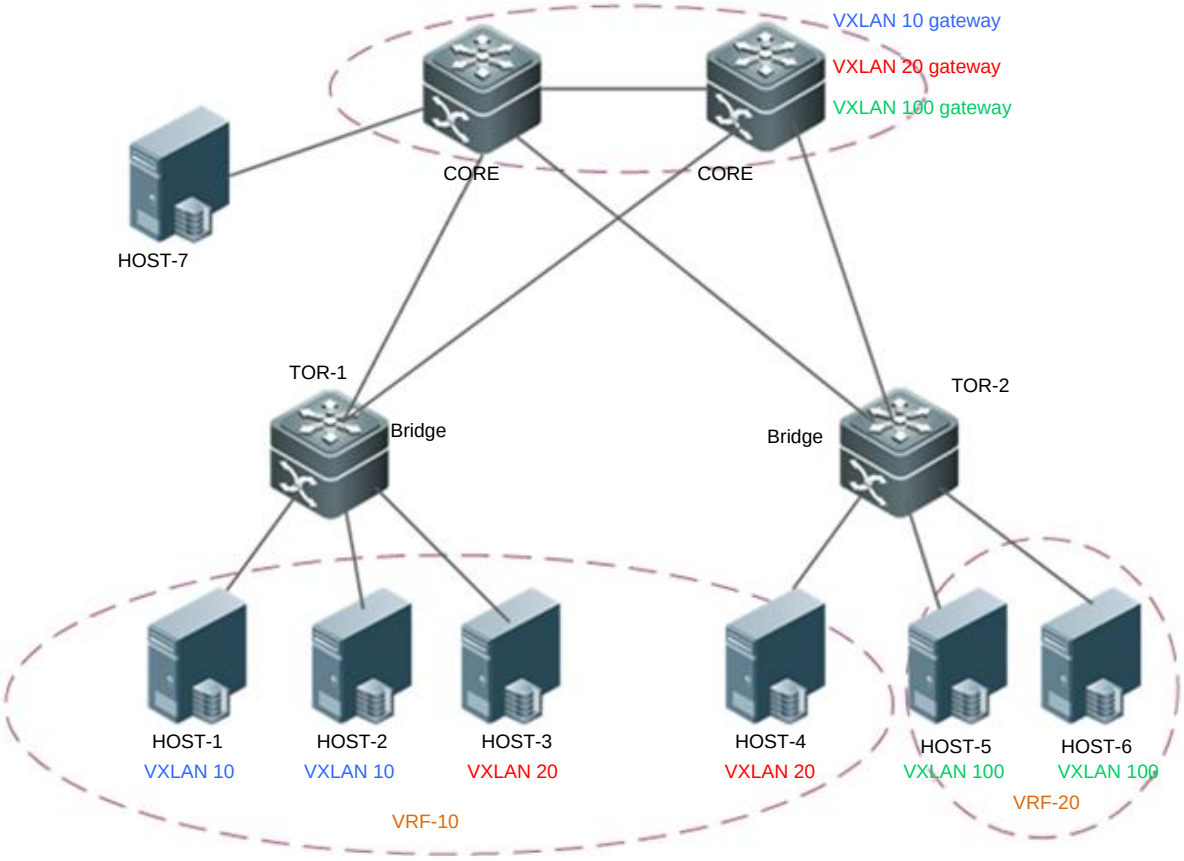
Tenant A rents VRF-10, which includes VXLAN 10 and VXLAN 20. Servers HOST-1 and HOST-2 belong to VXLAN 10 and Servers HOST-3 and HOST-4 belong to VXLAN 20.

Tenant B rents VRF-20, which includes VXLAN 100. Servers HOST-5 and HOST-6 belong to VXLAN 100.

The networks of Tenant A and Tenant B are isolated from each other.

The entire network is formed by a Border Gateway Protocol (BGP) network and includes CORE and TOR switches. The BGP neighbor relationship is formed between every two devices and the BGP-EVPN protocol family is supported. All VXLAN gateways on the network are deployed in the core switches in a centralized manner.

Figure 1-1



- Packets between HOST-1 and HOST-2 are forwarded through TOR-1 at L2 within the VXLAN.
- Packets between HOST-3 and HOST-4 are forwarded through TOR-1 > CORE > TOR-2 at L2 within the VXLAN.
- Packets between HOST-5 and HOST-6 are forwarded through TOR-2 at L2 within the VXLAN.
- Packets between VXLAN 10 and VXLAN 20 are forwarded through TOR-1 > CORE > TOR-2 at L3 across the VXLANs.
- VRF-10 and VRF-20 cannot communicate with each other.

Remarks:

CORE indicates a core switch that supports the VXLAN function.

When centralized all-active anycast gateways are deployed, multiple core gateways exist and the VXLAN gateways deployed on the core gateways are the same.

TOR1 and TOR2 are access switches that support the VXLAN function.

HOST-1, HOST-2, HOST-3, HOST-4, HOST-5, and HOST-6 are servers in the data center.

Deployment

- Configure an Internet Protocol version 4 (IPv4) unicast routing protocol, for example, the Open Shortest Path First (OSPF) protocol, on the switches to ensure that unicast routes are reachable.
- Configure the BGP routing protocol (supporting EVPN) on the switches to establish neighbor relationships between each other.

- Deploy the VXLAN gateway on the core switches.
- Deploy the VXLAN bridge on the TOR switches.

1.2.2. EVPN-based Multi-tenant Distributed Deployment

Scenario

The EVPN-based multi-tenant distributed deployment applies to data center networks that support multiple tenants. The difference between this deployment and the EVPN-based multi-tenant centralized deployment described in section 2.2.1 lies in that: On the distributed deployment network, gateways are deployed on the TOR switches, as shown in Figure 2-2.

Tenant A rents VRF-10, which includes VXLAN 10 and VXLAN 20.

Tenant B rents VRF-20, which includes VXLAN 100.

The networks of Tenant A and Tenant B are isolated from each other.

The entire network is formed by a BGP network and includes CORE and TOR switches. The BGP neighbor relationship is formed between every two devices and the BGP-EVPN protocol family is supported.

VXLAN gateways are deployed on TOR switches on the network. Anycast gateways can be deployed so that the IP addresses and MAC addresses of all gateways on the network are kept consistent. In this way, the gateway configuration does not need to be modified no matter which TOR switch a virtual machine of a customer is migrated to.

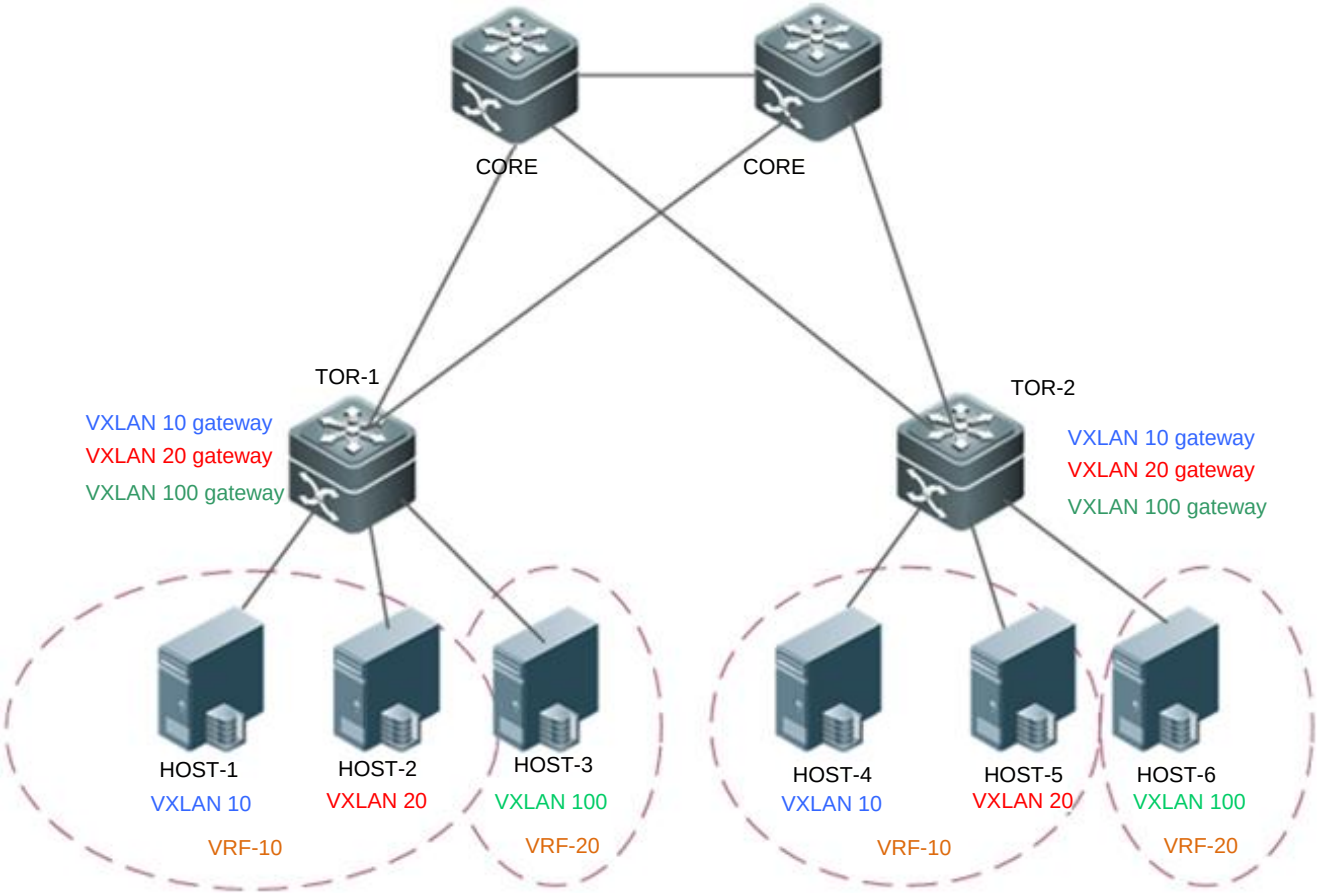
VXLANs are unnecessarily deployed on the core switches.

ARP suppression can be configured on TOR switches to curb flooding of ARP packets, and the TOR switches respond to ARP requests from hosts as a proxy.

The ARP proxy function can be enabled on the TOR switches for all or some VXLANs. In this way, L2 traffic in VXLANs is isolated and server communication traffic in the same VXLAN is forwarded at L3 rather than at L2.

ND suppression can be configured on TOR switches to curb flooding of IPv6 ND protocol packets, and the TOR switches respond to IPv6 NS multicast packets from hosts as a proxy.

Figure 1-2



- Packets between HOST-1 and HOST-4 are forwarded through TOR-1 > TOR-2 at L2 within the VXLAN.
- Packets between HOST-1 and HOST-2 are forwarded through TOR-1 at L3 across the VXLANs.
- Packets between HOST-1 and HOST-5 are forwarded through TOR-1 > TOR-2 at L3 across the VXLANs.
- VRF-10 and VRF-20 cannot communicate with each other.
- If the ARP proxy function is configured on VXLAN 10, packets between HOST-1 and HOST-4 are forwarded through TOR-1 > TOR-2 at L3 within the VXLAN.

Remarks:

CORE indicates a core switch that supports the BGP-EVPN function.

TOR1 and TOR2 are access switches that support the VXLAN function.

HOST-1, HOST-2, HOST-3, HOST-4, HOST-5, and HOST-6 are servers in the data center.

Deployment

- Configure an IPv4 unicast routing protocol, for example, the OSPF protocol, on the switches to ensure that unicast routes are reachable.
- Configure the BGP routing protocol (supporting EVPN) on the switches to establish neighbor relationships between each other.
- Deploy the VXLAN bridge on the core switches if required.
- Deploy the VXLAN gateway on the TOR switches.

- (Optional) Deploy ARP suppression on the TOR switches.
- (Optional) Deploy ARP proxy on the TOR switches.
- (Optional) Deploy IPv6 ND suppressionon the TOR switches.
- (Optional) Deploy the EVPN protocol packet control function on the TOR switches to reduce the traffic of EVPN packets.

1.2.3. EVPN-based Single-tenant VXLAN Routing Deployment

Scenario

Single-tenant VXLAN route deployment is shown in Figure 2-3.

In this scenario, only the VRF-10 is deployed, which includes VXLAN 10 and VXLAN 20.

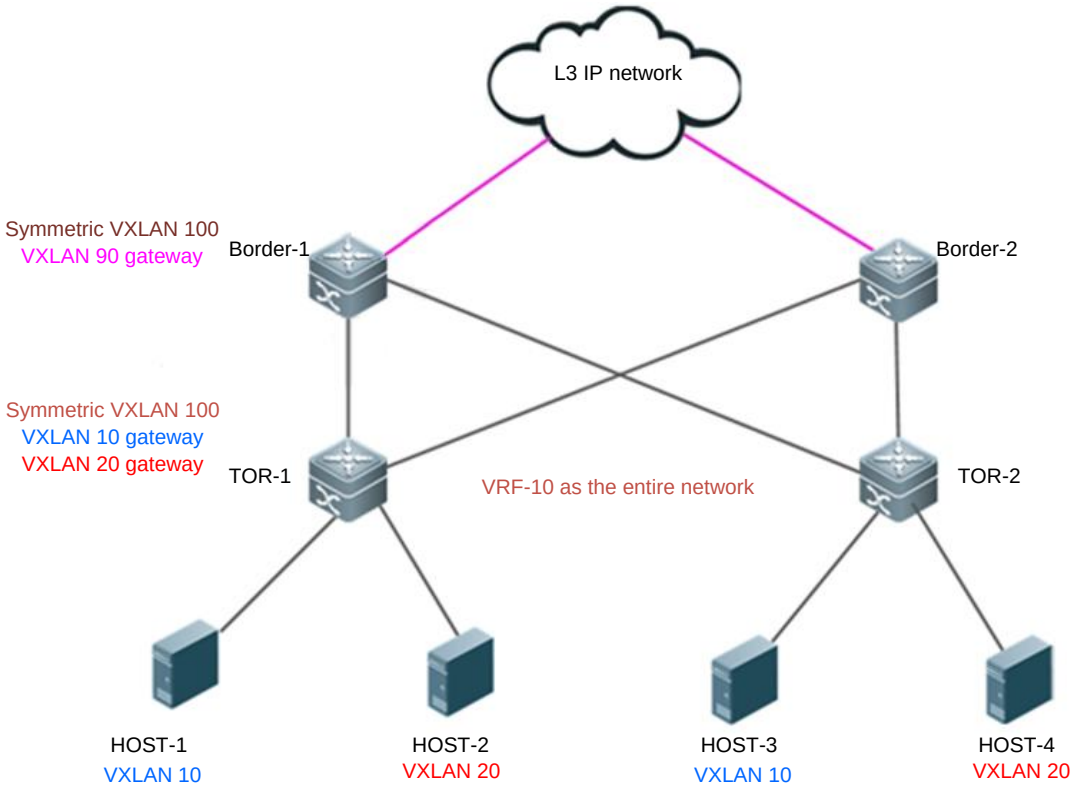
The border devices are connected to the external network. These devices are deployed in VRF-10 (including VXLAN 90) and interconnect with the external network at L3 via the overlay router interface.

The entire network is formed by a BGP network and includes TOR and border devices. The BGP neighbor relationship is formed between every two devices (except between Border-1 and Border-2) and the BGP-EVPN protocol family is supported.

The TOR and border devices must use a symmetric VXLAN (VXLAN 100) for interconnection with each other. The border devices import network routes to the TOR switches through the symmetric VXLAN.

VXLAN gateways are deployed on TOR switches on the network. Anycast gateways can be deployed so that the IP addresses and MAC addresses of all gateways on the network are kept consistent. In this way, the gateway configuration does not need to be modified no matter which TOR switch a virtual machine of a customer is migrated to.

Figure 1-3



- Packets between HOST-1 and HOST-3 are forwarded through TOR-1 > TOR-2 at L2 within the VXLAN.
- Packets between HOST-1 and HOST-2 are forwarded through TOR-1 at L3 across the VXLANs.
- To access the external network, HOST-1 forwards packets to the border device through TOR1 at L3 across the VXLANs, and then the border device forwards the packets to the external network at L3.

Deployment

- Configure an IPv4 unicast routing protocol, for example, the OSPF protocol, on the switches to ensure that unicast routes are reachable.
- Configure the BGP routing protocol (supporting EVPN) on the switches to establish neighbor relationships between each other (except between the border devices).
- Deploy the VXLAN on the border devices for L3 interconnection with the external network.
- Deploy the VXLAN gateway on the TOR switches.

1.2.4. EVPN-based Multi-tenant VXLAN Route Deployment

Scenario

VRF networks are usually allocated to different tenants to support the multi-tenant application in a data center. Multiple VXLANs can be assigned to each tenant. VXLANs of the same tenant can be mutually accessed through the L3 router, while VXLANs of different tenants cannot be mutually accessed, as shown in Figure 2-4.

Tenant A rents VRF-10, which includes VXLAN 10 and VXLAN 20.

Tenant B rents VRF-20, which includes VXLAN 30.

The border devices are connected to the external network. These devices are deployed in VRF-30 (including VXLAN 90) and interconnect with the external network at L3 via the overlay router interface.

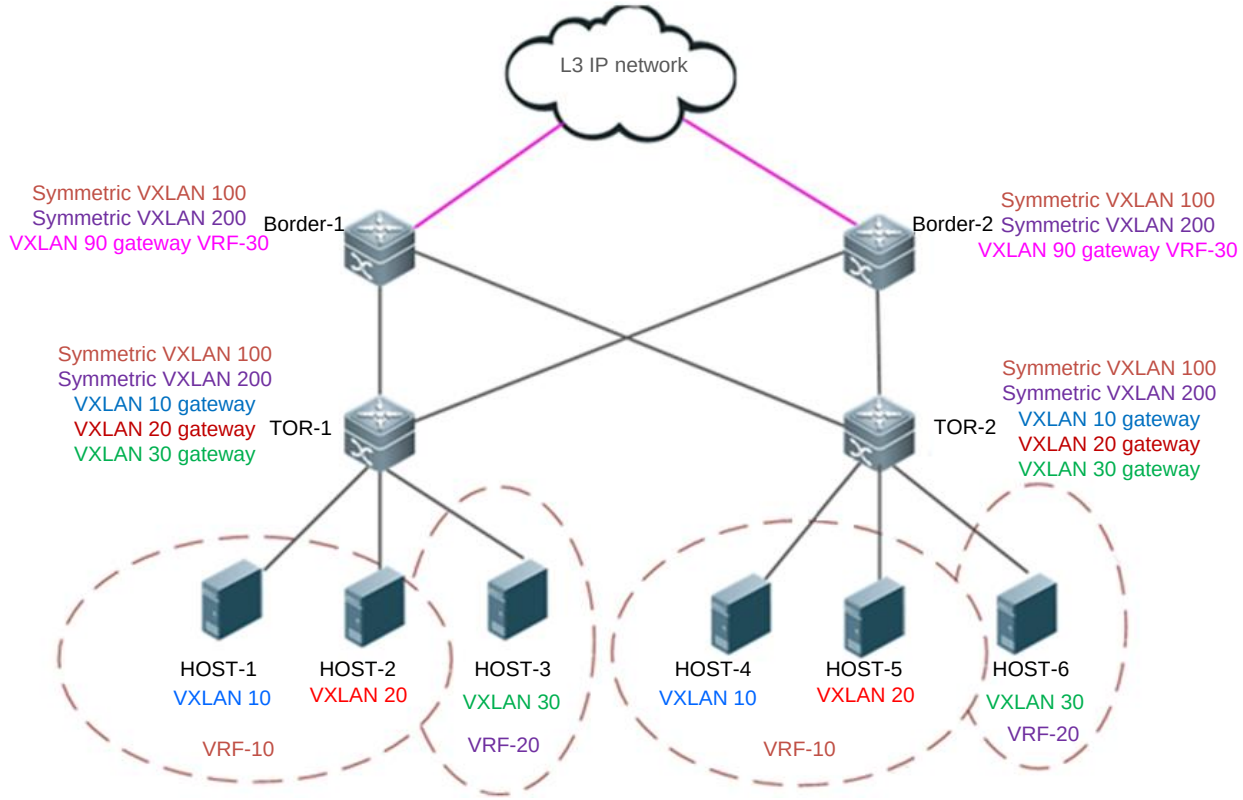
The networks of Tenant A and Tenant B are isolated from each other.

The entire network is formed by a BGP network and includes TOR and border devices. The BGP neighbor relationship is formed between every two devices (except between Border-1 and Border-2) and the BGP-EVPN protocol family is supported

The TOR and border devices must use a symmetric VXLAN (VXLAN 100 and VXLAN200) for interconnection with each other. The border devices import network routes to the TOR switches through the symmetric VXLAN.

VXLAN gateways are deployed on TOR switches on the network. Anycast gateways can be deployed so that the IP addresses and MAC addresses of all gateways on the network are kept consistent. In this way, the gateway configuration does not need to be modified no matter which TOR switch a virtual machine of a customer is migrated to.

Figure 1-4



- Packets between HOST-1 and HOST-4 are forwarded through TOR-1 > TOR-2 at L2 within the VXLAN.
- Packets between HOST-1 and HOST-2 are forwarded through TOR-1 at L3 across the VXLANs.
- To access the external network, HOST-1 forwards packets to the border device through TOR1 at L3 across the VXLANs, and then the border device forwards the packets to the external network at L3.

Deployment

- Configure an IPv4 unicast routing protocol, for example, the OSPF protocol, on the switches to ensure that unicast routes are reachable.
- Configure the BGP routing protocol (supporting EVPN) on the switches to establish neighbor relationships between each other (except between the border devices).
- Deploy the VXLAN on the border devices for L3 interconnection with the external network.
- Deploy the VXLAN gateway on the TOR switches.

1.2.5. SDN Controller–based Centralized All-active Anycast Gateway Deployment

Scenario

SND controller–based centralized all-active anycast gateway deployment applies to data center networks that support the control of an SND controller, as shown in Figure 2-5.

1. VXLAN overlay network topology:

In this scenario, the VXLAN overlay network is a two-layer structure including a core layer and an access layer.

1) TOR switches serve as VXLAN bridges to directly connect to servers (virtual machines).
2) Core switches serve as VXLAN gateways. Multiple all-active VXLAN physical gateways are deployed in a centralized manner. The physical gateways are in the all-active state. The anycast function is deployed on each physical gateway and the same IP address and MAC address are configured on all gateways to form a logical gateway. The fault of any particular physical gateway does not affect the normal operation of the logical gateway.
3) Virtual tunnel end points (VTEPs), including VXLAN bridges and VXLAN gateways, interconnect with each other through the L3 underlay network.
4) At the underlay layer, an L3 network connection is established between each TOR switch and each physical gateway. However, all physical gateways are virtualized into one logical gateway VTEP to communicate with the external network. Only one VXLAN tunnel is established between a TOR switch and the logical gateway VTEP. Traffic on the tunnel is balanced to multiple physical gateways via the equal-cost multi-path routing (ECMP).
5) No VXLAN tunnel or direct physical link is established between physical gateways.
6) On the server (virtual machine), only one logical gateway is visible.
7) VRF networks are allocated to multiple tenants. Networks of tenants are isolated from each other.

2. SDN controller management:

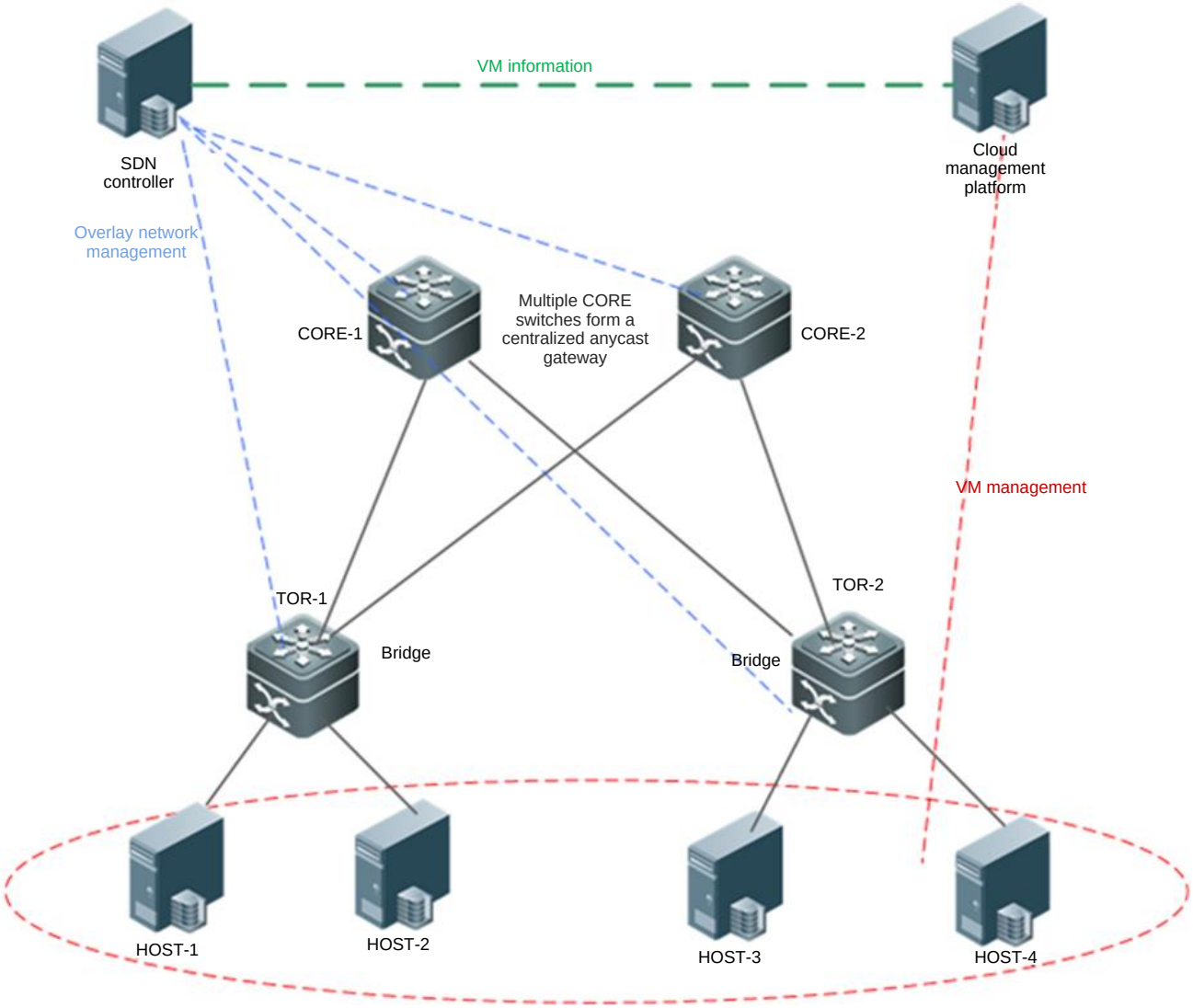
On the network, the administrator can configure the overlay network topology through the SDN controller and deliver the configurations to VTEPs. The administrator can also monitor the status of the overlay topology and network traffic through the SDN controller.

In addition, the administrator can manage the servers (virtual machines) on the entire network through the cloud management platform. The SDN controller can associate with the cloud management platform to acquire the configuration information (such as IP address and MAC address) of the virtual machine and deliver the configuration information to VTEPs. After the information is delivered, VXLAN forwarding entries are generated and synchronized on VTEPs.

3. VXLAN device automatic-learning capability

VTEPs can automatically learn the MAC address and the ARP routing table of the host if required, which can be used as an emergency solution for the case that the SDN controller fails. The automatic-learning function can be enabled according to the actual deployment.

Figure 1-5



Deployment

- Deploy the VXLAN bridging function on TOR switches and the VXLAN gateway function on the core switches.
- Configure an IPv4 unicast routing protocol, for example, the OSPF protocol, on all VTEPs (including the TOR and core switches) to ensure that unicast routes are reachable.
- On the core gateways, assign the gateway anycast IP addresses to different routing domains to avoid IP conflicts.

1.2.6. Deployment of an EVPN Distributed Network to Be Compatible with Non-EVPN VTEP Devices

Scenario

In a data center where an EVPN-based multi-tenant distributed network is deployed, one VTEP device that does not support the BGP-EVPN protocol (for example, a virtual switch supporting the VXLAN protocol) is connected. See the figure below.

BGP is deployed on the TOR and CORE switches. They mutually establish BGP neighbor relationships and support the EVPN routing protocol.

The VXLAN anycast gateways are deployed on the TOR switches and network-wide gateways share the same IP address and MAC address. TOR switches are directly connected to servers (virtual machines) and core switches are connected to external networks.

VXLANs do not need to be deployed on the core switches.

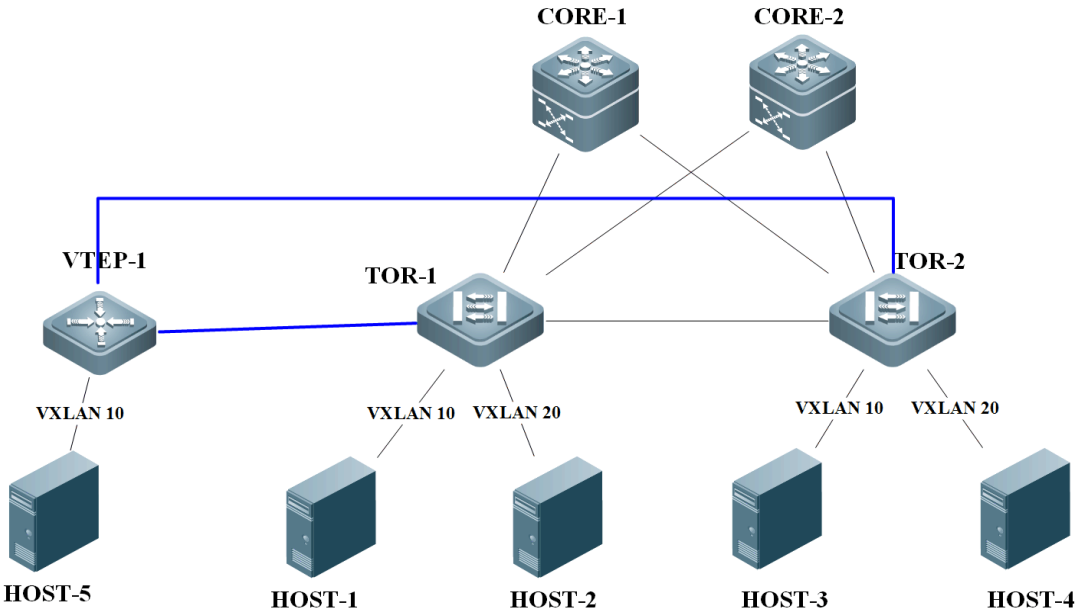
ARP suppression can be configured on TOR switches to curb flooding of ARP packets, and the TOR switches respond to ARP requests from hosts as a proxy.

The ARP proxy function can be enabled on the TOR switches for all or some VXLANs. In this way, L2 traffic in VXLANs is isolated and server communication traffic in the same VXLAN is forwarded at L3 rather than at L2.

ND suppression can be configured on TOR switches to curb flooding of IPv6 ND protocol packets, and the TOR switches respond to IPv6 NS multicast packets from hosts as a proxy.

When the device does not support the data center interconnection tunnel function, BGP-EVEN can be configured on VTEP-1 and all VTEP devices on the network so that VTEP-1 establish VXLAN tunnels with other VTEP devices, thereby forming a full mesh network. The figure below shows the topology.

Figure 1-6



Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

Deployment

- Configure an IPv4 unicast routing protocol (such as OSPF) on switches to ensure that unicast routes are reachable.
- Configure the BGP routing protocol (supporting EVPN) on the TOR and core switches so that the switches establish neighbor relationships mutually.
- Deploy the VXLAN gateway on the TOR switches and the VXLAN bridge on VTEPs.
- Configure core switches to interconnect to external networks at L3.
- (Optional) Deploy ARP suppression on the TOR switches.
- (Optional) Deploy ARP proxy on the TOR switches.

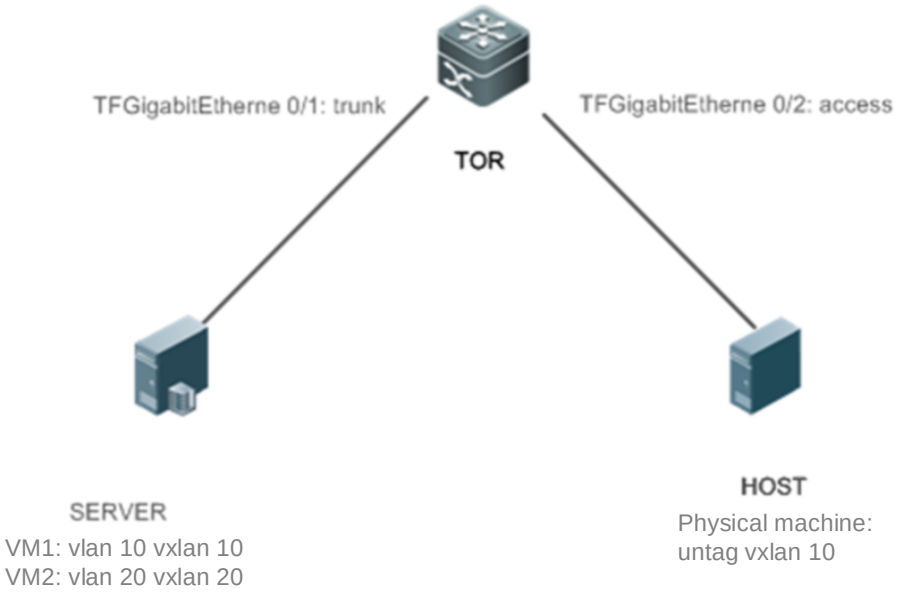
- (Optional) Deploy IPv6 ND suppression on the TOR switches.

1.2.7. Deployment of L2 Subinterfaces to Access a VXLAN

Scenario

A server can access a VXLAN through an L2 subinterface and the access using other subinterfaces is not affected.

Figure 1-7



- On the TOR, configure the VLAN or untagged access mode for subinterfaces and configure a VXLAN instance (that is, gateway).

Deployment

- Complete the function configuration on virtual machines on virtual servers as well as on the physical server.
- Create an L2 subinterface on the TOR switch, configure VXLAN encapsulation and VLAN or untagged encapsulation rule for the subinterface.
- Create an overlay router interface on the TOR switch and configure the VXLAN gateway IP address.
- Configure the VXLAN instance to associate with the overlay router interface on the TOR switch to implement VXLAN routing.

1.3. Features

Basic Concepts

VXLAN Packet Format

A VXLAN encapsulates the Ethernet frames into UDP packets and transmits them on the IP core network.

The VXLAN defines a VTEP entity, which encapsulates the data generated by the virtual machine into the UDP headers, and sends the data out. After the encapsulation, the MAC address and VLAN information of the virtual machine no longer serves as the basis for data forwarding.

The VTEP entity can be software, a hardware server, or other device. If the VTEP function is directly integrated into a hypervisor (also called virtual machine monitor), all virtual machine traffic

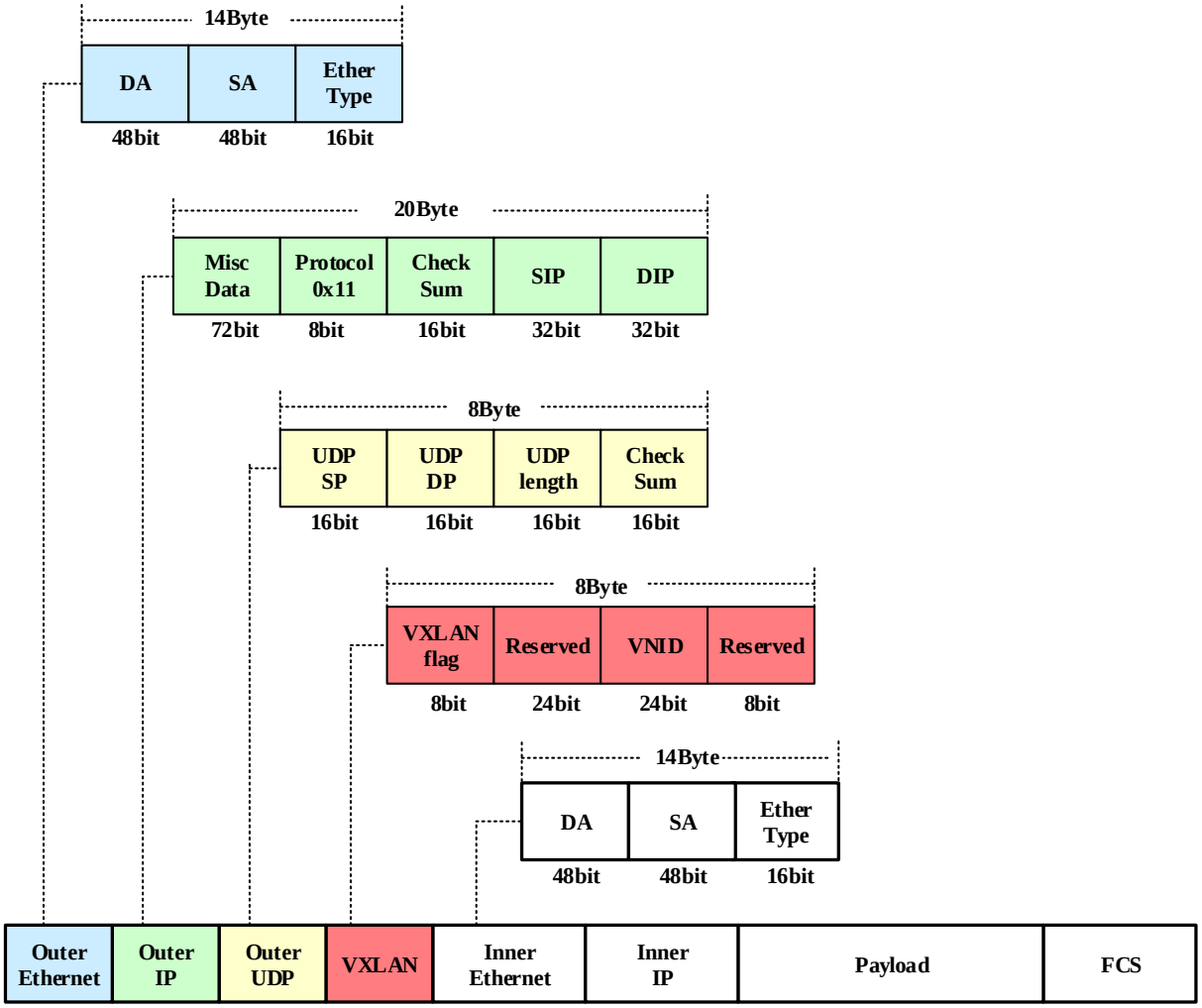
is marked with new VXLAN tags and UDP headers before entering the switch. This is equivalent to creating a tunnel between any two virtual machines.

As the VLAN information of the virtual machine is externally invisible, a new VXLAN label (VNI) is added. VNIs replace VLANs to represent different VXLAN segments. Same as the forwarding behavior of VLANs, only the virtual machines with the same VNI in the same VXLAN segment can communicate with each other.

The new UDP header and VNI form a new frame structure. After receiving the data frame sent from the virtual machine, a VTEP encapsulates four elements (which are the VXLAN header, outer UDP header, outer IPv4 header, and outer Ethernet frame header from inside out) to form a new frame header. In the new frame header, the original source and destination MAC addresses, inner VLAN tag, and Ethernet type that are carried by the inner data frame remain the same.

The format of an encapsulated VXLAN frame is as follows:

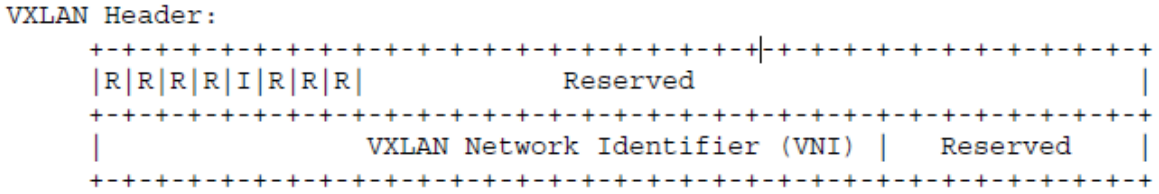
Figure 1-8



1.3.1. Packet Format

VXLAN Header Information

Figure 1-9

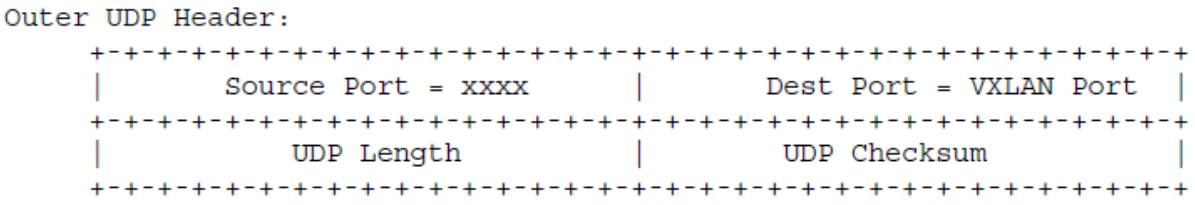


A VXLAN header has 64 bits. In the design of the current protocol version, the sole purpose of a VXLAN header is to carry the 24-bit VNI assigned by the VTEP.

- Flag (8 bits): The I bit must be set to 1 to indicate a valid VNI, and the R bit must be set to 0.
- VXLAN segment ID/VNI: Includes 24 bits and indicates the VXLAN network identifier. Only the virtual machines that belong to the same VXLAN can communicate with each other.
- Reserved: The 24th bit and 8th bit are reserved, and are set to 0.

Outer UDP Header

Figure 1-10

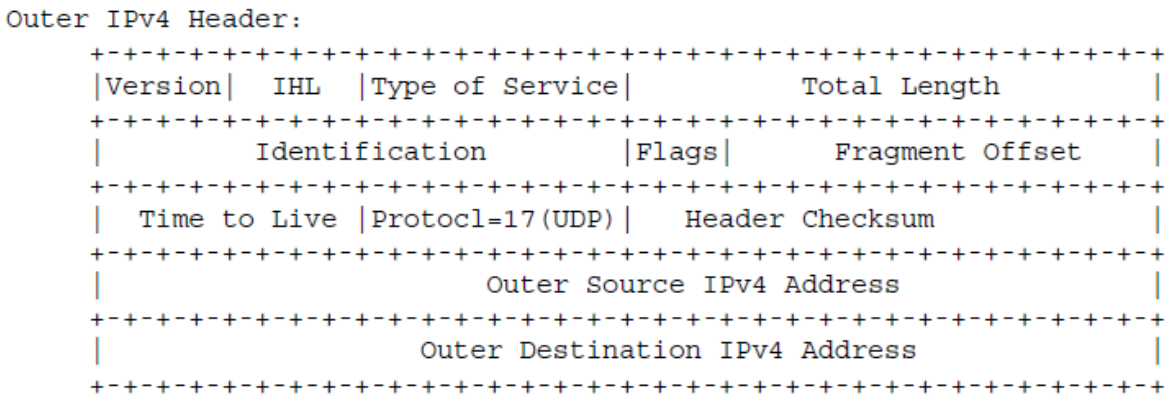


The definitions of the fields of the UDP header are as follows:

- Source Port: Indicates the source port ID of the UDP packet. Assigned by the VTEP, the source port ID is the result of the hash operation of the L2 header of the data frame. This hash result can serve as the basis for traffic load balancing.
- Dest Port: Indicates the destination port ID. The port ID assigned by the Internet Assigned Numbers Authority (IANA) is 4789.
- UDP Length: Indicates the length of the UDP header.
- UDP Checksum: Indicates the UDP checksum, which is set to 0 for transmission.

Outer IP Header

Figure 1-11



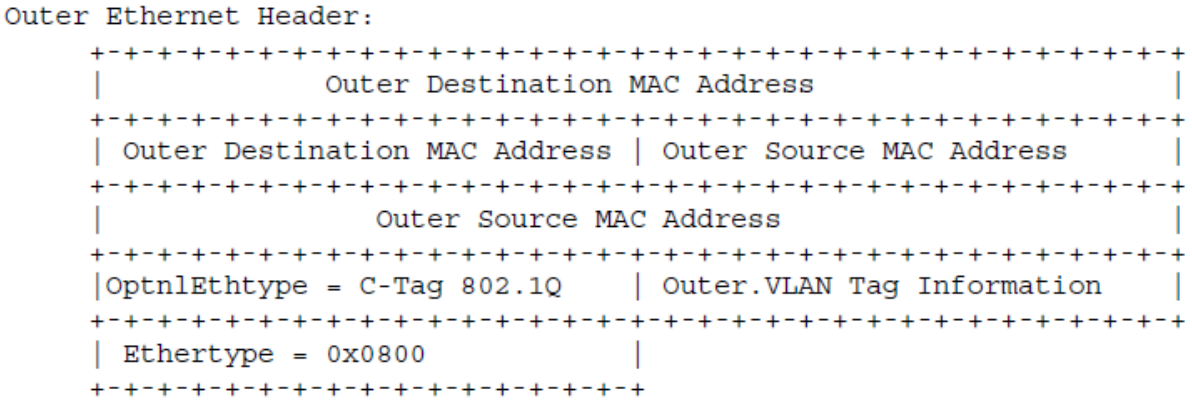
The definitions of the fields of the outer IP header are as follows:

- Source IPv4 Address: Identifies the IP address of the VTEP that corresponds to the virtual machine.
- Destination IPv4 Address: Indicates the unicast or multicast IP address. If it is a unicast IP address, it indicates the IP address of the VTEP corresponding to the virtual machine to be communicated with.

The IP address of the outer IP header is no longer the address of the virtual machines of both communication parties, but the address of the VTEPs at both ends of the tunnel. If the hypervisor directly takes over the work of the VTEP, the IP address is the IP address of the NIC of the server that runs the hypervisor. If the VTEP is an access switch, the IP address is the IP address of an egress interface or the IP address of an L3 switch virtual interface (SVI).

Outer Ethernet Header

Figure 1-12



The definitions of the fields of the outer Ethernet header are as follows:

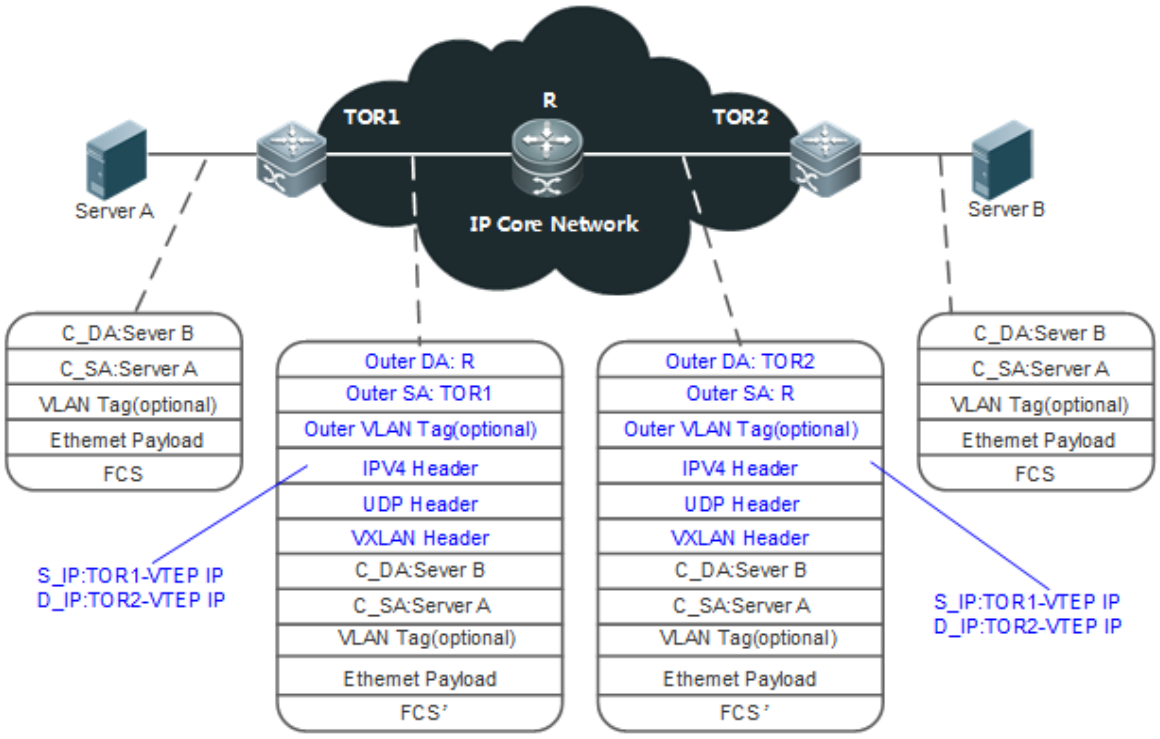
- Destination MAC address: Next-hop MAC address directed to the IP address of the destination VTEP.
- Source MAC address: MAC address of the local VTEP.
- VLAN tag: Optional.

1.3.2. Forwarding Model

VXLAN Bridging Principle

VXLAN encapsulates Ethernet packets within UDP packets to transmit them on the IP network. On the receiver, the VXLAN packets are decapsulated into Ethernet packets and then forwarded, as shown in Figure 2-13.

Figure 1-13



- Switch TOR1 receives the common Ethernet packet, and then encapsulates the packet into a VXLAN packet.
- The VXLAN packet is forwarded in the IP core network. As shown in Figure 1-13, R forwards the VXLAN packet.
- Switch TOR2 receives the VXLAN packet, and then decapsulates and forwards it at L2 of the LAN.

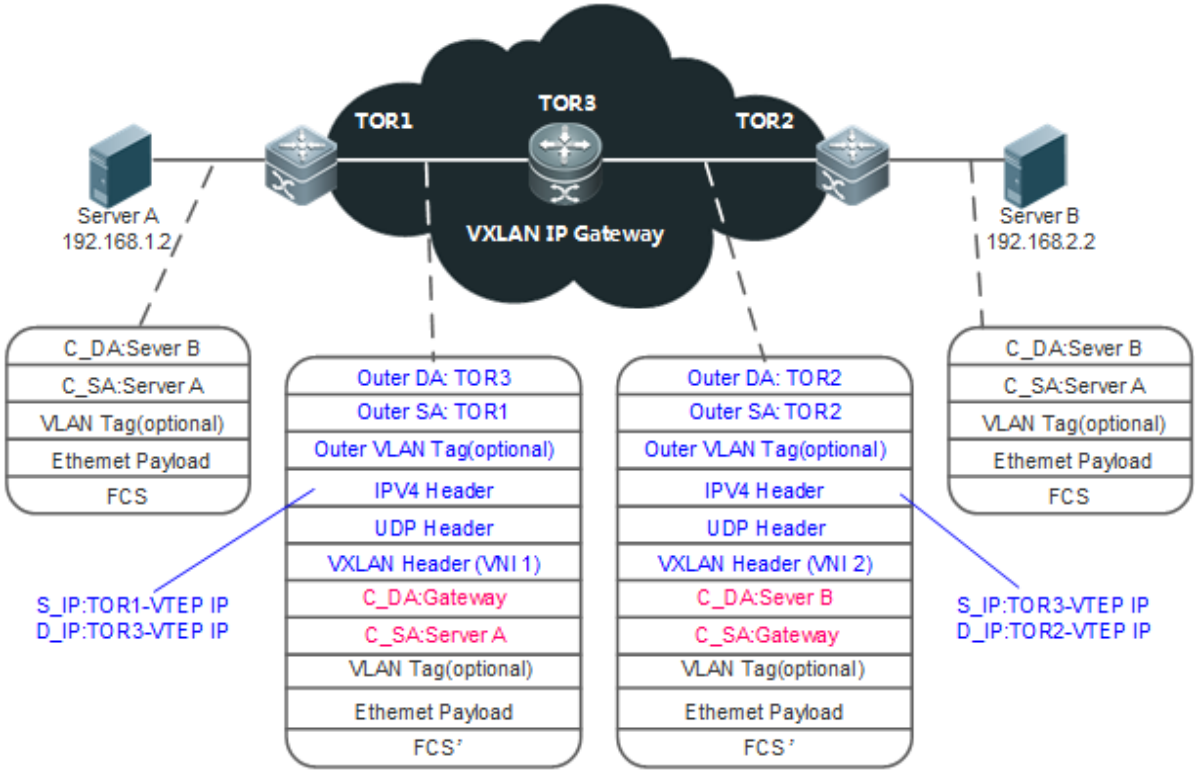
Overview

Feature	Description
VXLAN Bridging and Forwarding	Encapsulates broadcast, multicast, and unknown unicast packets into IP multicast packets to realize flooding. The well-known unicast packets are encapsulated and forwarded by searching the VXLAN address table for the MAC address and IP address.

VXLAN Routing Principle

VXLANs interconnect with each other through the VXLAN IP gateway, as shown in Figure 1-14.

Figure 1-14



- To implement cross-VXLAN communication, Server A first sends a packet to the IP gateway, which is deployed on TOR3.
- The packet sent by Server A is encapsulated by TOR1 into a VXLAN packet and then sent to TOR3.
- After receiving the VXLAN packet, TOR3 finds that the destination MAC address is the local MAC address and sends the packet to TOR2 after VXLAN routing.
- After receiving the packet from TOR3, TOR2 decapsulates the packet and sends it to Server B.

Overview

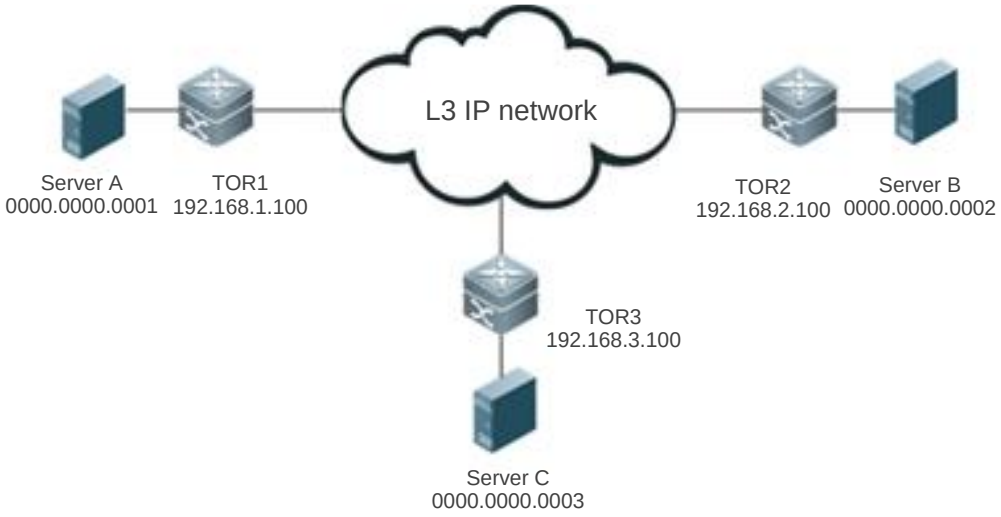
Feature	Description
VXLAN Routing and Forwarding	Implements cross-VXLAN communication and supports communication between a conventional IP network and a VXLAN. A VXLAN router can serve as a VXLAN IP gateway.

1.3.3. Forwarding Process

Working Principle

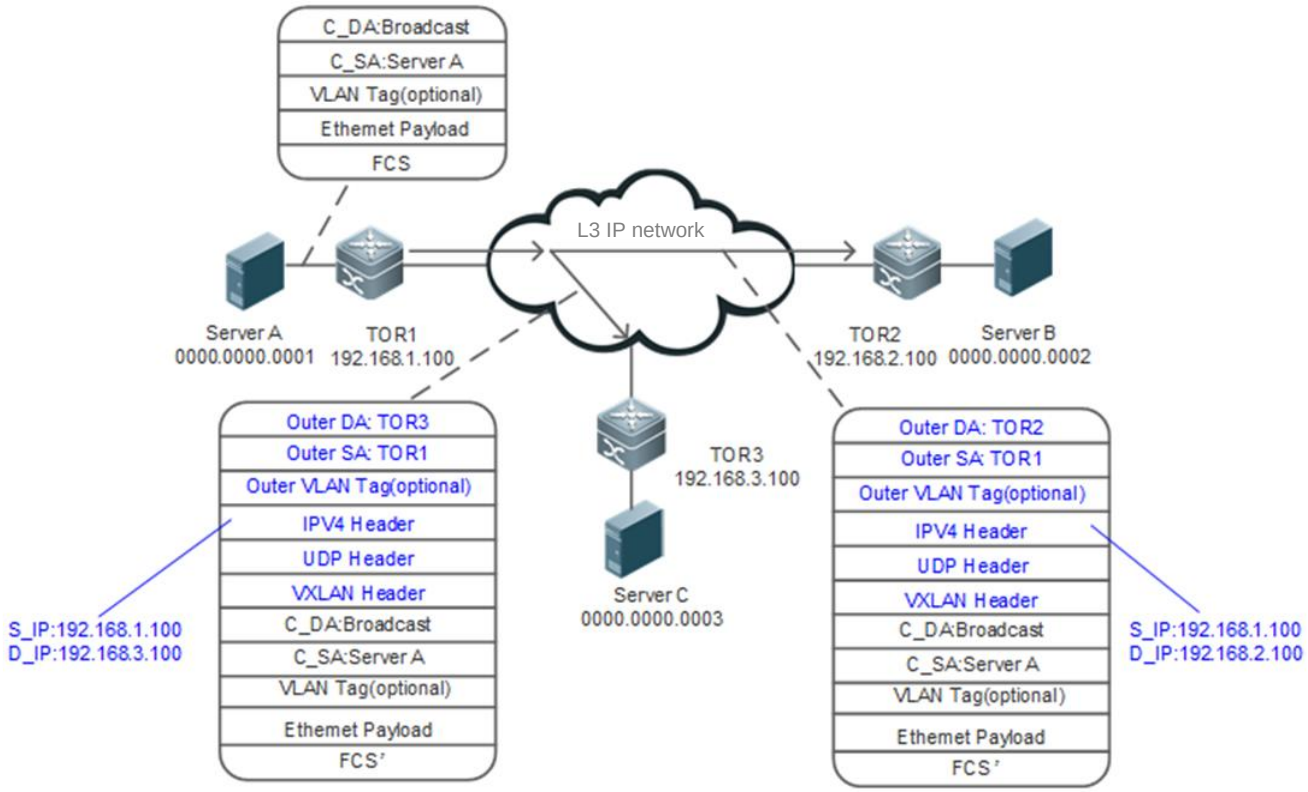
AS shown in Figure 1-15, three servers use a VXLAN to achieve L2 interconnection on the IP network. The VXLAN VNI
is 100.

Figure 1-15



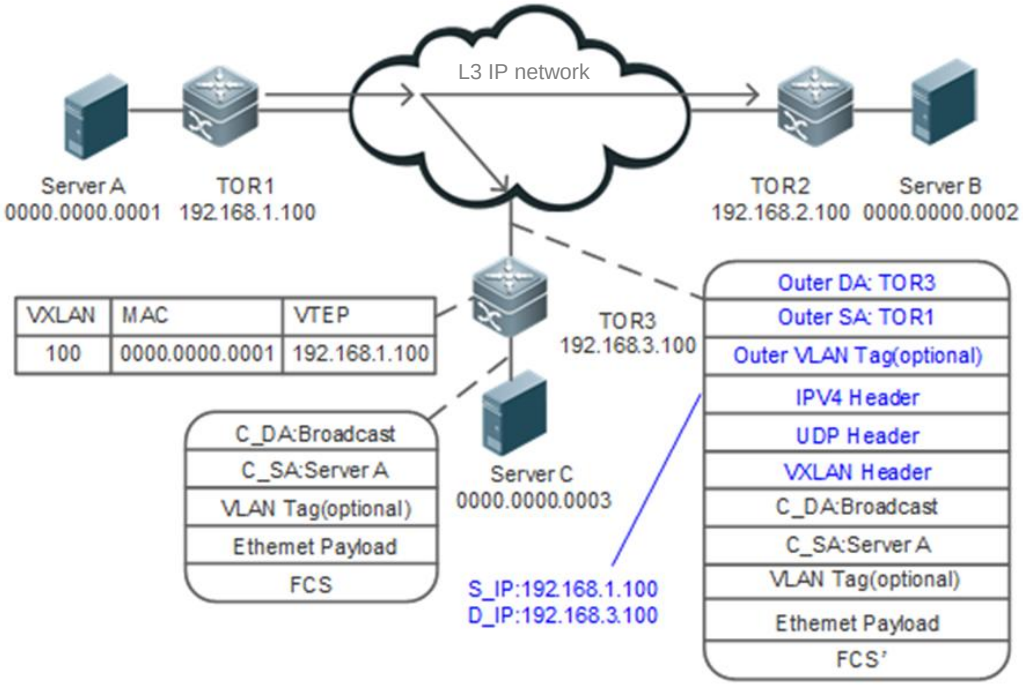
The VXLAN packet forwarding process is described by using an example in which Server A sends an Address Resolution Protocol (ARP) request to Server B and Server B returns an ARP response.

Figure 1-16



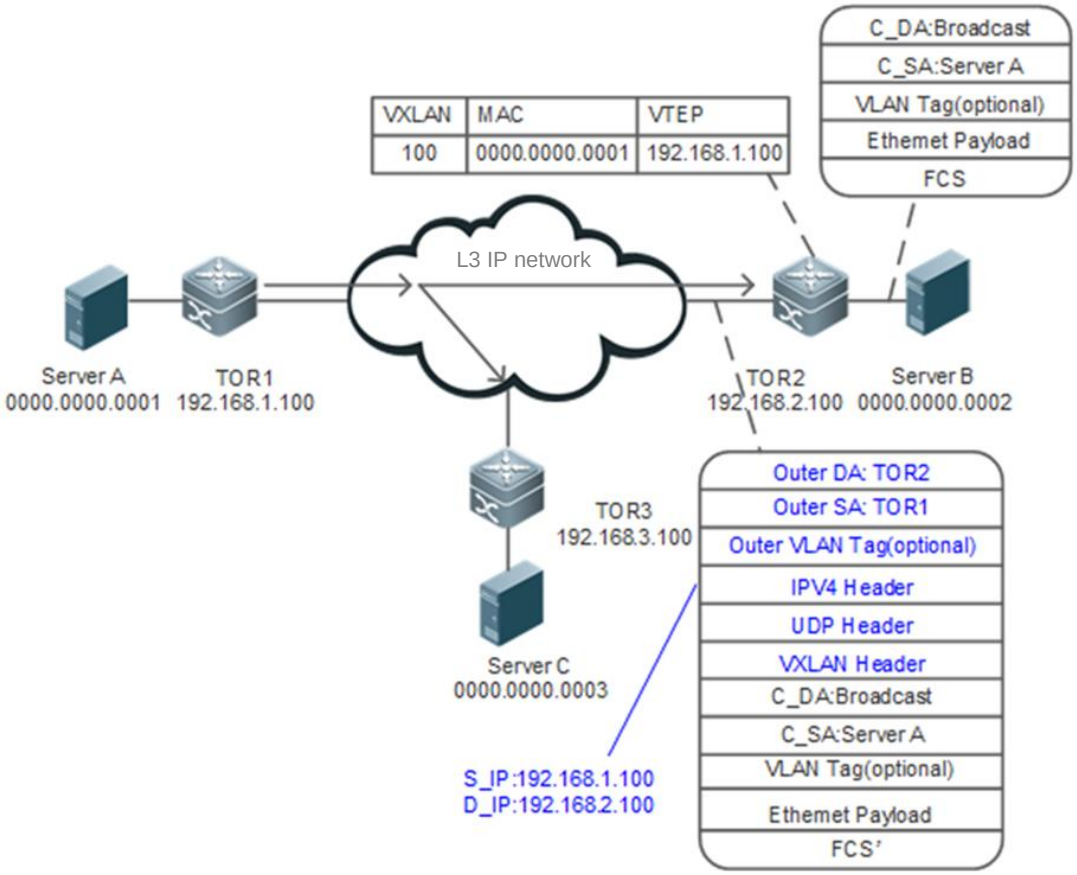
1. Server A sends an ARP request, which is a broadcast packet. After receiving the ARP request, switch TOR1 floods the broadcast packet in tunnel header replication mode, encapsulates it into two unicast packets, and sends them to TOR2 and TOR3 through tunnels. (Switch TOR1 floods the broadcast packet to all tunnels. The tunnel between TOR1 and TOR2, and tunnel between TOR1 and TOR3 are created.)
2. The IP core network forwards the multicast VXLAN packet.

Figure 1-17



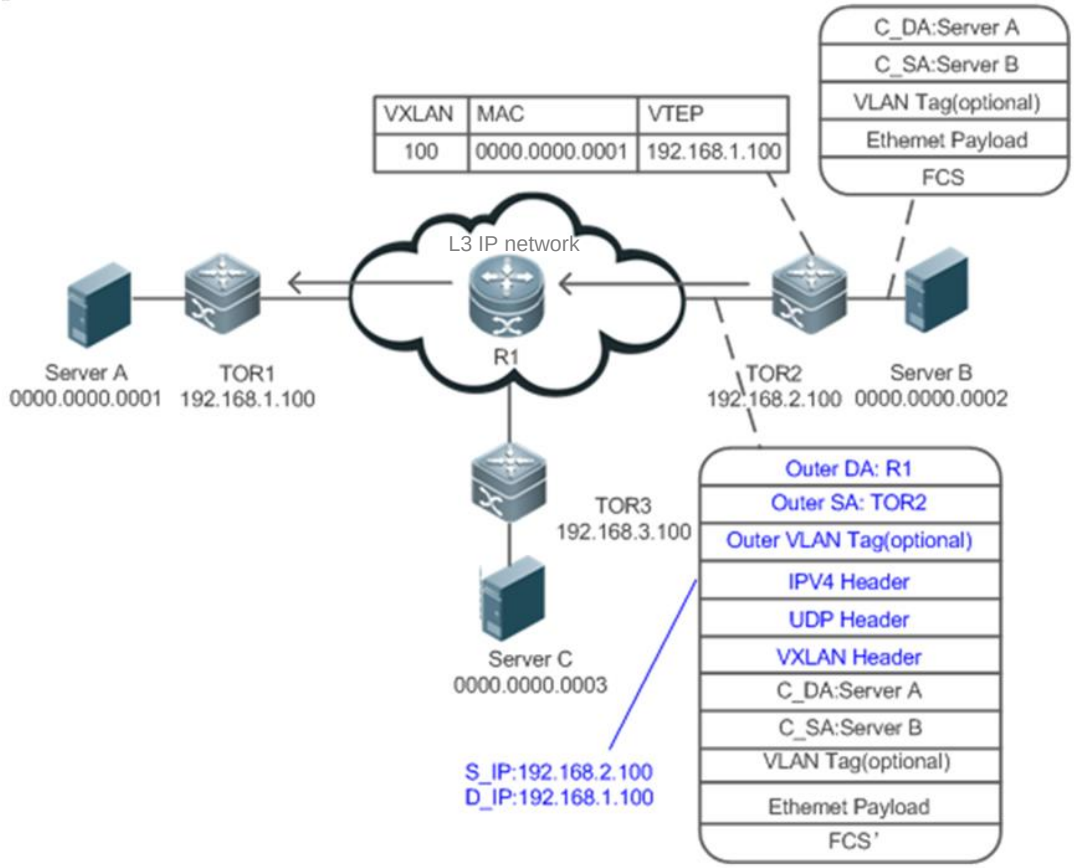
3. After receiving the VXLAN packet, TOR3 decapsulates the packet into an Ethernet packet and implements VXLAN address learning (the VXLAN ID is 100, the MAC address is 0000.0000.0001, and the IP address is 192.168.1.100).

Figure 1-18



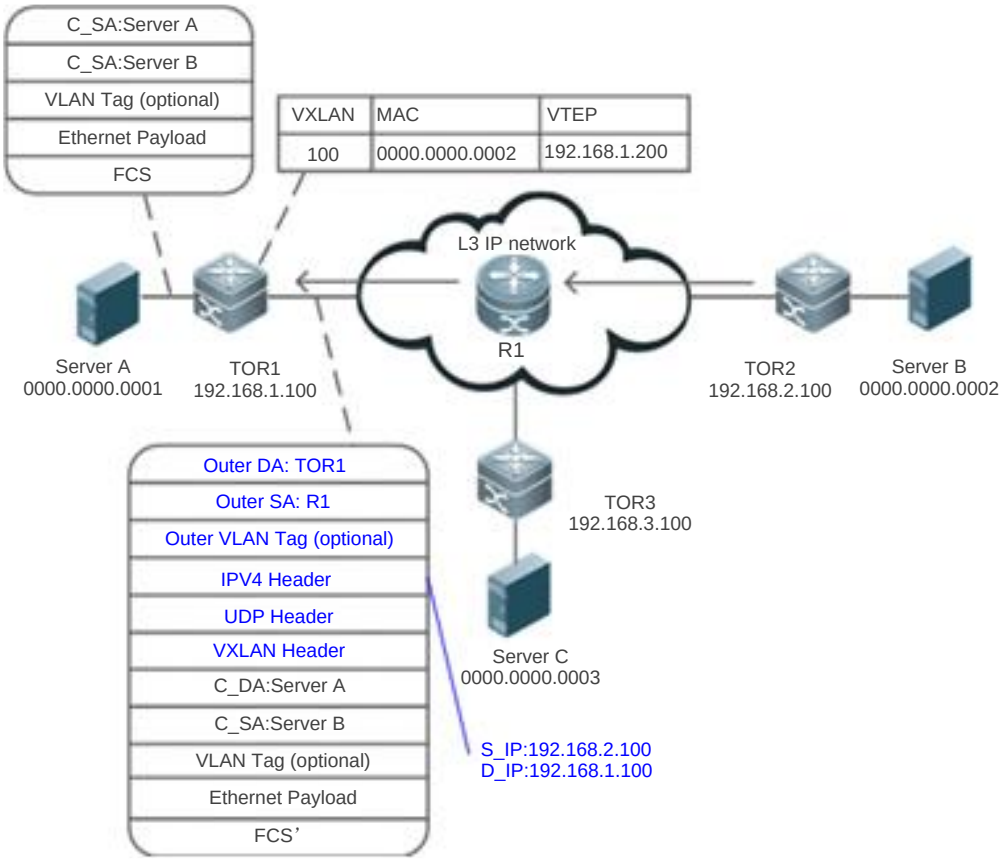
4. After receiving the VXLAN packet, TOR2 decapsulates the packet into an Ethernet packet, implements address learning (the VXLAN ID is 100, the MAC address is 0000.0000.0001, and the IP address is 192.168.1.100) and forwards the packet. Then, Server B receives the ARP request packet and returns a response packet.

Figure 1-19



5. After receiving the ARP response packet from Server B, TOR2 searches the address table and finds that the destination IP address is 192.168.1.100. Then, TOR2 encapsulates the packet into a unicast VXLAN packet (the outer source IP address is 192.168.2.100) destined for the switch at 192.168.1.100.

Figure 1-20

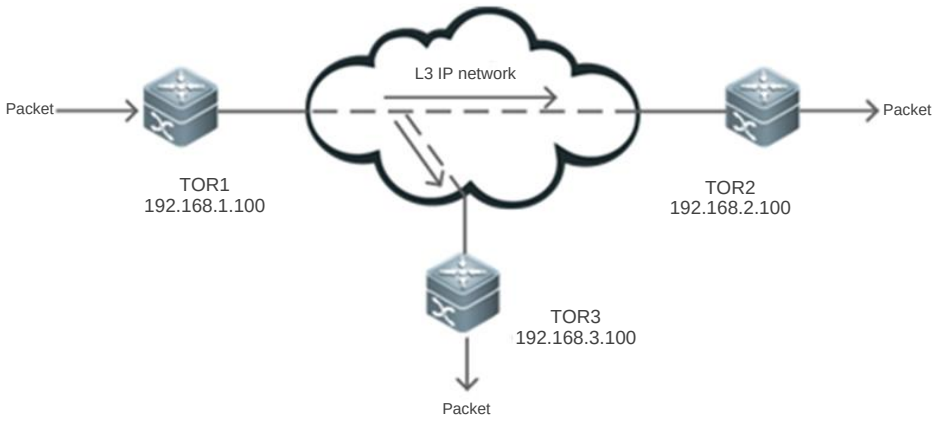


6. The IP core network forwards the VXLAN packet.
7. TOR1 receives the ARP response packet encapsulated in the VXLAN, decapsulates the packet into an Ethernet packet, implements VXLAN address learning (the VXLAN ID is 100, the MAC address is 0000.0000.0002, and the IP address is 192.168.2.100), and forwards the packet. Then, Server A receives the ARP response packet.

Multicast VXLAN Packet Flooding

A VXLAN uses multicast packets to flood broadcast, multicast, and unknown unicast packets. After receiving an ARP request packet, TOR1 encapsulates the packet into a multicast VXLAN packet and sends it to TOR2 and TOR3, as shown in Figure 1-21.

Figure 1-21



VTEP Address Learning

As shown in Figure 1-21, in the process of using multicast packets to flood broadcast, multicast, and unknown unicast packets, TOR2 and TOR3 learn the VTEP information during decapsulation, and therefore establish neighbor relationships.

Related Configuration

Configuring VXLAN Type Instance

No VXLAN instance is configured on the switches by default.

Run the **vxlan** *vni-number* command to create a VXLAN instance.

Configuring VLAN Associated with VXLAN Instance

Run the **extend-vlan** *vlan-id* command in VXLAN instance configuration mode to configure the associated VLAN.

1.4. Configuration

1.4.1. Configuring VXLAN SDN

Configuration Effect

- Create a VXLAN instance and associate it with the overlay router interface and overlay tunnel interface. Provide the VXLAN routing (IP gateway) function to achieve cross-VXLAN communication. The VXLAN configurations can be delivered by the SDN controller over communication mechanisms such as Network Configuration Protocol (NETCONF), or can be implemented by CLI configuration.
- Configure the anycast gateway and an anycast MAC address to provide centralized anycast all-active gateway function. The centralized anycast all-active gateways serve as one logical gateway (VTEP) to communicate with external devices and use the same VTEP IP address. Only one tunnel is configured between each TOR bridge device and the logical gateway. Packets are balanced to physical gateways via the underlay ECMP to achieve the gateway all-active function.
- Enable the SDN controller to deliver the host routes and VXLAN forwarding flow table to the gateways and the gateways generate routes and entries through automatic learning. You can run the configuration commands to enable or disable the automatic learning function on the gateways. When the SDN controller malfunctions, the automatic learning function ensures that the VXLAN works normally.

Notes

- The VXLAN configurations can be delivered by the SDN controller over communication mechanisms such as NETCONF, or can be implemented by CLI configuration. Only configuration delivery from the SDN controller is recommended in normal cases.
- The VXLAN instances require support from existing unicast routes on the network. Therefore, an IPv4 unicast routing protocol, for example, the OSPF protocol, must be configured on the network devices.
- On the centralized anycast gateways, assign the gateway anycast IP addresses to different routing domains to avoid IP conflicts.

Configuration Steps

Creating VXLAN Instances

- Mandatory.

Creating Overlay Router Interfaces

- Mandatory for VXLAN gateways.

Configuring Overlay Router Interfaces as Anycast

- Mandatory for centralized anycast gateways.

Configuring Anycast MAC Address

- Mandatory for centralized anycast gateways.

Configuring Overlay Tunnel

- Mandatory.

Configuring Source and Destination IP Addresses for Overlay Tunnel

- Mandatory.

Associating VXLAN Instance with Overlay Router Interface

- Mandatory for gateways.

Associating VXLAN Instance with VLAN

- Mandatory for TOR bridges.

Associating VXLAN Instance with Overlay Tunnel

- Mandatory.
- This is used to statically designate a VXLAN tunnel.

Configuring Storm Control of VXLAN Instances

- Optional.
- This function is required only when the storm rate needs to be limited based on VXLAN instances.

Configuring Static VXLAN MAC Address Table

- Optional. The VXLAN MAC address table delivered by the SDN controller is represented as a static VXLAN MAC address table.
- You can also configure the static VXLAN MAC address table via CLI configuration.

Configuring VXLAN UDP Destination Port

- Optional. As the VXLAN UDP destination port used by early devices may not be Port 4789, you can run this command to achieve compatibility. In addition, you can also run this command to customize the VXLAN UDP destination port.
- The VXLAN UDP destination port 4789 designated by IANA is used by default.

Enabling ARP Automatic Learning

- Optional. ARP automatic learning is enabled by default.
- After the ARP automatic learning function is enabled, the gateways can automatically learn the APR entries without relying on the SDN control to deliver.

Enabling IPv6 ND Automatic Learning

- Optional. IPv6 ND automatic learning is enabled by default.

- After the IPv6 ND automatic learning function is enabled, the device can automatically learn the host ND entries, with no need to thoroughly rely on ND entries delivered by the SDN controller.

Verification

After SDN-VXLAN is enabled, virtual machines can communicate with each other.

- Run the **show vxlan** *vni-number* command to check whether the VXLAN devices can learn their mutual VTEP neighbor relationships.
- Run the **show vxlan mac** command to check whether the VXLAN MAC address is learned.
- Run the **show arp** command to check whether all local/remote entries are learned. Run the **show ip route** command to check whether the routes of VXLAN IP gateways are learned.
- Run the **show ipv6 neighbors** command to check whether all local/remote IPv6 ND entries are learned. Run the **show ipv6 route** command to check whether the routes of the VXLAN IPv6 gateways are learned.
- Run the **show vxlan udp-port** command to display the VXLAN UDP destination port.

Related Commands

Creating or Entering VXLAN Instances

Command	**vxlan** *vni-number*
Parameter Description	*vni-number*: Indicates the VNI. The value ranges from 1 to 16777215.
Command Mode	Global configuration mode
Usage Guide	N/A

Associating VXLAN Instance with VLAN

Command	**extend-vlan** *vlan-id-list*
Parameter Description	*vlan-id-list*: Indicates the VLAN ID queue. The VLAN ID ranges from 1 to 4094.
Command Mode	VXLAN configuration mode
Usage Guide	Use this command to associate the VXLAN instance with the VLAN. After receiving the VLAN packet, the device will be associated with the VLAN instance.

Creating Overlay Router Interfaces

Command	**interface OverlayRouter** *port-id*
Parameter Description	*port-id*: Indicates the ID of an overlay router interface. The ID ranges from 1 to 16,777,215.
Command Mode	Global configuration mode
Usage Guide	Similar to SVI in a VLAN, this interface serves as the VXLAN IP gateway in the VXLAN routing environment.

Configuring VRF Network for Overlay Router Interface	
Command	**vrf forwarding** *vrf-name*
Parameter Description	*vrf-name*: Indicates the VRF network to which the overlay router interface belongs.
Command Mode	Overlay router interface configuration mode
Usage Guide	Allocate VRF networks to different VXLAN tenants. The traffic of VXLAN instances of different VRF networks is isolated from each other.

Configuring IP Address for Overlay Router Interface	
Command	**ip address** *ip-address mask*
Parameter Description	*ip-address*: Indicates the IP address of the overlay router interface. *mask*: Indicates the subnet mask.
Command Mode	Overlay router interface configuration mode
Usage Guide	Similar to the IP address of the SVI in a VLAN, this IP address serves as the address of the VXLAN IP gateway in the VXLAN routing environment.

Configuring an IPv6 Address for the Overlay Router Interface	
Command	**ipv6 address** *ip-address mask*
Parameter Description	*ip-address*: Indicates the IPv6 address of the overlay router interface. *mask*: Indicates the subnet mask.
Command Mode	Overlay router interface configuration mode
Usage Guide	This IPv6 address serves as the VXLAN IPv6 gateway address in the VXLAN routing environment. It is similar to the IP address of an SVI in a VLAN.

Configuring the Overlay Router Interface as an Anycast Interface	
Command	**anycast-gateway**
Parameter Description	N/A
Command Mode	Overlay router interface configuration mode
Usage Guide	Configure the gateway as an anycast gateway.

Associating VXLAN Instance with Overlay Router Interface	
Command	**router-interface** *interface-name*
Parameter Description	*interface-name*: Indicates the name of the overlay router interface.
Command Mode	VXLAN configuration mode
Usage Guide	Different VXLANs cannot be associated with the same overlay router interface.

Configuring Virtual MAC Address for Anycast Gateways	
Command	**fabric anycast-gateway-mac** *mac-addr*
Parameter Description	*mac-addr:* Indicates the MAC address. The format is xxxx.xxxx.xxxx.
Command Mode	Global configuration mode
Usage Guide	All gateways on which the anycast function is enabled use this MAC address as the gateway MAC address. The virtual MAC address for an anycast gateway must not be the same as the local MAC address or the same as the MAC address of any device on the overlay network.

Creating Overlay Tunnel Interfaces	
Command	**interface OverlayTunnel** *port-id*
Parameter Description	*port-id*: Indicates the ID of the overlay tunnel interface. The ID ranges from 1 to 6144.
Command Mode	Global configuration mode
Usage Guide	This interface is used to statically create an overlay tunnel. You can run the **tunnel-interface** command to associate it with a VXLAN.

Configuring Source IP Address for Tunnel of Overlay Tunnel Interface	
Command	**tunnel source** *ip-address*
Parameter Description	*ip-address*: Indicates the source IP address of a tunnel.
Command Mode	Overlay tunnel interface configuration mode
Usage Guide	Use this command to designate a source IP address for the overlay tunnel. Designate this IP address as the outer source IP address of a packet for encapsulation and forwarding.

Configuring Destination IP Address for Tunnel of Overlay Tunnel Interface	
Command	**tunnel destination** *ip-address*
Parameter Description	*ip-address*: Indicates the destination IP address of a tunnel.
Command Mode	Overlay tunnel interface configuration mode
Usage Guide	Use this command to designate a destination IP address for the overlay tunnel. Designate this IP address as the outer destination IP address of a packet for encapsulation and forwarding. The destination IP address of the tunnel is globally unique. Different overlay tunnels cannot be configured with the same destination IP address. Otherwise, a conflict occurs.

Associating VXLAN Instance with Overlay Tunnel Interface	
Command	**tunnel-interface** *interface-name*
Parameter Description	*interface-name*: Indicates the name of the overlay tunnel interface.
Command Mode	VXLAN configuration mode
Usage Guide	Use this command to designate the VXLAN VTEP statically.

Configuring Storm Control of VXLAN Instances	
Command	**storm-control {broadcast | multicast | unicast} [***kbps-value* **| pps** *pps-value*]
Parameter Description	*kbps-value*: Indicates the rate limit value (unit: kbit/s). *pps-value*: Indicates the rate limit value (unit: packet count/s).
Command Mode	VXLAN configuration mode
Usage Guide	Configure the storm control when the storm rate needs to be limited based on the VNI.

Configuring Static VXLAN MAC Address Table

Command	**vxlan mac static** *mac-addr* **vni** *vxlan-id* **interface** *interface-name*
Parameter Description	*mac-addr*: Indicates the MAC entry address. The format is xxxx.xxxx.xxxx. *vxlan-id*: Indicates the VNI of the MAC entry. *interface-name*: Indicates the next hop egress of the MAC table. It can be an overlay tunnel interface, an Ethernet interface, or an aggregate port (AP). *vid*: Indicates the ID of a VLAN to which the MAC entry belongs.
Command Mode	Global configuration mode
Usage Guide	1. Use this command to deliver the static VXLAN MAC entries via the SDN controller or configure the static VXLAN MAC entries via CLI static configuration. This command is mainly used for setting the host forwarding table. 2. When the next-hop interface is not an overlay tunnel interface, a VID must be configured and the VID is not required for overlay tunnel interfaces.

Configuring VXLAN UDP Destination Port

Command	**vxlan udp-port** *port-number*
Parameter Description	*port-number*: Indicates the UDP destination port ID. The value ranges from 0 to 65535 and the default value is 4789.
Command Mode	Global configuration mode
Usage Guide	Note that the UDP destination port cannot be same as commonly used UDP ports.

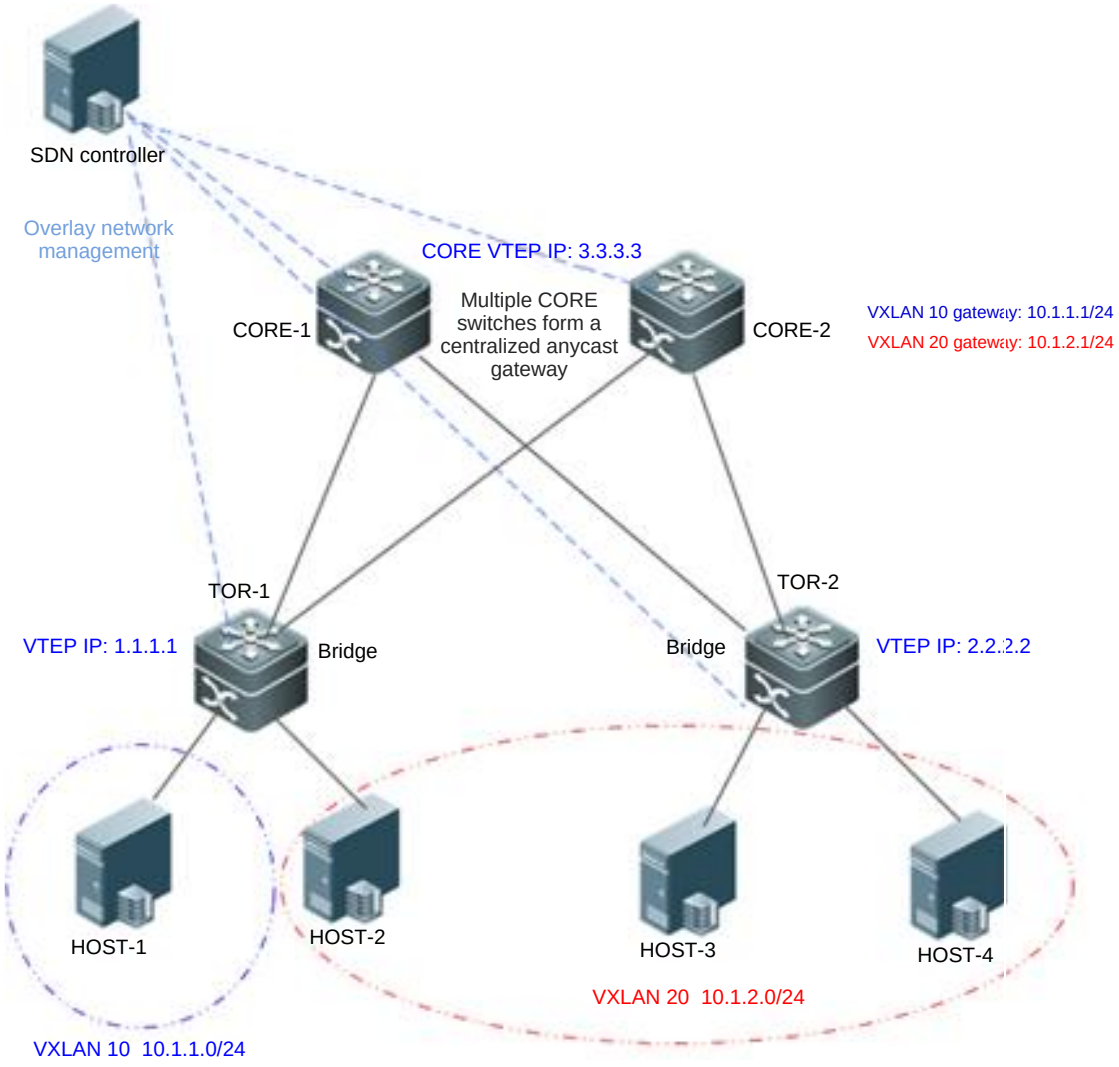
Configuration Example

Note:

- Only configuration related to the VXLAN is described below.
- Only IPv4 configuration is used as an example below and the IPv6 scenario configuration is largely the same as the IPv4 scenario configuration.

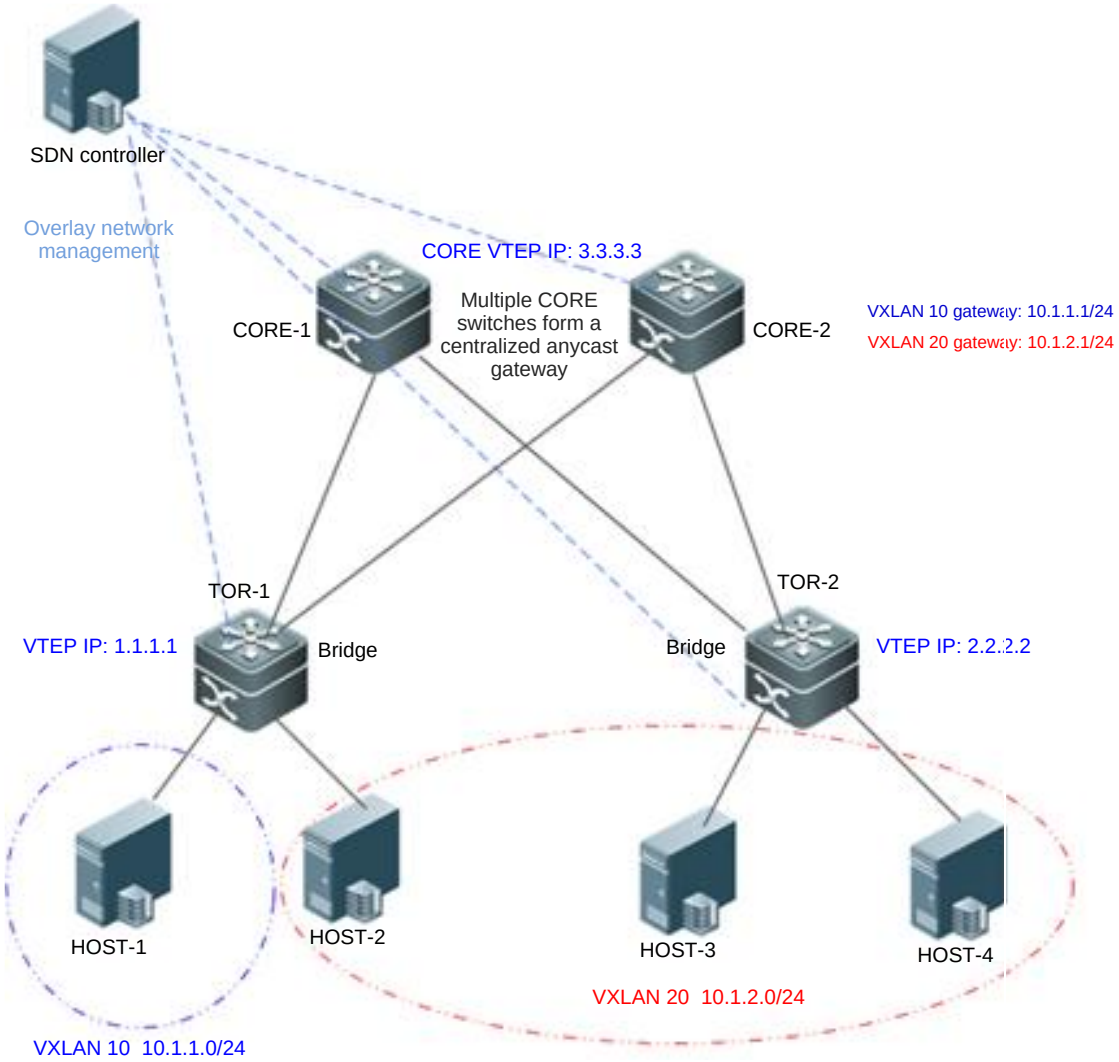
VXLAN Configuration Instance

Scenario
Figure 1-22



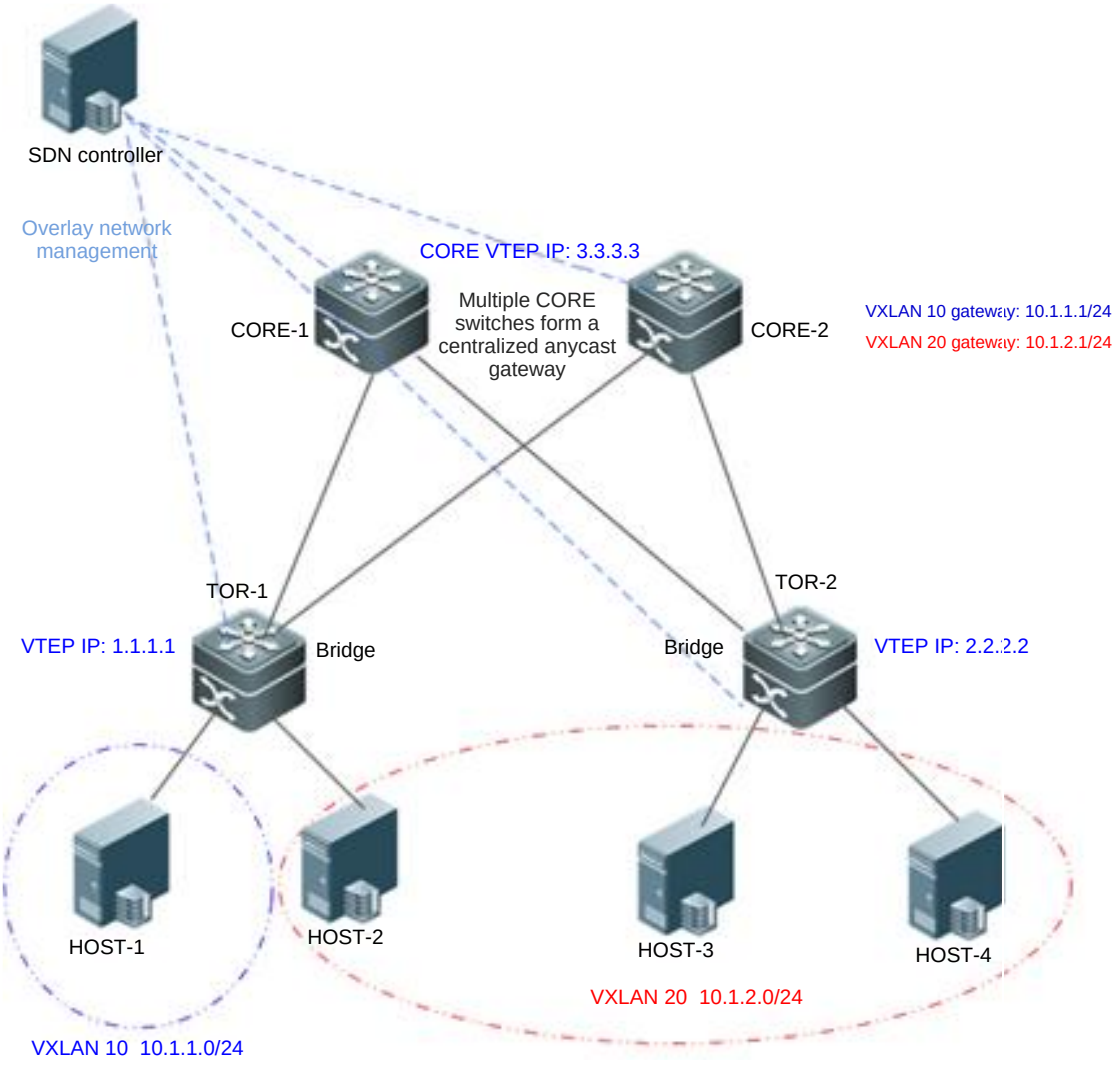
Configuration Steps	• Configure an IPv4 unicast routing protocol such as the OSPF protocol on TOR-1, TOR-2, CORE-1, and CORE-2 to ensure that unicast routes are reachable. • Configure loopback IP addresses on TOR-1, TOR-2, CORE-1, and CORE-2 and distribute packets via the unicast routing protocol. The VTEP IP addresses of CORE-1 and CORE-2 must be the same and be allocated to different routing domains. • Configure a VXLAN on the virtual server and designate the gateway address of the virtual machine. . • Establish a BGP neighbor relationship between CORE-1 and CORE-2 and configure the BGP-EVPN routing protocol on them. The following configuration can be delivered by the SDN controller: • Create VXLAN instances VXLAN 10 and VXLAN 20 on TOR-1, and associate them with VLAN 10 and VLAN 20 respectively. Configure the address learning mode as SDN controller advertisement. Configure two overlay tunnels to connect TOR-1 with TOR-2 and CORE. Associate VXLAN 10 and VXLAN 20 with

Scenario
Figure 1-22



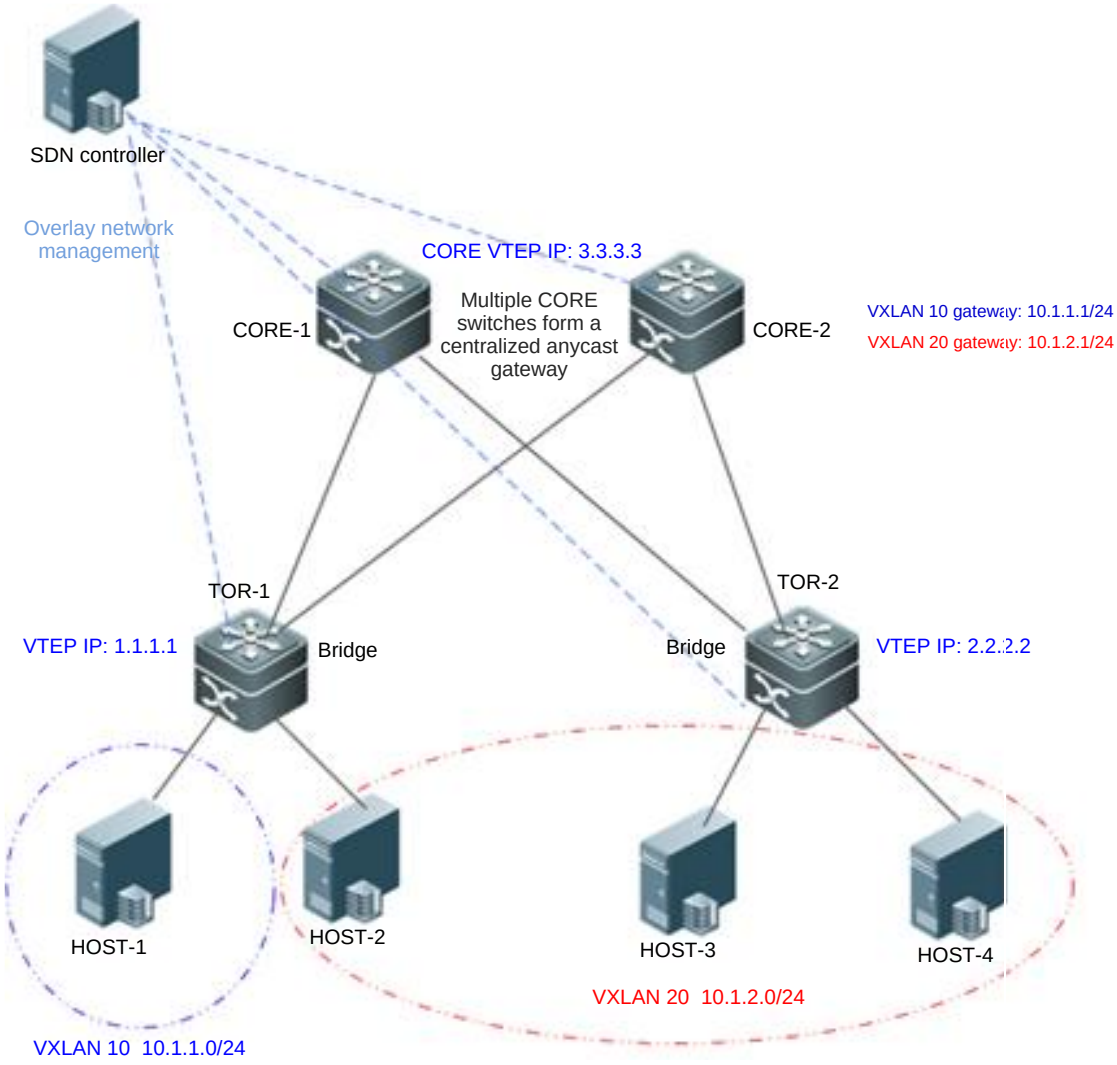
	the two tunnels separately. • Create VXLAN instance VXLAN 20 on TOR-2 and associate it with VLAN 20. Configure the address learning mode as SDN controller advertisement. Configure two overlay tunnels to connect TOR-2 with TOR-1 and CORE-2. Associate VXLAN 20 with the two tunnels. • Create VXLAN instances VXLAN 10 and VXLAN 20 on CORE-1. Configure the address learning mode as SDN controller advertisement. Configure the anycast MAC address. Configure two overlay router gateway interfaces and configure their IP addresses as 10.1.1.1/24 and 10.1.2.1/24 respectively. Associate VXLAN 10 with the overlay router gateway interface with the IP address 10.1.1.1/24. Associate VXLAN 20 with the overlay router gateway interface with the IP address 10.1.2.1/24. Configure two overlay tunnels to connect CORE-1 with TOR-1 and TOR-2. Associate VXLAN 10 and VXLAN 20 with the two tunnels separately. Enable the synchronization function on all-active VXLAN gateways to synchronize the automatically learned entries between the gateways.

Scenario
Figure 1-22



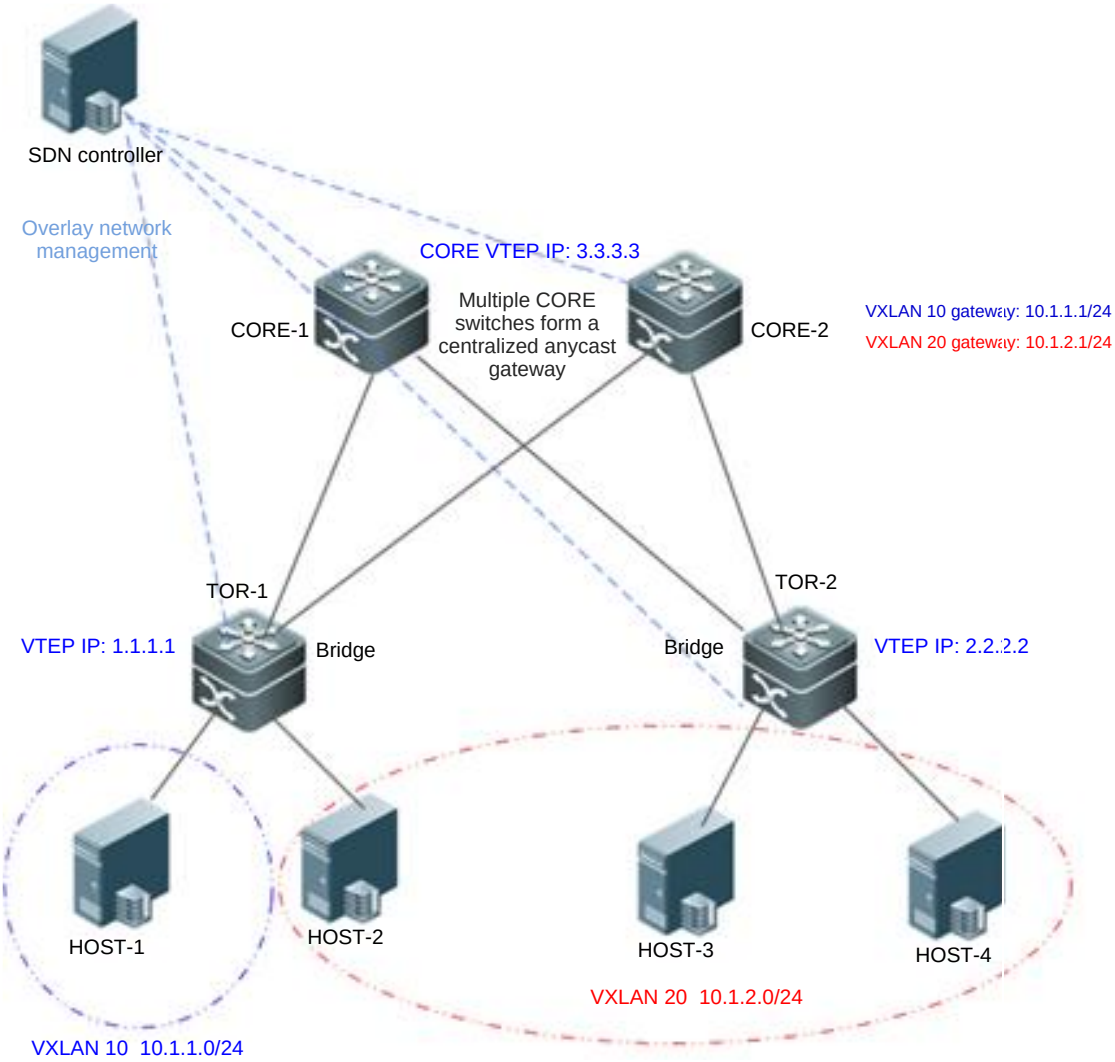
	• The configuration of CORE-2 is same with that of CORE-1.
TOR1	TOR1# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR1(config)#interface loopback 0 TOR1(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255 TOR1(config-if-Loopback 0)# exit TOR1(config)# interface OverlayTunnel 1 TOR1(config-if-OverlayTunnel 1)# tunnel source 1.1.1.1 TOR1(config-if-OverlayTunnel 1)# tunnel destination 2.2.2.2 TOR1(config-if-OverlayTunnel 1)# exit TOR1(config)# interface OverlayTunnel 2 TOR1(config-if-OverlayTunnel 2)# tunnel source 1.1.1.1 TOR1(config-if-OverlayTunnel 2)# tunnel destination 3.3.3.3

Scenario
Figure 1-22



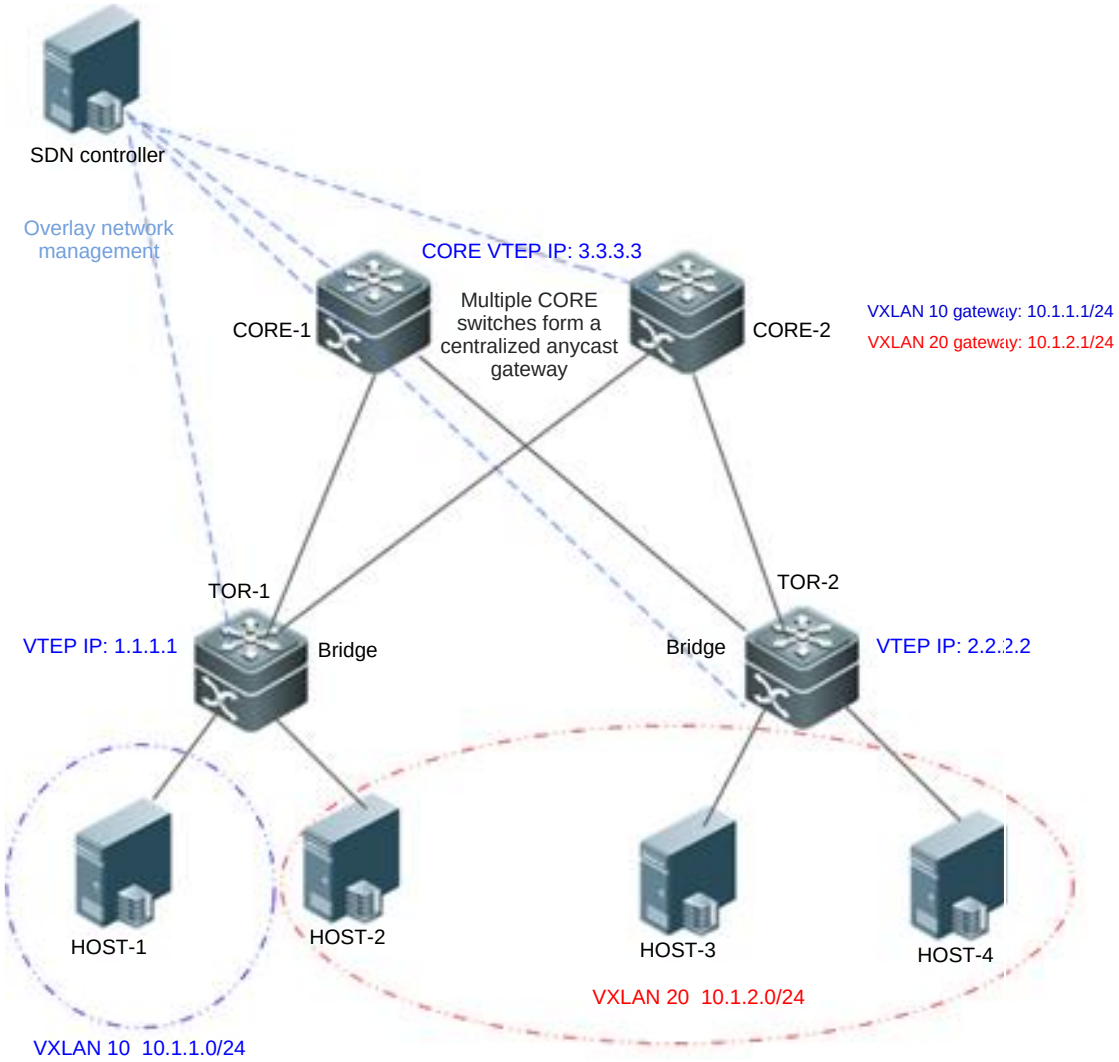
	TOR1(config-if-OverlayTunnel 2)# exit TOR1(config)# vxlan 10 TOR1(config-vxlan)# tunnel-interface OverlayTunnel 1 TOR1(config-vxlan)# tunnel-interface OverlayTunnel 2 TOR1(config-vxlan)# extend-vlan 10 TOR1(config-vxlan)# end TOR1(config)# vxlan 20 TOR1(config-vxlan)# tunnel-interface OverlayTunnel 1 TOR1(config-vxlan)# tunnel-interface OverlayTunnel 2 TOR1(config-vxlan)# extend-vlan 20 TOR1(config-vxlan)# end
TOR2	TOR2# configure terminal Enter configuration commands, one per line. End with CNTL/Z.

Scenario
Figure 1-22



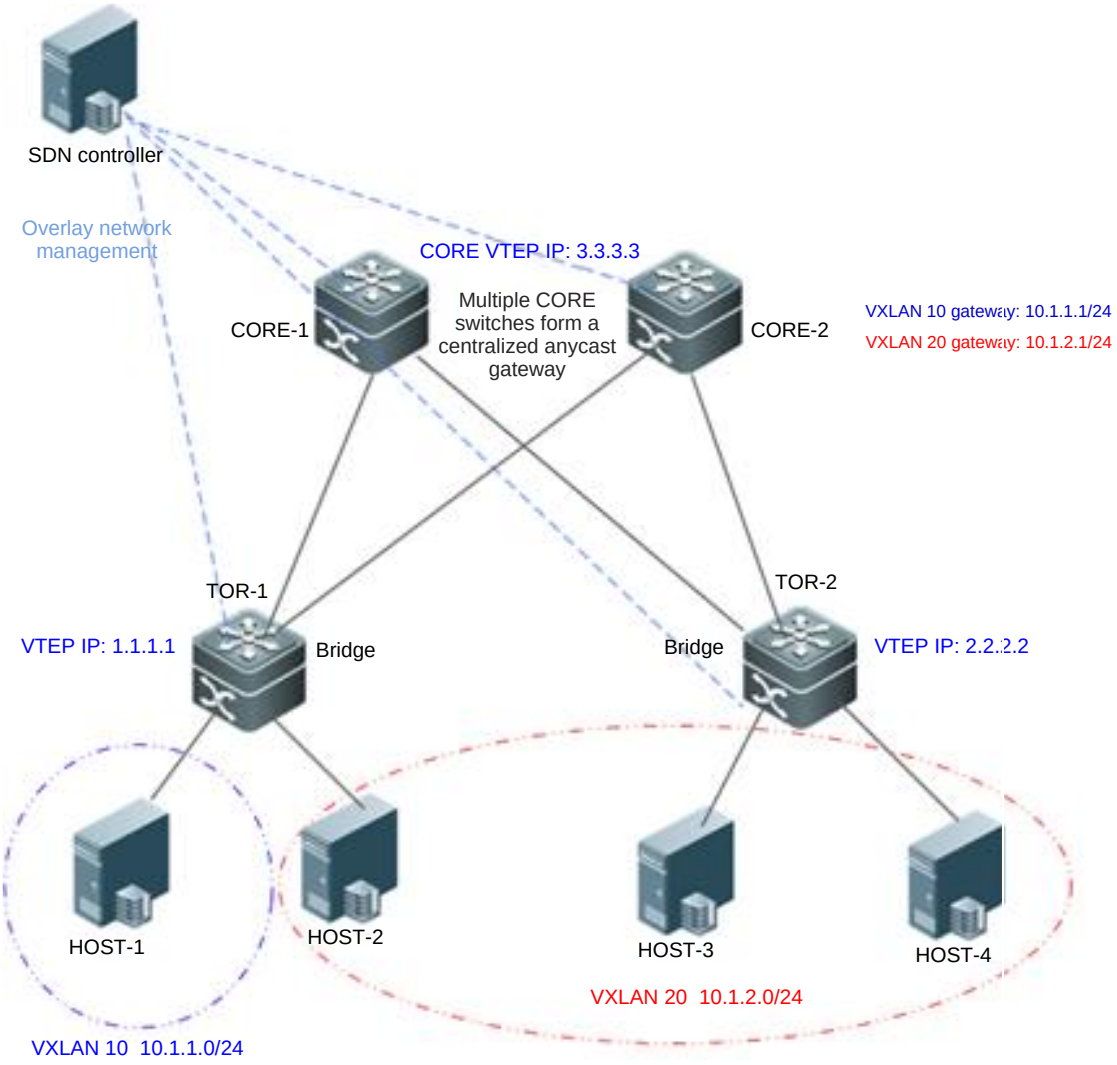

```
TOR2(config)#interface loopback 0
TOR2(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
TOR2(config-if-Loopback 0)# exit
TOR2(config)# interface OverlayTunnel 1
TOR2(config-if-OverlayTunnel 1)# tunnel source 2.2.2.2
TOR2(config-if-OverlayTunnel 1)# tunnel destination 1.1.1.1
TOR2(config-if-OverlayTunnel 1)# exit
TOR2(config)# interface OverlayTunnel 2
TOR2(config-if-OverlayTunnel 2)# tunnel source 2.2.2.2
TOR2(config-if-OverlayTunnel 2)# tunnel destination 3.3.3.3
TOR2(config-if-OverlayTunnel 2)# exit
TOR2(config)# vxlan 20
TOR2(config-vxlan)# tunnel-interface OverlayTunnel 1
TOR2(config-vxlan)# tunnel-interface OverlayTunnel 2
```

Scenario
Figure 1-22



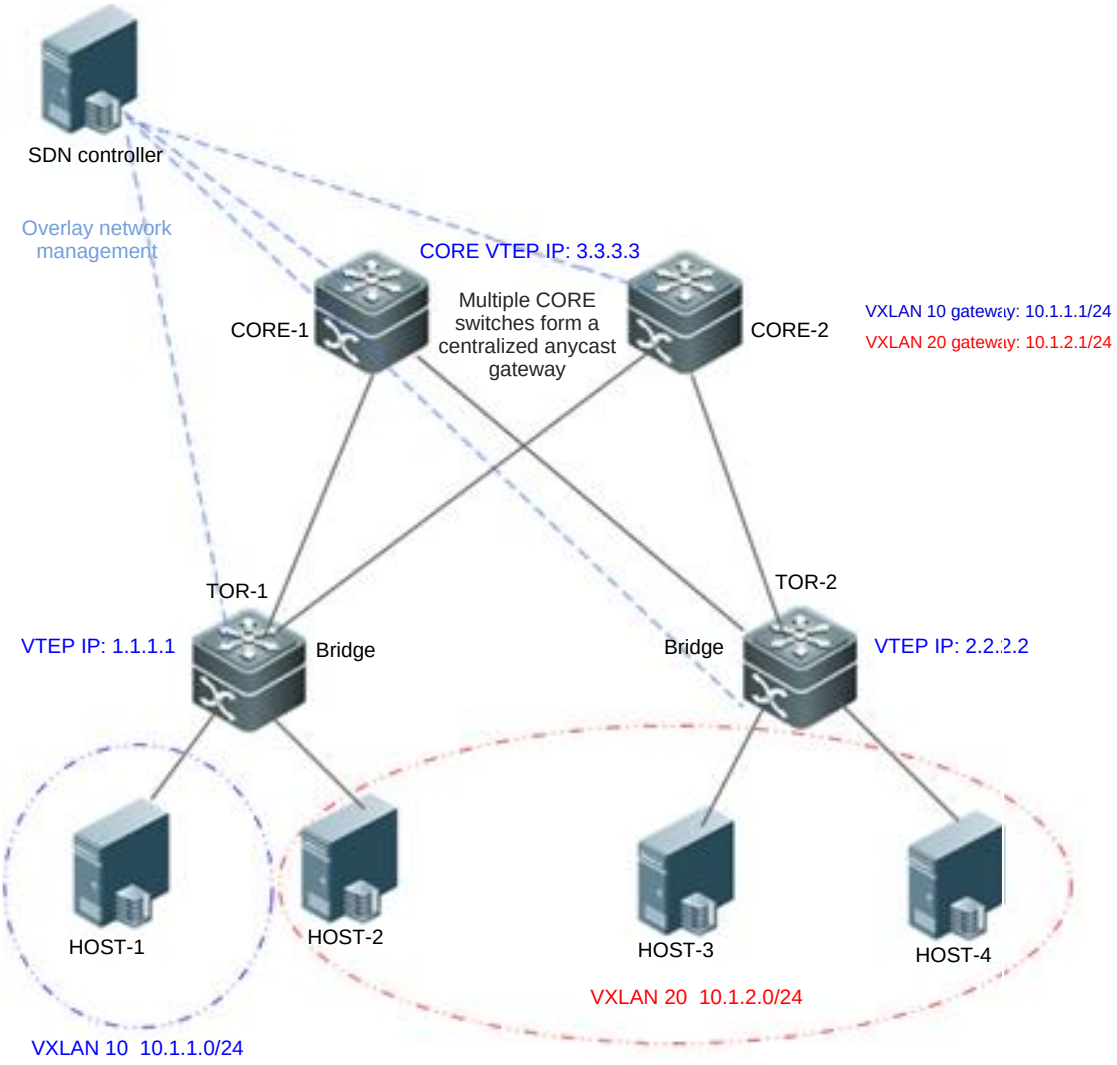
	TOR2(config-vxlan)# extend-vlan 20 TOR2(config-vxlan)# end
CORE1	Create VXLAN instances VXLAN10 and VXLAN20 on CORE-1. Set the address learning mode to SDN controller advertisement. Configure an anycast MAC address. Configure two overlay router gateway interfaces and set their IP addresses to 10.1.1.1/24 and 10.1.2.1/24 respectively. Configure VXLAN10 to associate with the overlay router gateway interface with the IP address of 10.1.1.1/24. Configure VXLAN20 to associate with the overlay router gateway interface with the IP address of 10.1.2.1/24. Configure two overlay tunnels reachable to TOR1 and TOR2 respectively. Configure VXLAN10 and VXLAN20 to associate with the two tunnels respectively. Configure loopback 1 on both CORE-1 and CORE-2 and set the IP address to 3.3.3.4 and 3.3.3.5 for loopback 1. The two core switches establish a BGP neighbor relationship through loopback 1. Configure the L2VPN EVPN address family activation command. TOR1# configure terminal Enter configuration commands, one per line. End with CNTL/Z.

Scenario
Figure 1-22



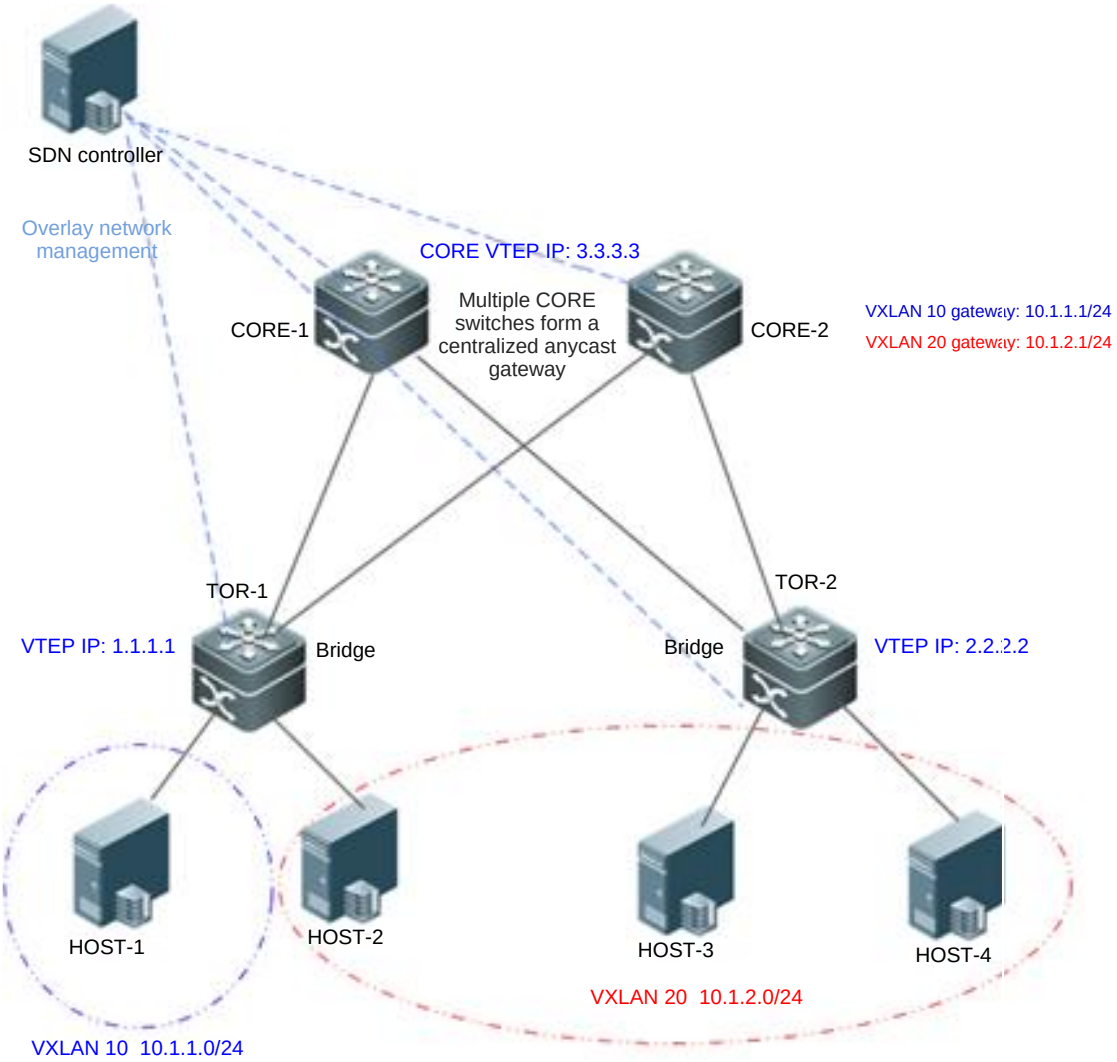

```
CORE1(config)# fabric anycast-gateway-mac 0000.1234.5678
CORE1(config)# interface loopback 0
CORE1(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
CORE1(config-if-Loopback 0)# exit
CORE1(config)# interface loopback 1
CORE1(config-if-Loopback 0)# ip address 3.3.3.4 255.255.255.255
CORE1(config-if-Loopback 0)# exit
CORE1(config)# route bgp  10000
CORE1(config-router)# neighbor 3.3.3.5 remote-as 10000
CORE1(config-router)# neighbor 3.3.3.5 update-source Loopback 1
CORE1(config-router)# address-family l2vpn evpn
CORE1(config-router-af)# neighbor 3.3.3.5 activate
CORE1(config-router-af)# neighbor 3.3.3.5 send-community extended
CORE1(config-router-af)# exit-address-family
```

Scenario
Figure 1-22



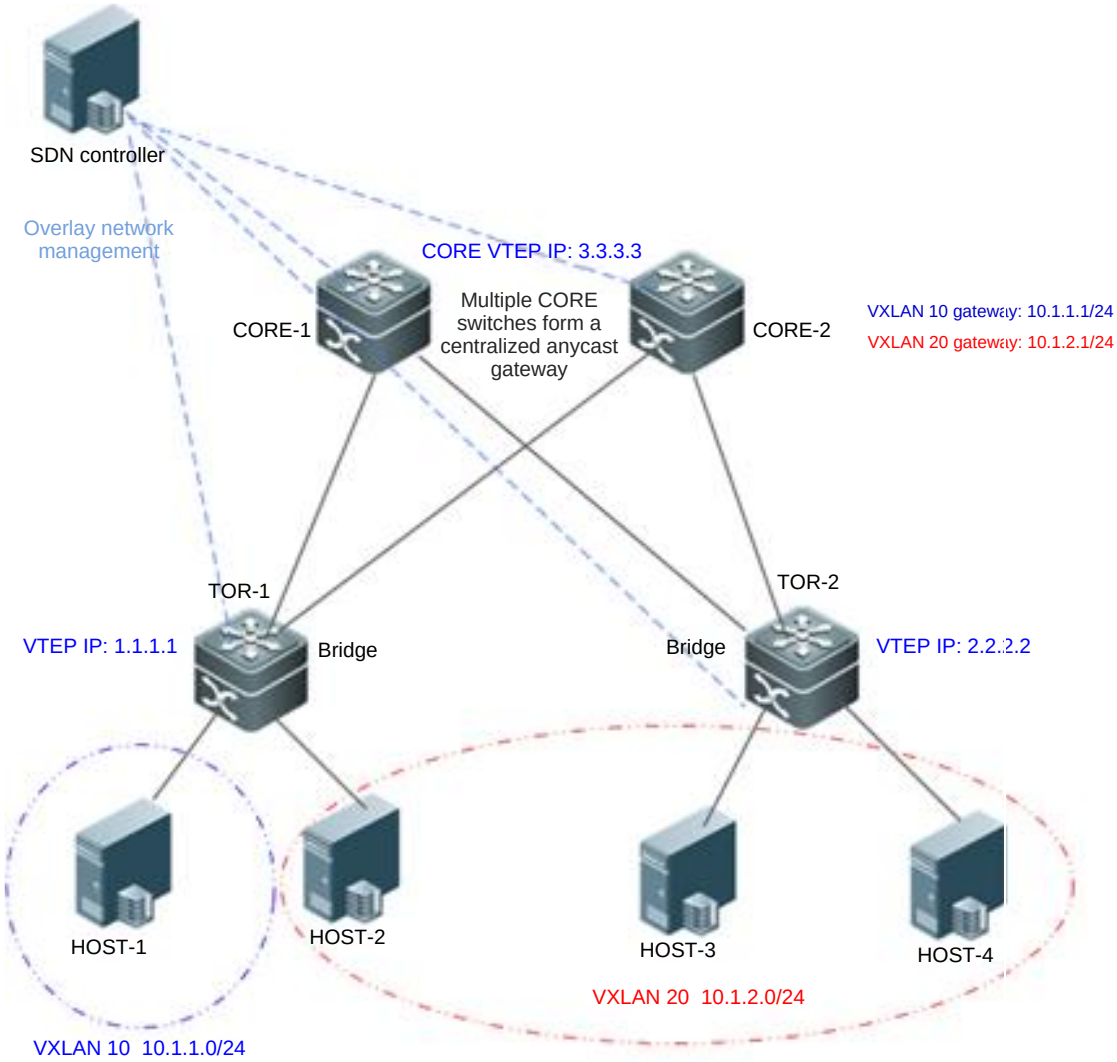

```
CORE1(config-router)#exit
CORE1(config)# interface OverlayTunnel 1
CORE1(config-if-OverlayTunnel 1)# tunnel source 3.3.3.3
CORE1(config-if-OverlayTunnel 1)# tunnel destination 2.2.2.2
CORE1(config-if-OverlayTunnel 1)# exit
CORE1(config)# interface OverlayTunnel 2
CORE1(config-if-OverlayTunnel 2)# tunnel source 3.3.3.3
CORE1(config-if-OverlayTunnel 2)# tunnel destination 1.1.1.1
CORE1(config-if-OverlayTunnel 2)# exit
CORE1(config)# interface overlayrouter 1
CORE1(config-if-OverlayRouter 1)# ip address 10.1.1.1/24
CORE1(config-if-OverlayRouter 1)# anycast-gateway
CORE1(config-if-OverlayRouter 1)# exit
CORE1(config)# interface overlayrouter 2
```

Scenario
Figure 1-22




```
CORE1(config-if-OverlayRouter 2)# ip address 10.1.2.1/24
CORE1(config-if-OverlayRouter 2)# anycast-gateway
CORE1(config-if-OverlayRouter 2)# exit
CORE1(config)# vxlan 10
CORE1(config-vxlan)# tunnel-interface OverlayTunnel 1
CORE1(config-vxlan)# tunnel-interface OverlayTunnel 2
CORE1(config-vxlan)# router-interface OverlayRouter 1
CORE1(config-vxlan)# end
CORE1(config)# vxlan 20
CORE1(config-vxlan)# tunnel-interface OverlayTunnel 1
CORE1(config-vxlan)# tunnel-interface OverlayTunnel 2
CORE1(config-vxlan)# router-interface OverlayRouter 2
CORE1(config-vxlan)# end
```

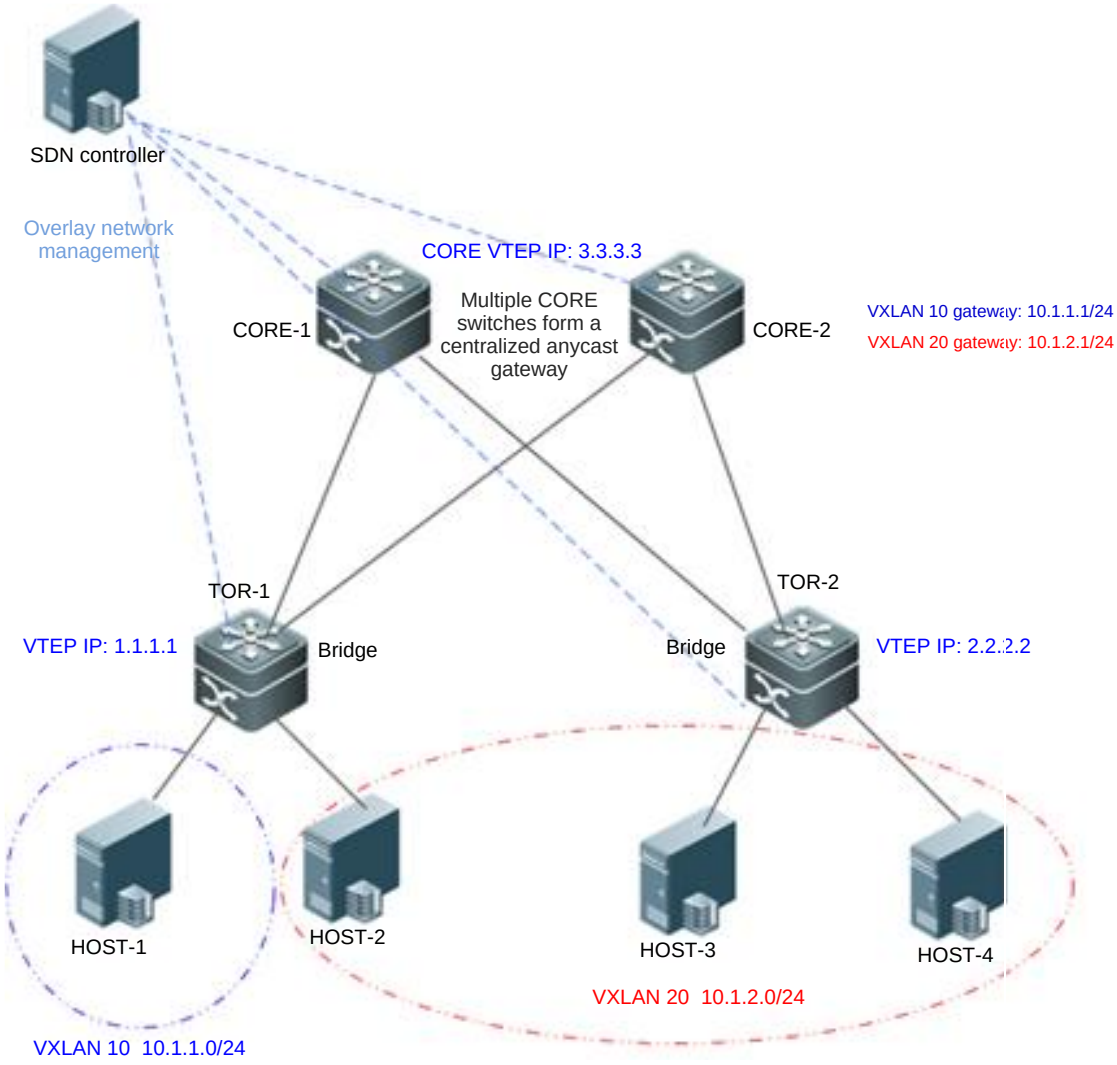
Scenario
Figure 1-22



CORE-2	Same with that of CORE-1
Verification	• Verify that the virtual machines of HOST-1, HOST-2, HOST-3, and HOST-4 can ping each other. • Display the static MAC entries of hosts delivered by the SDN controller on TOR and core switches.

```
TOR1(config)#show vxlan mac
Vxlan      MAC Address        Type    Location Interface                Vlan
---------- ------------------ -------- -------- ---------------
-------------- ------
    10    0000.0000.0001       STATIC    LOCAL    GigabitEthernet 0/1
10
    10  0000.1234.5678    STATIC  REMOTE  OverlayTunnel 2          -
    20    0000.0000.0002       STATIC    LOCAL    GigabitEthernet 0/2
20
```

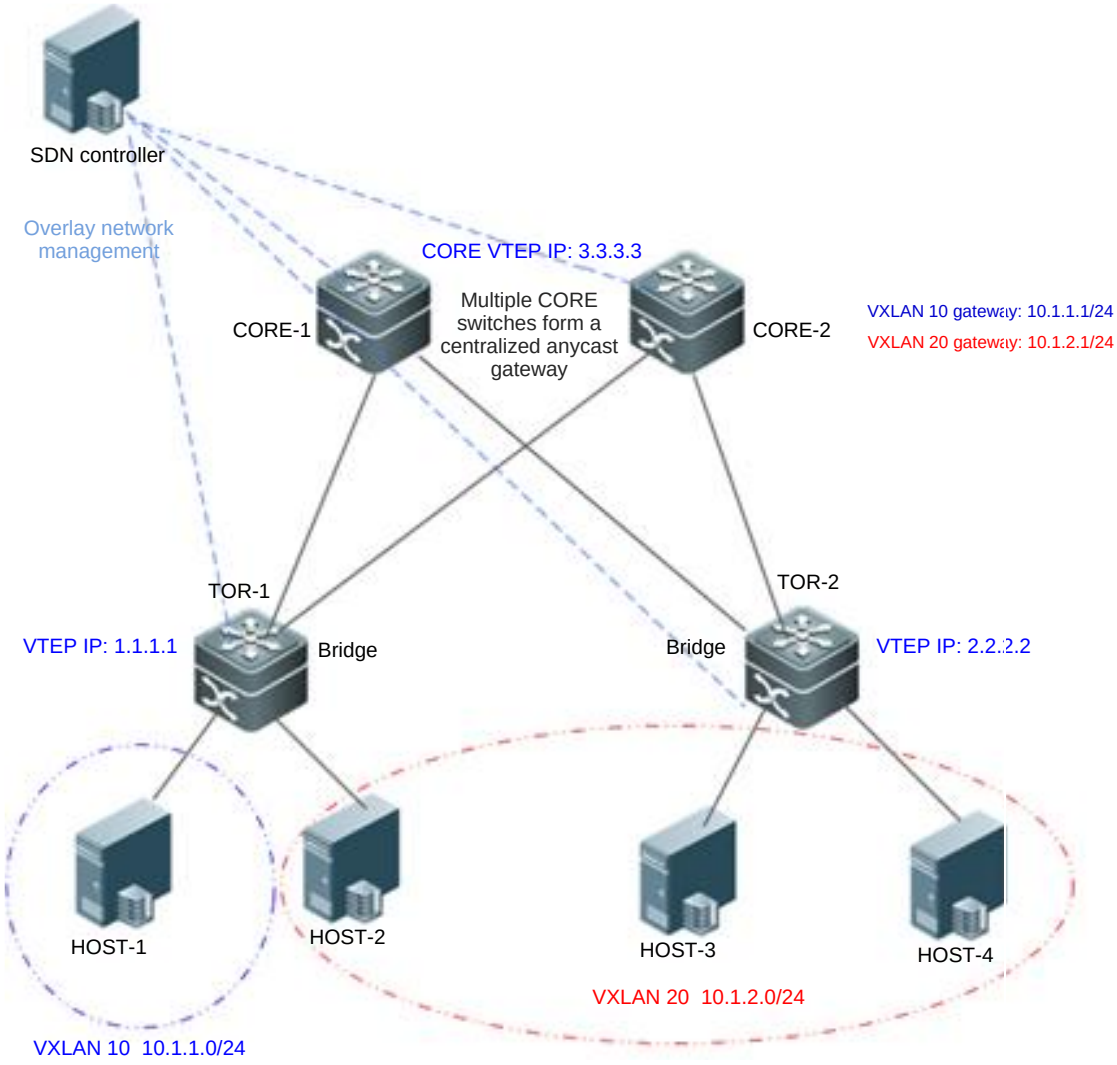
Scenario
Figure 1-22




```
    20  0000.0000.0003    STATIC  REMOTE  OverlayTunnel 1        -
    20  0000.0000.0004    STATIC  REMOTE  OverlayTunnel 1        -
    20  0000.1234.5678    STATIC  REMOTE  OverlayTunnel 2        -
```

```
TOR2(config)#show vxlan mac
Vxlan     MAC Address       Type    Location Interface              Vlan
----------  ------------------- -------- -------- ----------------
-------------- ------
    10  0000.0000.0001    STATIC  REMOTE  OverlayTunnel 1        -
    10  0000.1234.5678    STATIC  REMOTE  OverlayTunnel 2        -
    20  0000.0000.0002    STATIC  REMOTE  OverlayTunnel 1        -
    20   0000.0000.0003      STATIC   LOCAL    GigabitEthernet 0/1
20
    20   0000.0000.0004      STATIC   LOCAL    GigabitEthernet 0/2
```

Scenario
Figure 1-22



20

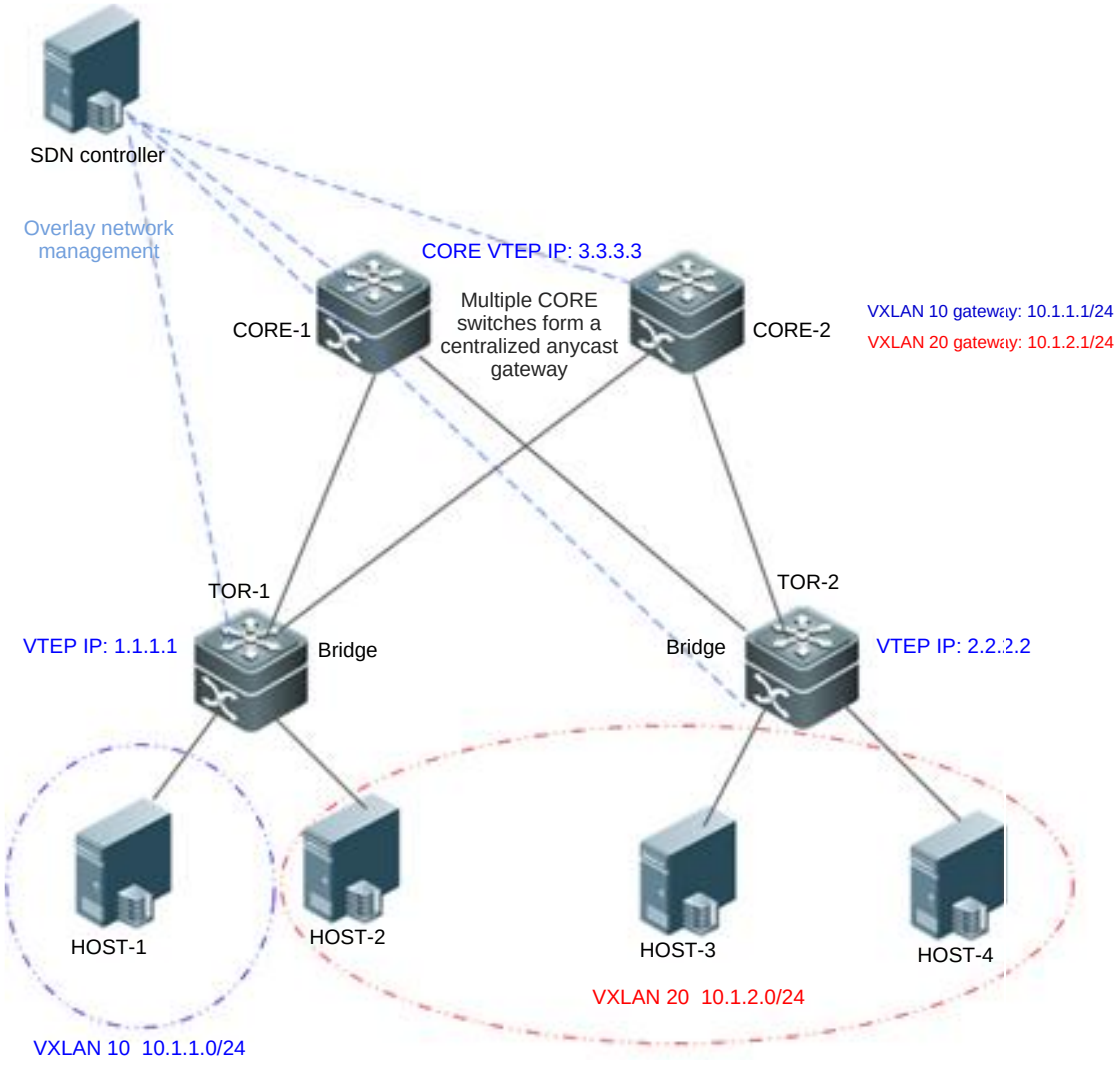
20 0000.1234.5678 STATIC REMOTE OverlayTunnel 2 -

CORE1#show vxlan mac

Vxlan	MAC Address	Type	Location	Interface	Vlan
10	0000.0000.0001	STATIC	REMOTE	OverlayTunnel 2	-
20	0000.0000.0002	STATIC	REMOTE	OverlayTunnel 2	-
20	0000.0000.0003	STATIC	REMOTE	OverlayTunnel 1	-
20	0000.0000.0004	STATIC	REMOTE	OverlayTunnel 1	-

CORE2#show vxlan mac

Scenario
Figure 1-22




```
Vxlan      MAC Address        Type    Location Interface                Vlan
---------- ------------------ -------- -------- ---------------
-------------- ------
    10  0000.0000.0001    STATIC  REMOTE  OverlayTunnel 2          -
    20  0000.0000.0002    STATIC  REMOTE  OverlayTunnel 2          -
    20  0000.0000.0003    STATIC  REMOTE  OverlayTunnel 1          -
    20  0000.0000.0004    STATIC  REMOTE  OverlayTunnel 1
```

1.4.2. Configuring VXLAN EVPN

Configuration Effect

- Enable the control plane learning function to implement VXLAN tunnel learning, MAC address learning, and route learning via control plane protocols, thereby

finally implementing VXLAN bridging, VXLAN routing, and data communication between VXLANs and between a VXLAN and an external network.

- Support functions such as anycast gateways, symmetric VXLAN instances, and ARP suppression in EVPN control plane mode.

Notes

- The VXLAN instances require support from existing unicast routes on the network. Therefore, an IPv4 unicast routing protocol, for example, the OSPF protocol must be configured on the network devices.
- The MP-BGP-EVPN protocol is required for VXLANs to implement VXLAN tunnel learning, MAC address learning, and route learning. Therefore, the devices on the network must complete BGP-related configurations.

Configuration Steps

Configuring Loopback Interface Associated with Local End

- Mandatory.
- Configure the IP address of a loopback interface as the IP address of the local VTEP. One VTEP device can be associated with only one loopback interface to serve as the VXLAN VTEP IP address.
- If the L3 egress is an overlay router interface during static route configuration, the next-hop IP address cannot be set to the VTEP IP address.

Configuring Virtual MAC Address for Anycast Gateways

- Optional.
- Configure a unified virtual MAC address for all anycast gateways on the network. The anycast function can be enabled on the VXLAN overlay router interface of the local device only after the virtual MAC address is configured.

Configuring ARP Suppression

- Optional.
- After ARP suppression is enabled, the switch responds to the ARP request from the host as a proxy, reducing the flooded ARP data.
- ARP suppression is generally enabled on the TOR bridge devices in a centralized deployment scenario, or on the distributed gateways in a distributed deployment scenario.

Configuring ARP Proxy

- Optional.
- After ARP suppression is enabled on a VTEP device, you can enable the ARP proxy function on an overlay router interface.
- After ARP proxy is enabled, the VTEP device responds to ARP requests from hosts as a proxy and the MAC address used for proxy response is the gateway MAC address configured on the VTEP device. In this way, the MAC address in the ARP request responses are the MAC address of the VTEP device, and the traffic between hosts in the same VNI is forwarded at L3.

ARP proxy can be enabled only on VXLAN gateways and is generally enabled on distributed gateways in distributed deployment scenarios.

Configuring IPv6 ND Suppression

- Optional.

- After IPv6 ND suppression is enabled, the VTEP device responds to NS multicast packets from hosts as a proxy, to reduce flooded NS multicast packets on the network.
- IPv6 ND suppression is generally enabled on distributed gateways in distributed deployment scenarios.

Configuring the EVPN Protocol Packet Control Function

- In symmetric EVPN deployment scenarios, the EVPN protocol packet control function can be configured on TOR switches to reduce the traffic of EVPN packets.

Currently, the EVPN protocol packet control function includes the following:

- Extracting MAC entries from EVPN MAC-IP type-2 routes (ARP entries) on a L2-VPN VXLAN instance
- Extracting MAC entries from EVPN MAC-IPv6 type-2 routes (IPv6 ND entries) on a L2-VNI VXLAN instance
- Banning synchronization of the local MAC address to the remote VTEP through EVPN messages on an L2-VNI VXLAN instance
- Banning delivery of the MAC addresses remotely synchronized through EVPN messages to the local MAC address table on an L2-VNI VXLAN instance
- Stopping an L2-VNI VXLAN instance from generating EVPN type-2 routes

Configuring Remote ARP Packet Learning

- Mandatory for centralized gateways and not recommended for other devices.
- After the remote ARP packet learning function is enabled, the VXLAN gateways can learn the VXLAN route entries from the encapsulated VXLAN ARP packets received from the VXLAN tunnels.

Configuring Remote IPv6 ND Protocol Packet Learning

- Mandatory for centralized gateways and not recommended for other devices.
- After the remote IPv6 ND protocol packet learning function is enabled, the VXLAN gateway can learn IPv6 ND entries from VXLAN-encapsulated IPv6 protocol packets received from VXLAN tunnels.

Creating VXLAN Instances

- Mandatory.

Associating VXLAN Instance with Overlay Router Interface

- Mandatory for VXLAN gateways.
- Only after the VXLAN is associated with the overlay router interface, the device can provide the VXLAN routing function and serve as a VXLAN IP gateway.

Associating VXLAN Instance with VLAN

- Mandatory for VXLAN devices directly connected to the host.
- Only after a VLAN is associated with a VXLAN instance, packets of the VLAN can be encapsulated into VXLAN packets and then forwarded.
- After a VLAN is associated with a VXLAN, all packets of the VLAN will be encapsulated into VXLAN packets. Therefore, an SVI cannot be used as the VLAN IP gateway on the device.

Configuring Storm Control of VXLAN Instances

- Optional.

- This function is required only when the storm rate needs to be limited based on VXLAN instances.

Configuring VXLAN UDP Destination Port

- Optional. As the VXLAN UDP destination port used by early devices may not be Port 4789, you can run this command to achieve compatibility. In addition, you can also run this command to customize the VXLAN UDP destination port.
- The VXLAN UDP destination port 4789 designated by IANA is used by default.

Configuring Symmetric Instances

- Optional.
- Symmetric instances need to be configured only in symmetric scenarios. Only one symmetric instance can be configured for each VRF network. After a symmetric instance is configured in a VRF network, L3 forwarding of other asymmetric instances is all switched to the symmetric instance for implementation.

Configuring Static VXLAN Network Routes

- Optional.
- Configure the static VXLAN network routes based on VXLAN instances if required.

Verification

Based on EVPN control plane learning, VXLAN tunnels, VXLAN MAC entries, and VXLAN route entries can be formed. Run the following commands for verification.

- Run the **show vxlan** *vni-number* command to check whether the local and remote VXLAN devices can learn mutual VTEP neighbor relationships.
- Run the **show vxlan mac** command to check whether the VXLAN MAC address is learned.
- Run the **show arp** command to check whether the ARP entry of the VXLAN IP gateway is learned.
- Run the **show ipv6 neighbors** command to check whether all local/remote IPv6 ND entries are learned.
- Run the **show vxlan udp-port** command to display the VXLAN UDP destination port.

Related Commands

Configuring Loopback Interface Associated with Local End

Command	**source** loopback *loopback-port-id*
Parameter Description	*Loopback-port-id*: Indicates the ID of the loopback interface.
Command Mode	VTEP configuration mode
Usage Guide	The local VETP IP address is the configured loopback interface IP address.

Configuring Virtual MAC Address for Anycast Gateways

Command	**fabric anycast-gateway-mac** *mac-addr*
Parameter Description	*mac-addr*: Indicates the MAC address. The format is xxxx.xxxx.xxxx.
Command Mode	Global configuration mode
Usage Guide	All gateways on which the anycast function is enabled use this MAC address as the gateway MAC address. The virtual MAC address for an anycast gateway must not be the same as the local MAC address or the same as the MAC address of any device on the overlay network.

Configuring Remote ARP Packet Learning

Command	**remote arp learn enable**
Parameter Description	N/A
Command Mode	VTEP configuration mode
Usage Guide	Enable or disable the remote ARP packet learning function globally. After this function is enabled, the VXLAN gateways can learn the VXLAN route entries from the encapsulated VXLAN ARP packets received from the VXLAN tunnels.

Configuring Remote IPv6 ND Protocol Packet Learning

Command	remote nd learn enable
Parameter Description	*N/A*
Command Mode	VTEP configuration mode
Usage Guide	Enable or disable the remote IPv6 ND packet learning function globally. After this function is enabled, the device can learn IPv6 ND entries from the VXLAN-encapsulated IPv6 NS packets received from VXLAN tunnels.

Configuring ARP Suppression

Command	arp suppress enable
Parameter Description	N/A
Command Mode	VTEP configuration mode
Usage Guide	Enable or disable ARP suppression globally. After ARP suppression is enabled, the switch responds to the ARP requests from the host as a proxy.The VNI-based ARP suppression may be also supported, depending on the product type. You can configure global ARP suppression or VNI-based ARP suppression based on the actual application scenario.

Configuring VNI-based ARP Suppression

Command	arp suppress enable
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	Enable or disable VNI-based ARP suppression. After ARP suppression is enabled, the switch responds to ARP requests from hosts as a proxy. The global ARP suppression may be also supported, depending on the product type. You can configure global ARP suppression or VNI-based ARP suppression based on the actual application scenario.

Configuring ARP Proxy

Command	route-in-vni
Parameter Description	*N/A*
Command Mode	Overlay router interface configuration mode
Usage Guide	After the intra-VNI routing function (ARP proxy) is enabled on an overlay router interface, the VTEP device uses its gateway MAC address to respond to all ARP requests from hosts in the VNI, to which the overlay router interface belongs, when serving as an ARP proxy. In this way, the communication traffic between hosts in the same VNI is forwarded through VXLAN routes.

Configuring Global IPv6 ND Suppression

Command	nd suppress enable
Parameter Description	*N/A*
Command Mode	VTEP configuration mode
Usage Guide	Enable or disable the global IPv6 ND suppression function. After IPv6 ND suppression is enabled, the device responds to IPv6 NS multicast packets from hosts as a proxy. The VNI-based IPv6 ND suppression may be also supported, depending on the product type. You can configure global IPv6 ND suppression or VNI-based IPv6 ND suppression based on the actual application scenario.

Configuring VNI-based IPv6 ND Suppression

Command	nd suppress enable
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	Enable or disable the VNI-based IPv6 ND suppression function. After IPv6 ND suppression is enabled, the device responds to IPv6 NS multicast packets from hosts as a proxy. The global IPv6 ND suppression may be also supported, depending on the product type. You can configure global IPv6 ND suppression or VNI-based IPv6 ND suppression based on the actual application scenario.

Extracting MAC Entries from EVPN MAC-IP Type-2 Routes (ARP Entries)

Command	**evpn arp mac-learning enable**
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	After this command is configured, the device parses one ARP entry and one MAC entry from a MAC-IP type-2 route synchronized from the VXLAN-EVPN neighbor. This command is disabled by default and the device parses one ARP entry but no MAC entry from a MAC-IP type-2 route synchronized from the VXLAN-EVPN neighbor. This command is configured on a VXLAN instance and affects only the EVPN entry parsing of the VXLAN instance. Other VXLAN instances, for which this command is not configured, are not affected. This command can be used in combination with the **evpn mac advertise disable** command. After they are executed, the network-wide VXLAN-EVPN neighbors synchronize only MAC-IP type-2 routes but no MAC-only type-2 routes. All devices parse and extract MAC entries from MAC-IP type-2 routes. This command is configured on L2-VNI VXLAN instances.

Extracting MAC Entries from EVPN MAC-IPv6 Type-2 Routes (IPv6 ND Entries)

Command	**evpn nd mac-learning enable**
Parameter Description	*N/A*
3	VXLAN configuration mode
Usage Guide	After this command is configured, the device parses one IPv6 ND entry and one MAC entry from a MAC-IPv6 type-2 route (IPv6 ND entry) synchronized from the VXLAN-EVPN neighbor. This command is disabled by default and the device parses one IPv6 ND entry but no MAC entry from a MAC-IPv6 type-2 route synchronized from the VXLAN-EVPN neighbor. This command is configured on a VXLAN instance and affects only the EVPN entry parsing of the VXLAN instance. Other VXLAN instances, for which this command is not configured, are not affected. This command can be used in combination with the **evpn mac advertise disable** command. After they are executed, the network-wide VXLAN-EVPN neighbors synchronize only MAC-IPv6 type-2 routes but no MAC-only type-2 routes. All devices parse and extract MAC entries from MAC-IPv6 type-2 routes. This command is configured on L2-VNI VXLAN instances.

Configuring an L2-VNI VXLAN Instance Not to Synchronize the Local MAC Address to the Remote VTEP Through EVPN Messages

Command	evpn mac advertise disable
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	This command is not configured on a device by default. The device generates one MAC-only type-2 route through the VXLAN-EVPN protocol based on a locally learned MAC entry, and synchronizes the type-2 route to the EVPN neighbor (that is, remote VTEP). Then, the remote VTEP can learn the MAC entry from the MAC-only type-2 route. After this command is configured, the device does not generate VXLAN-EVPN MAC-only type-2 routes based on MAC entries, and therefore, it will not advertise MAC-only type-2 routes to the EVPN neighbor. This command is configured on a VXLAN instance and affects only whether the VXLAN instance generates MAC-only type-2 routes. Other VXLAN instances, for which this command is not configured, can still generate MAC-only type-2 routes. This command can be used in combination with the **evpn arp mac-learning enable** and **evpn nd mac-learning enable** commands. After they are executed, the network-wide VXLAN-EVPN neighbors synchronize only MAC-IP type-2 routes but no MAC-only type-2 routes. All devices parse and extract MAC entries from MAC-IP or MAC-IPv6 type-2 routes. Note: This command can be configured only on L2-VNI VXLAN instances (that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

Configuring an L2-VNI VXLAN Instance Not to Deliver MAC Addresses Remotely Synchronized Through EVPN Messages to the Local MAC Address Table

Command	evpn mac inactive
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	After this command is configured, the device does not learn MAC entries from VXLAN-EVPN type-2 routes (MAC-IP or MAC-only type-2 routes) synchronized from neighbors. This command is not configured on a device by default. The device learns MAC entries from VXLAN-EVPN type-2 routes synchronized from neighbors. This command is configured on a VXLAN instance and affects only whether the VXLAN instance learns MAC entries from VXLAN-EVPN type-2 routes. Other VXLAN instances, for which this command is not configured, can still learn MAC entries. Note: This command can be configured only on L2-VNI VXLAN instances (that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

Configuring an L2-VNI VXLAN Instance Not to Generate EVPN Type-2 Routes	
Command	**evpn rt-2 advertise disable**
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	This command is not configured on a device by default. The device generates one MAC-only type-2 route through the VXLAN-EVPN protocol based on a locally learned MAC entry, and synchronizes the type-2 route to the EVPN neighbor (that is, remote VTEP). Then, the remote VTEP learns the MAC entry from the MAC-only type-2 route. In addition, the device generates one MAC-IP type-2 route through the VXLAN-EVPN protocol based on a locally learned ARP entry and synchronizes the type-2 route to the EVPN neighbor. Then, the remote VTEP learns the ARP entry and host route from the MAC-IP type-2 route. The device generates one MAC-IPv6 type-2 route through the VXLAN-EVPN protocol based on a locally learned IPv6 ND entry, and synchronizes the type-2 route to the EVPN neighbor. Then, the remote VTEP learns the IPv6 ND entry and host route from the MAC-IPv6 type-2 route. After this command is configured, the MAC entries, ARP entries, and IPv6 ND entries of the device are not used to generate VXLAN-EVPN type-2 routes and therefore, no type-2 route is advertised to the EVPN neighbor. This command is configured on a VXLAN instance and affects only whether the VXLAN instance generates type-2 routes. Other VXLAN instances, for which this command is not configured, can still generate type-2 routes. Note: This command can be configured only on L2-VNI VXLAN instances (that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

Creating Overlay Router Interfaces	
Command	**interface OverlayRouter** *port-id*
Parameter Description	*port-id*: Indicates the ID of an overlay router interface. The ID ranges from 1 to 16,777,215.
Command Mode	Global configuration mode
Usage Guide	Similar to SVI in a VLAN, this interface serves as the VXLAN IP gateway in the VXLAN routing environment.

Configuring IP Address for Overlay Router Interface	
Command	**ip address** *ip-address mask*
Parameter Description	*ip-address*: Indicates the IP address of the overlay router interface. *mask*: Indicates the subnet mask.
Command Mode	Interface configuration mode

Command	**ip address** *ip-address mask*
Usage Guide	Similar to the IP address of the SVI in a VLAN, this IP address serves as the address of the VXLAN IP gateway in the VXLAN routing environment.

Configuring an IPv6 Address for the Overlay Router Interface

Command	**ipv6 address** *ip-address mask*
Parameter Description	*ip-address*: Indicates the IPv6 address of the overlay router interface. *mask*: Indicates the subnet mask.
Command Mode	Overlay router interface configuration mode
Usage Guide	This IPv6 address serves as the VXLAN IPv6 gateway address in the VXLAN routing environment. It is similar to the IP address of an SVI in a VLAN.

Associating Overlay Router Interface with VRF Network

Command	**vrf forwarding** *table name*
Parameter Description	*Table name*: Indicates the VRF network associated with the overlay router interface.
Command Mode	Interface configuration mode
Usage Guide	Use this command to associate an overlay router interface with a VRF network in the VXLAN routing environment, to implement VXLAN L3 route isolation.

Creating or Entering VXLAN Instances

Command	**vxlan** *vni-number*
Parameter Description	*vni-number*: Indicates the VNI. The value ranges from 1 to 16777215.
Command Mode	Global configuration mode
Usage Guide	N/A

Configuring Symmetric Instances

Command	**symmetric**
Parameter Description	N/A
Command Mode	VXLAN configuration mode
Usage Guide	No symmetric instance is configured by default. Symmetric instances are used to manage the L3 forwarding entries of all asymmetric instances of the

Command	symmetric
	VRF networks associated with the symmetric instances.

Associating VXLAN Instance with Overlay Router Interface

Command	**router-interface** *interface-name*
Parameter Description	*interface-name*: Indicates the name of the overlay router interface.
Command Mode	VXLAN configuration mode
Usage Guide	Different VXLANs cannot be associated with the same overlay router interface.

Configuring VXLAN UDP Destination Port

Command	**vxlan udp-port** *port-number*
Parameter Description	*port-number*: Indicates the UDP destination port ID. The value ranges from 0 to 65535 and the default value is 4789.
Command Mode	Global configuration mode
Usage Guide	Note that the UDP destination port cannot be same as commonly used UDP ports.

Configuring Storm Control of VXLAN Instances

Command	**storm-control {broadcast | multicast | unicast} [***kbps-value* **| pps** *pps-value*]
Parameter Description	*kbps-value*: Indicates the rate limit value (unit: kbit/s). *pps-value*: Indicates the rate limit value (unit: packet count/s).
Command Mode	VXLAN configuration mode
Usage Guide	Configure the storm control when the storm rate needs to be limited based on the VNI.

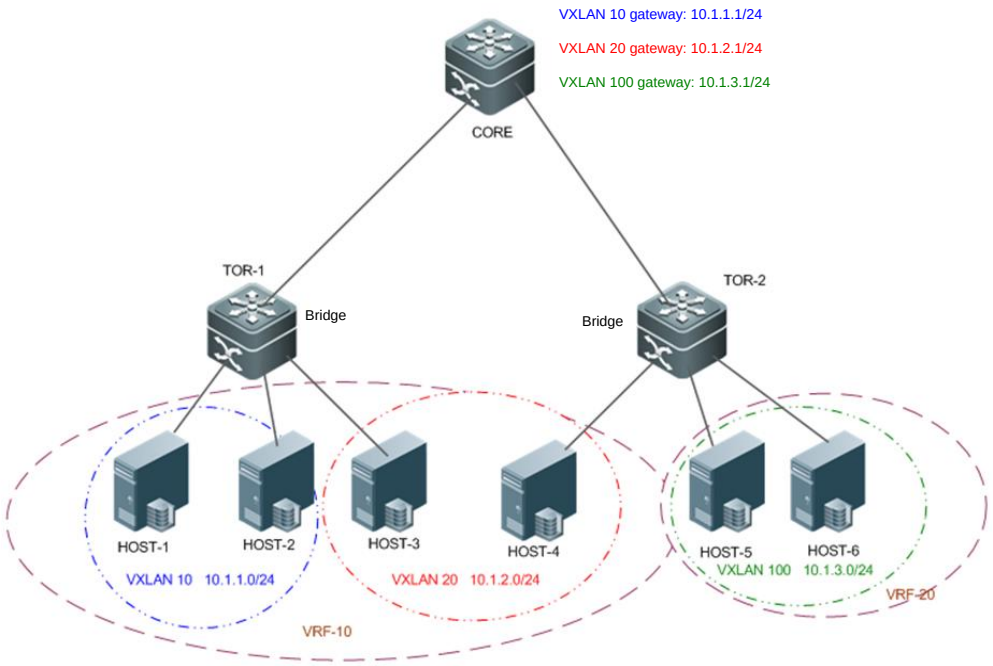
Configuration Example

Note:

- Only configuration related to the VXLAN is described below.
- Only IPv4 configuration is used as an example below and the IPv6 scenario configuration is largely the same as the IPv4 scenario configuration.

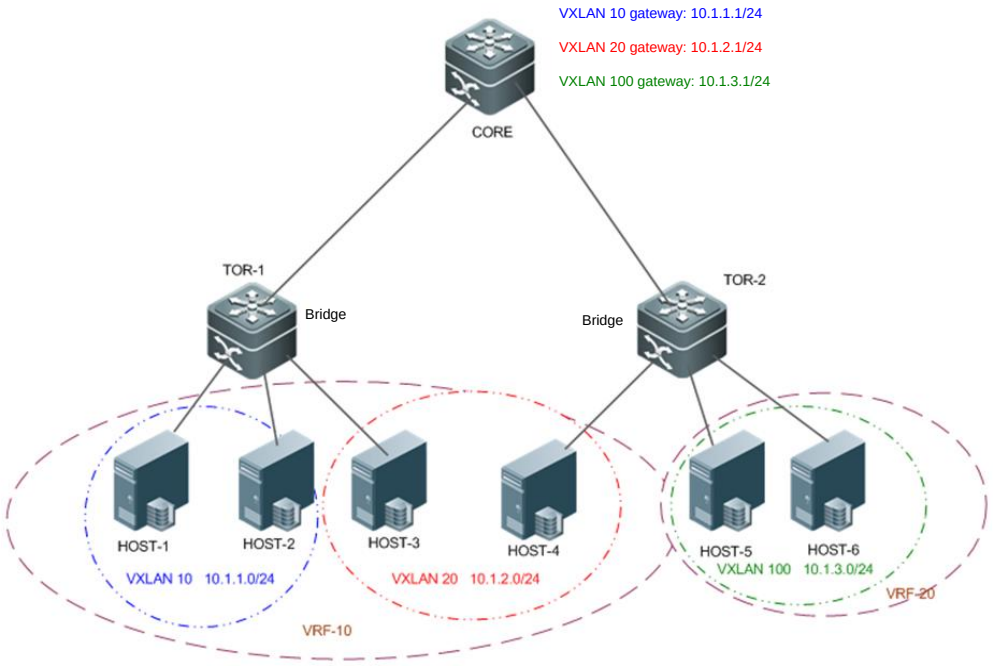
1.4.2.1. Configuring EVPN-based Multi-tenant Centralized Scenario

Scenario
Figure 1-23



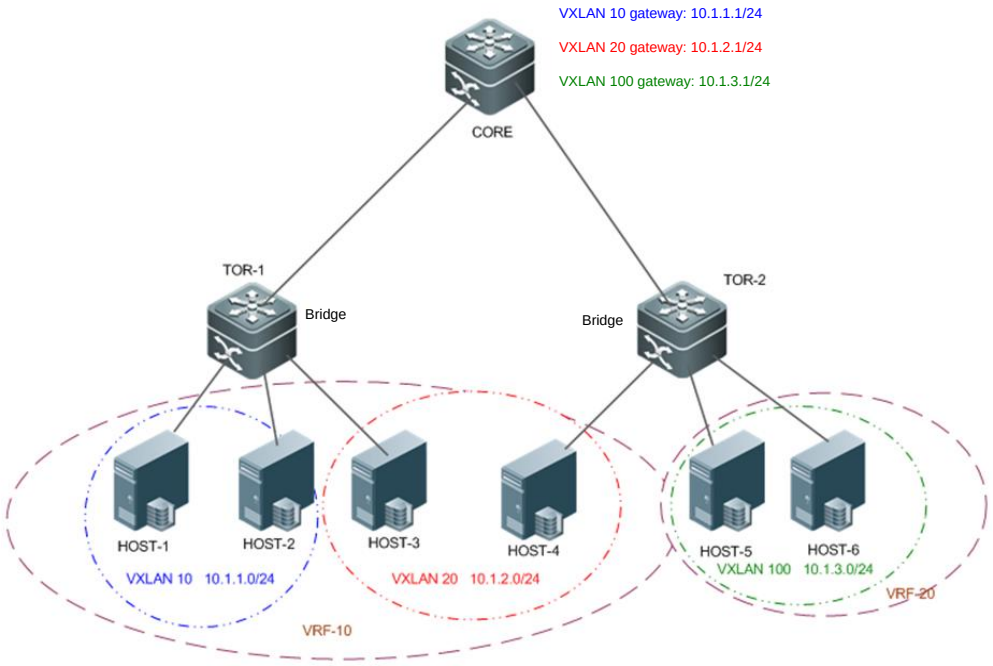
Configuration Steps	• Configure an IPv4 unicast routing protocol such as the OSPF protocol on CORE, TOR-1, and TOR-2 to ensure that unicast routes are reachable. • Configure the BGP-EVPN routing protocol on CORE, TOR-1, and TOR-2 to establish BGP neighbor relationships between the three devices and to support the EVPN protocol family. • Configure the EVI for BGP-EVPN on CORE, TOR-1, and TOR-2. For details, see *BGP-EVPN Configuration Guide*. • Configure a VXLAN on the virtual server and designate the gateway address of the virtual machine. • Associate the VTEP with the loopback interface on TOR-1, TOR-2, and CORE to establish tunnels. • Create VXLAN instances on TOR-1, TOR-2, and CORE and associate the VXLAN instances with VLANs. • Create overlay router interfaces and configure the VXLAN gateway IP address on CORE. Configure different VRF networks for different overlay router interfaces to determine their respective tenants. • Associate VXLAN instances with overlay router interfaces on CORE to realize VXLAN routing. • Enable the remote ARP packet learning function on CORE to generate VXLAN routing entries dynamically. • (Optional) Configure ARP suppression on TOR-1 and TOR-2 to reduce the ARP packets entering the VXLAN.
HOST	Configuring the IP address and gateway according to Figure 2-23 (the detailed configuration on the server is omitted herein).

Scenario
Figure 1-23



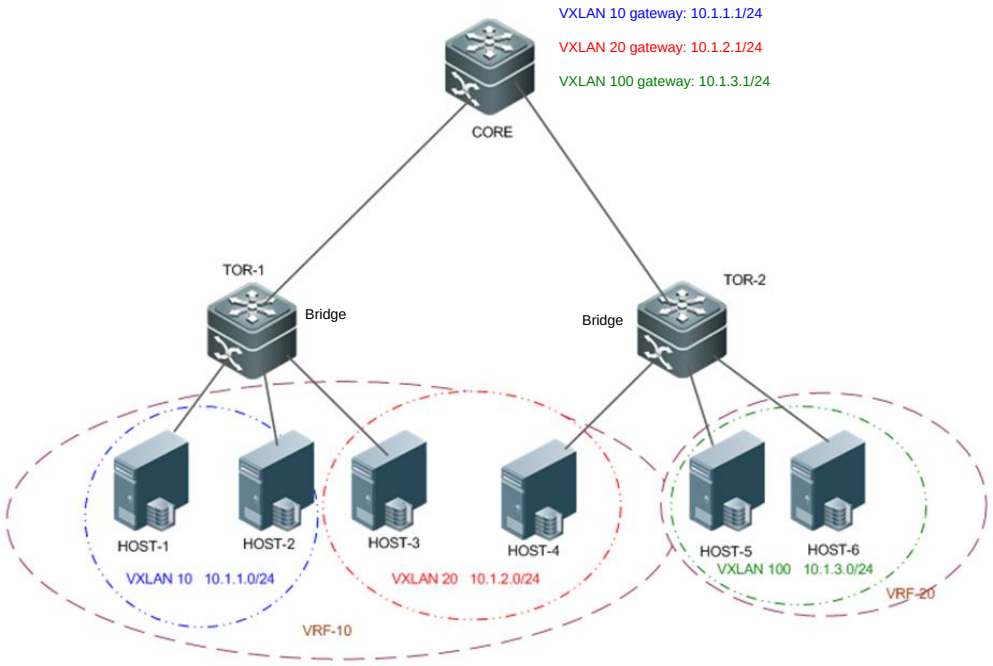
CORE	The configuration of the OSPF, and Ethernet interface is omitted herein. The following describe only the VXLAN configuration. CORE# configure terminal Enter configuration commands, one per line. End with CNTL/Z. CORE(config)# interface Loopback 1 CORE(config-if- Loopback 1)# ip address 1.1.1.1/32 CORE(config-if- Loopback 1)# exit CORE(config)# vtep CORE(config-vtep)# source loopback 1 CORE(config-vtep)# remote arp learn enable CORE(config-vtep)# exit CORE(config)# ip vrf vrf-10 CORE(config-vrf)# rd 10:10 CORE(config-vrf)# route-target both 1000:1000 CORE(config-vrf)# exit CORE(config)# ip vrf vrf-20 CORE(config-vrf)# rd 20:20 CORE(config-vrf)# route-target both 2000:2000 CORE(config-vrf)# exit CORE(config)# int overlayrouter 10 CORE(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10 CORE(config-if-OverlayRouter 10)# ip address 10.1.1.1/24

Scenario
Figure 1-23



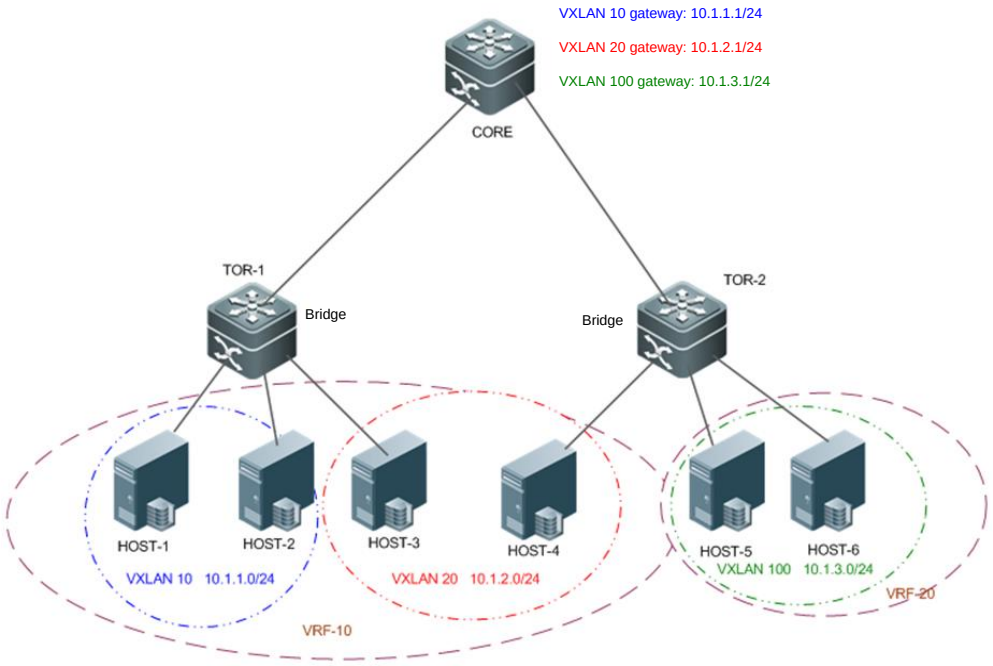

```
CORE(config-if-OverlayRouter 10)# exit
CORE(config)# int overlayrouter 20
CORE(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10
CORE(config-if-OverlayRouter 20)# ip address 10.1.2.1/24
CORE(config-if-OverlayRouter 20)# exit
CORE(config)# int overlayrouter 100
CORE(config-if-OverlayRouter 100)# ip vrf forwarding vrf-20
CORE(config-if-OverlayRouter 100)# ip address 10.1.3.1/24
CORE(config-if-OverlayRouter 100)# exit
CORE(config)# vxlan 10
CORE(config-vxlan)# router-interface OverlayRouter 10
CORE(config-vxlan)# exit
CORE(config)# vxlan 20
CORE(config-vxlan)# router-interface OverlayRouter 20
CORE(config-vxlan)# exit
CORE(config)# vxlan 100
CORE(config)# vxlan 100
CORE(config-vxlan)# router-interface OverlayRouter 100
CORE(config-vxlan)# exit
CORE(config)# router bgp 64512
CORE(config-router)# neighbor 2.2.2.2 remote-as 64512
CORE(config-router)# neighbor 3.3.3.3 remote-as 64512
```

Scenario
Figure 1-23



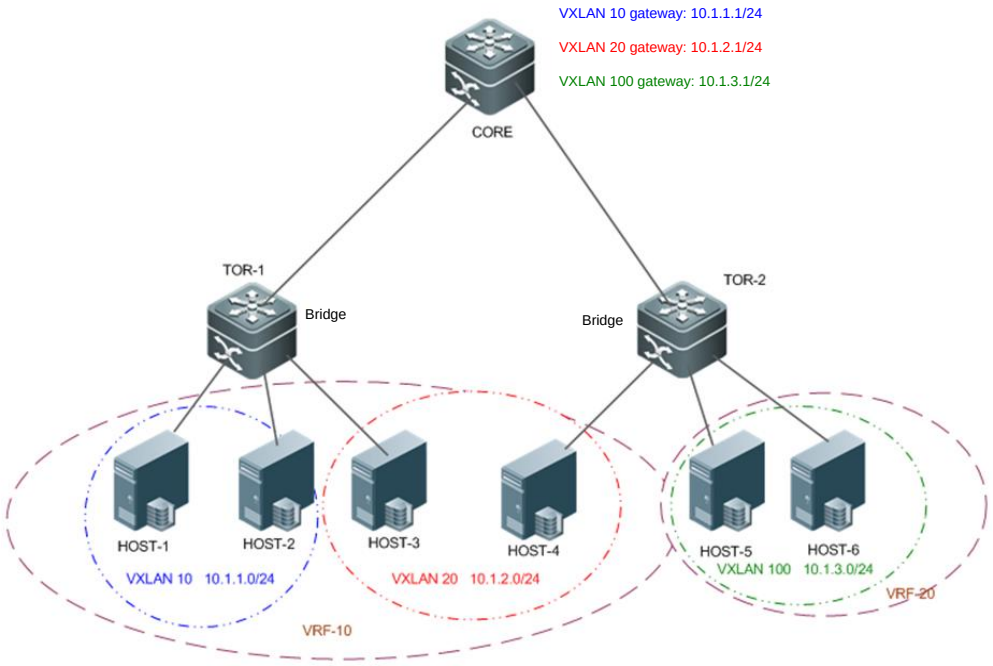

```
CORE(config-router)# neighbor 2.2.2.2 update-source Loopback 1
CORE(config-router)# neighbor 3.3.3.3 update-source Loopback 1
CORE(config-router)# address-family l2vpn evpn
CORE(config-router-af)# neighbor 2.2.2.2 activate
CORE(config-router-af)# neighbor 3.3.3.3 activate
CORE(config-router-af)# neighbor 2.2.2.2 route-reflector-client
CORE(config-router-af)# neighbor 3.3.3.3 route-reflector-client
CORE(config-router-af)# exit
CORE(config-router)# address-family ipv4 vrf vrf-10
CORE(config-router-af)# network 10.1.1.0 mask 255.255.255.0
CORE(config-router-af)# network 10.1.2.0 mask 255.255.255.0
CORE(config-router-af)# exit
CORE(config-router)# address-family ipv4 vrf vrf-20
CORE(config-router-af)# network 10.1.3.0 mask 255.255.255.0
CORE(config-router-af)# exit
CORE(config-router)# exit
CORE(config)# evpn
CORE(config-evpn)# vni 10
CORE(config-evpn-vni)# rd auto
CORE(config-evpn-vni)# route-target both auto
CORE(config-evpn-vni)# exit
CORE(config-evpn)# vni 20
```

Scenario
Figure 1-23



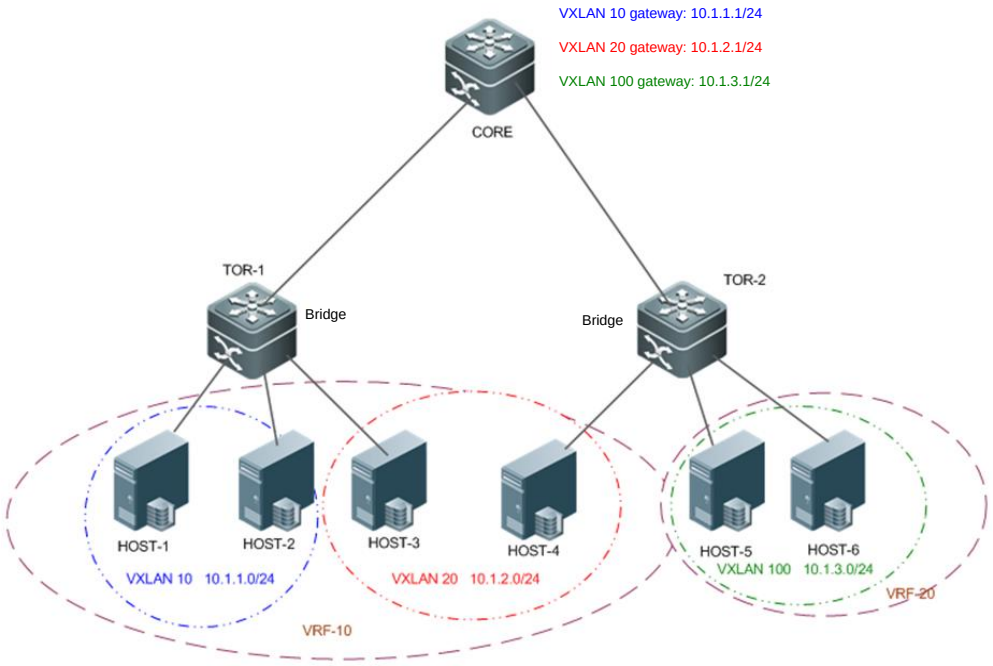
	CORE(config-evpn-vni)# rd auto CORE(config-evpn-vni)# route-target both auto CORE(config-evpn-vni)# exit CORE(config-evpn)# vni 100 CORE(config-evpn-vni)# rd auto CORE(config-evpn-vni)# route-target both auto CORE(config-evpn-vni)# exit
TOR1	TOR1# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR1(config)# interface Loopback 1 TOR1(config-if- Loopback 1)# ip address 2.2.2.2/32 TOR1(config-if- Loopback 1)# exit TOR1(config)# vtep TOR1(config-vtep)# source loopback 1 TOR1(config-vtep)# arp suppress enable TOR1(config-vtep)# exit TOR1(config)# vxlan 10 TOR1(config-vxlan)# extend-vlan 10 TOR1(config-vxlan)# arp suppress enable TOR1(config-vxlan)# exit TOR1(config)# vxlan 20

Scenario
Figure 1-23

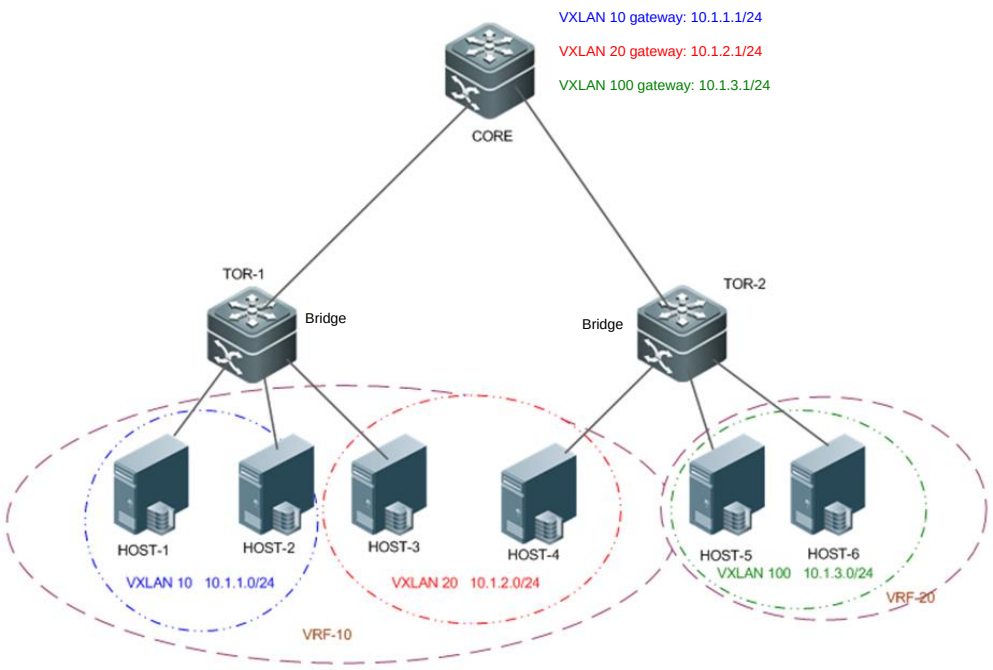



```
TOR1(config-vxlan)# extend-vlan 20
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# router bgp 64512
TOR1(config-router)# neighbor 1.1.1.1 remote-as 64512
TOR1(config-router)# neighbor 1.1.1.1 update-source loopback 1
TOR1(config-router)# address-family l2vpn evpn
TOR1(config-router-af)# neighbor 1.1.1.1 activate
TOR1(config-router-af) #exit
TOR1(config-router)# exit
TOR1(config)# evpn
TOR1(config-evpn)# vni 10
TOR1(config-evpn-vni)# rd auto
TOR1(config-evpn-vni)# route-target both auto
TOR1(config-evpn-vni)# exit
TOR1(config-evpn)# vni 20
TOR1(config-evpn-vni)# rd auto
TOR1(config-evpn-vni)# route-target both auto
TOR1(config-evpn-vni)# exit
```

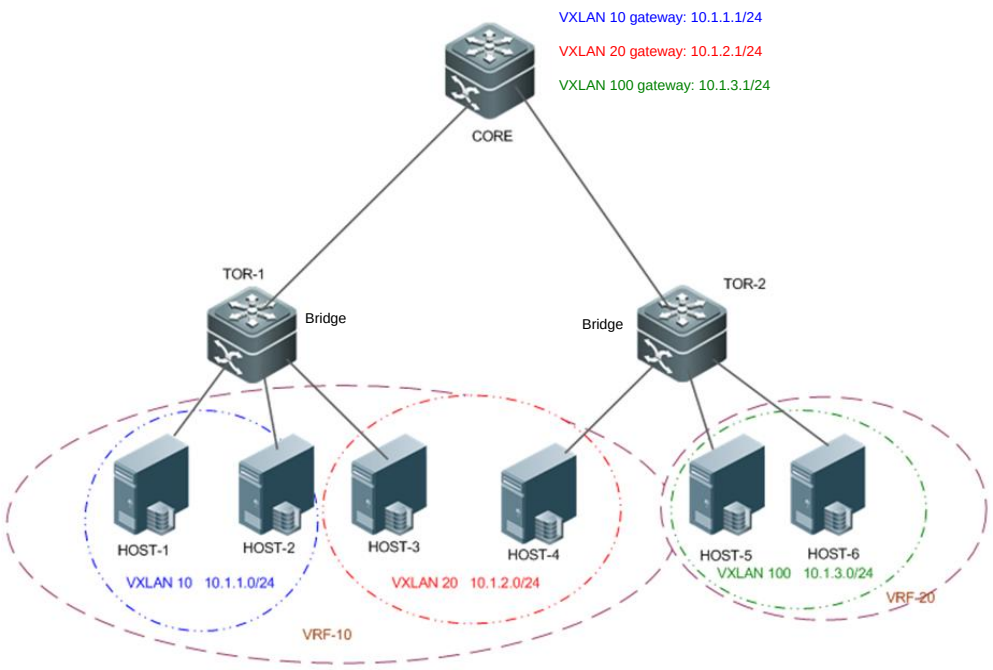
Scenario
Figure 1-23



TOR2	TOR2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR2(config)# interface Loopback 1 TOR2(config-if- Loopback 1)# ip address 3.3.3.3/32 TOR2(config-if- Loopback 1)# exit TOR2(config)# vtep TOR2(config-vtep)# source loopback 1 TOR2(config-vtep)# arp suppress enable TOR2(config-vtep)# exit TOR2(config)# vxlan 100 TOR2(config-vxlan)# extend-vlan 100 TOR2(config-vxlan)# arp suppress enable TOR2(config-vxlan)# exit TOR2(config)# vxlan 20 TOR2(config-vxlan)# extend-vlan 20 TOR2(config-vxlan)# arp suppress enable TOR2(config-vxlan)# exit TOR2(config)# router bgp 64512 TOR2(config-router)# neighbor 1.1.1.1 remote-as 64512 TOR2(config-router)# neighbor 1.1.1.1 update-source loopback 1 TOR2(config-router)# address-family l2vpn evpn TOR2(config-router)# neighbor 1.1.1.1 activate

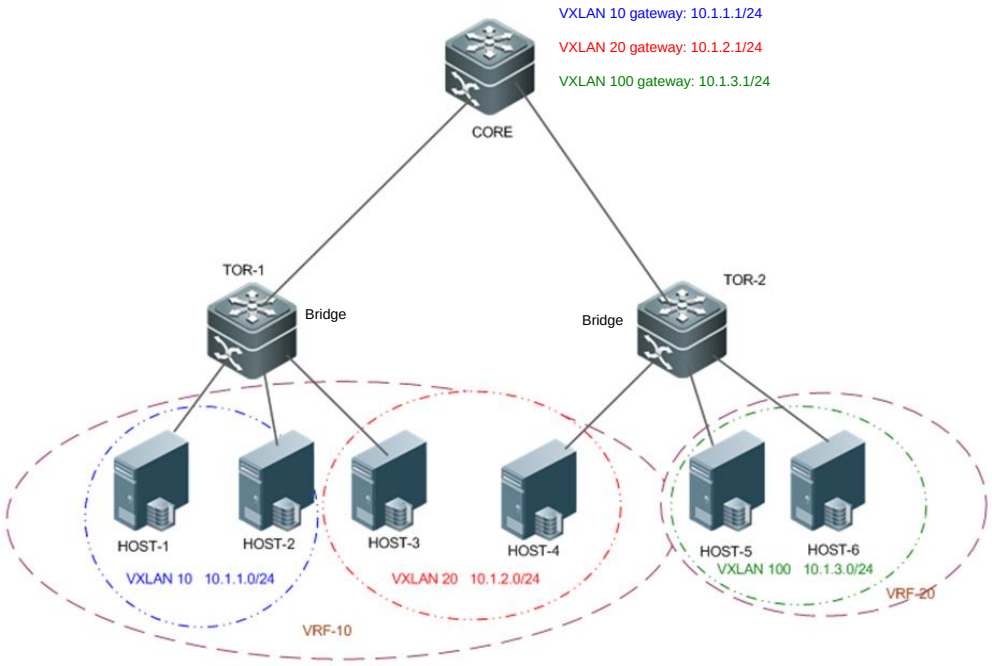
Scenario **Figure 1-23**	
	TOR2(config-router-af)# exit TOR2(config-router)# exit TOR2(config)# evpn TOR2(config-evpn)# vni 20 TOR2(config-evpn-vni)# rd auto TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# exit TOR2(config-evpn)# vni 100 TOR2(config-evpn-vni)# rd auto TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# exit
Verification	• Verify that HOST-1, HOST-2, HOST-3, and HOST-4 can ping each other. • Verify that HOST-5 and HOST-6 can ping each other. • Verify that HOST-1, HOST-2, HOST-3, and HOST-4 cannot ping HOST-5 and HOST-6. • Verify that the virtual machines can be migrated between the hosts on the same VXLAN and can access the network normally after migration without modifying the configuration.

Scenario
Figure 1-23



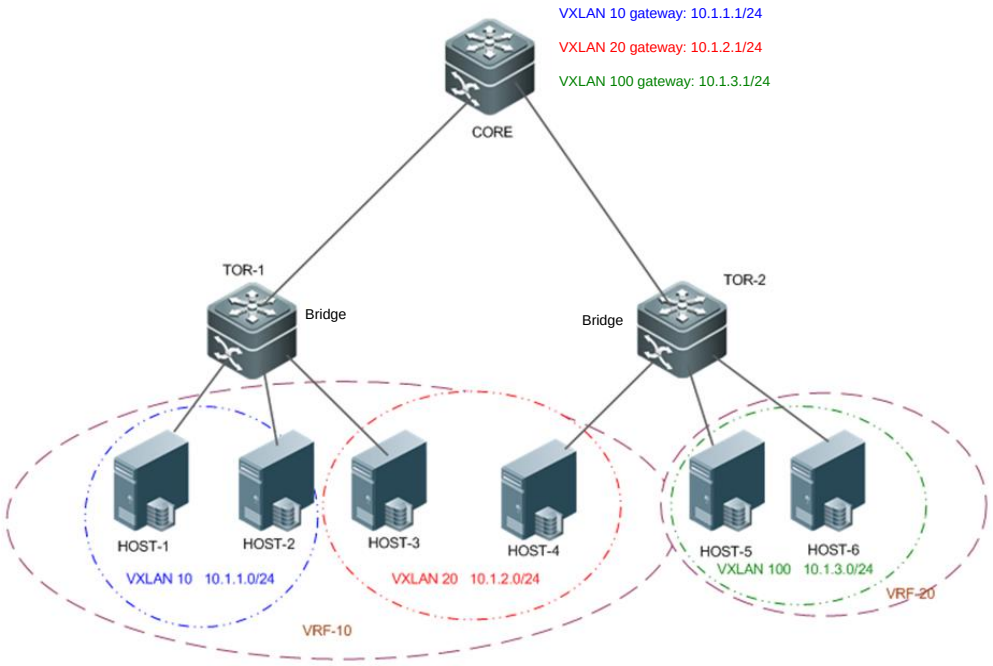

```
TOR1#sho vxlan
VXLAN Total Count: 2
VXLAN Capacity   : 8000

VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : -
   Extend VLAN         : 10
   VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6145   2.2.2.2        1.1.1.1         dynamic

VXLAN 20
   Symmetric property  : FALSE
   Router Interface    : -
   Extend VLAN         : 20
   VTEP Adjacency Count: 2
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
```

Scenario
Figure 1-23




```
   ---------------------- --------------- ---------------- -------
   OverlayTunnel 6145     2.2.2.2         1.1.1.1          dynamic
   OverlayTunnel 6146     2.2.2.2         3.3.3.3          dynamic


CORE#sho vxlan
VXLAN Total Count: 3
VXLAN Capacity   : 8000


VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 10 (non-anycast)
   Extend VLAN         : -
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface            Source IP       Destination IP  Type
   ---------------------- --------------- ---------------- -------
   OverlayTunnel 6147     1.1.1.1         2.2.2.2          dynamic


VXLAN 20
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 20 (non-anycast)
```

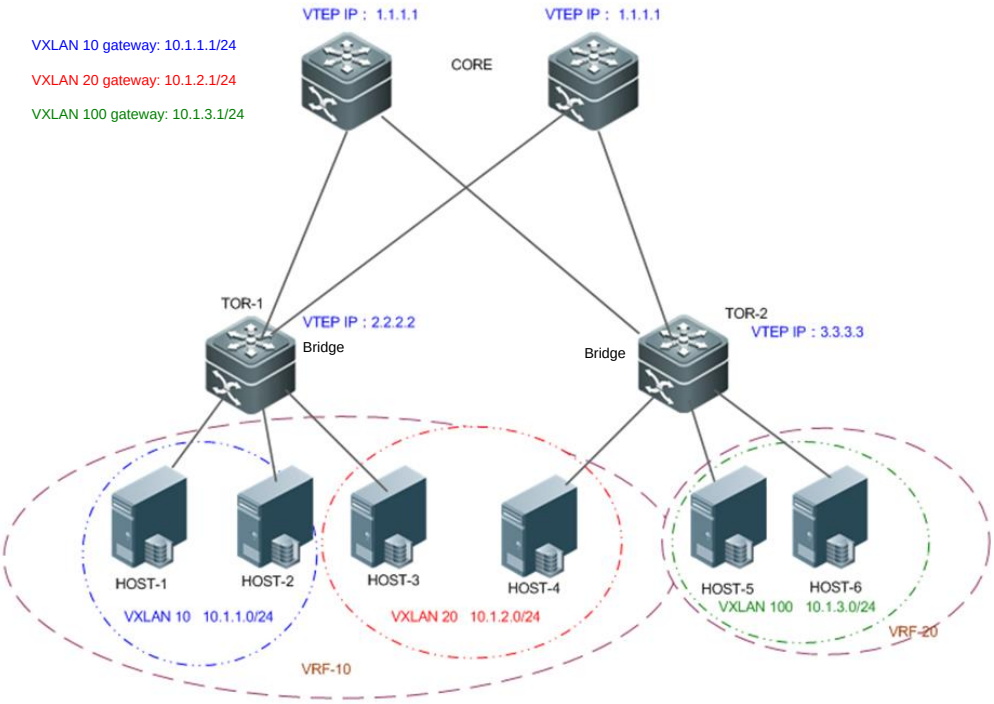
Scenario
Figure 1-23




```
   Extend VLAN        : -
VTEP Adjacency Count: 2
VTEP Adjacency List :
Interface            Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6147    1.1.1.1        2.2.2.2        dynamic
OverlayTunnel 6148     1.1.1.1        3.3.3.3        dynamic


VXLAN 100
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 100 (non-anycast)
   Extend VLAN        : -
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface            Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6148     1.1.1.1        3.3.3.3        dynamic
```

1.4.2.2. Configuring EVPN-based Multi-tenant Centralized All-active Anycast Gateway Scenario

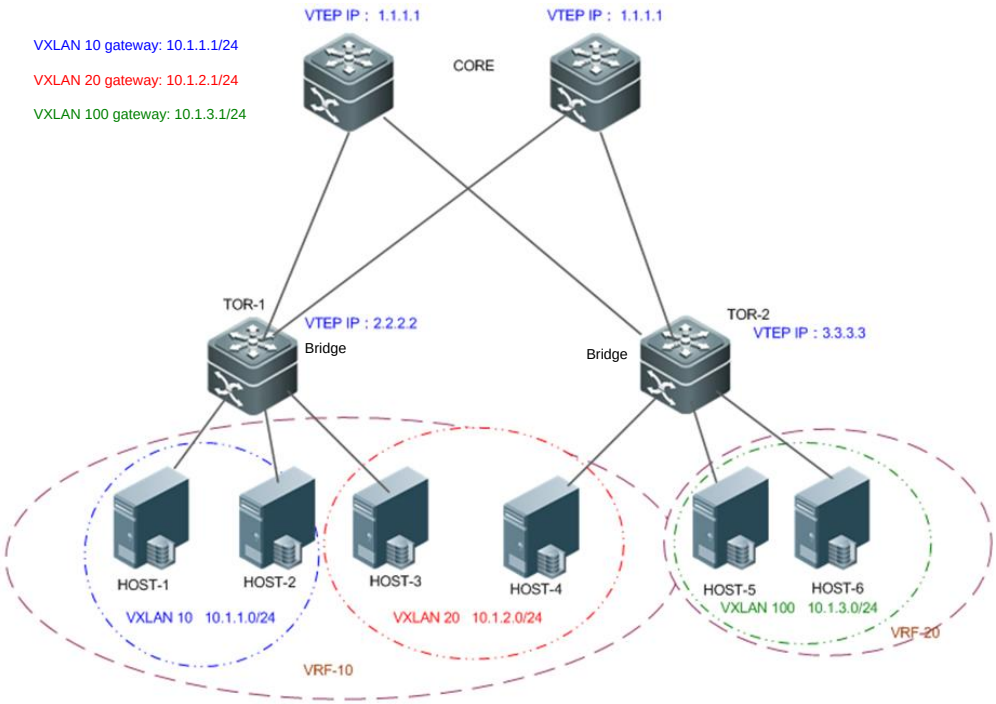
Scenario
Figure 1-24



Configuration Steps

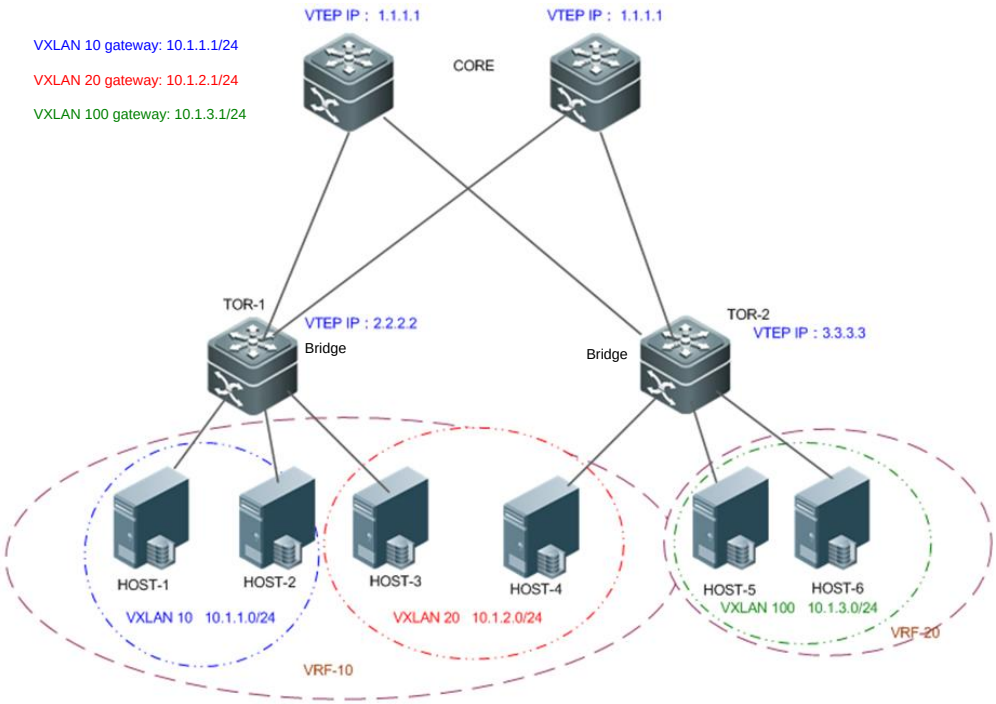
- Configure an IPv4 unicast routing protocol such as the OSPF protocol on CORE, TOR-1, and TOR-2 to ensure that unicast routes are reachable.
- Configure the BGP-EVPN routing protocol on CORE, TOR-1, and TOR-2 to establish BGP neighbor relationships between the three devices and to support the EVPN protocol family.
- Configure the EVI for BGP-EVPN on CORE, TOR-1, and TOR-2. For details, see *BGP-EVPN Configuration Guide*.
- Configure a VXLAN on the virtual server and designate the gateway address of the virtual machine.(Omitted).
- Associate the VTEP with the loopback interface on TOR-1, TOR-2, and CORE to establish tunnels.
- Note that the same loopback interface IP address needs to be configured on CORE1 and CORE2 as the VTEP IP address for tunnel establishment.
- After the loopback interface IP address is configured, no tunnel is established between CORE1 and CORE2.
- On TOR1, only one tunnel whose VTEP IP address is 1.1.1.1 and one tunnel whose VTEP IP address is 3.3.3.3 can be viewed.
- On TOR2, only one tunnel whose VTEP IP address is 1.1.1.1 and one tunnel whose VTEP IP address is 2.2.2.2 can be viewed.
- On CORE1, only one tunnel whose VTEP IP address is 2.2.2.2 and one tunnel whose VTEP IP address is 3.3.3.3 can be viewed.
- On CORE2, only one tunnel whose VTEP IP address is 2.2.2.2 and

Scenario
Figure 1-24



	one tunnel whose VTEP IP address is 3.3.3.3 can be viewed. • Create VXLAN instances on TOR-1, TOR-2, and CORE and associate the VXLAN instances with VLANs. • Create overlay router interfaces and configure the VXLAN gateway IP address on CORE. Configure different VRF networks for different overlay router interfaces to determine their respective tenants. Associate VXLAN instances with overlay router interfaces to realize VXLAN routing. • Note that the overlay router interface configurations on CORE1 and CORE2 must be the same. That is, the IP addresses and masks configured for the overlay router interfaces associated with the same VXLAN instance must be the same on CORE1 and CORE2 and belong to the same tenant (VRF). • In addition, an anycast gateway must be configured for all overlay router interfaces. • Configure the same anycast gateway MAC address on CORE1 and CORE2 to ensure that all VXLAN anycast gateways on CORE use the same MAC address. • Enable the remote ARP packet learning function on CORE to generate VXLAN routing entries dynamically. • (Optional) Configure ARP suppression on TOR-1 and TOR-2 to reduce the ARP packets entering the VXLAN.
HOST	Configure the IP address and gateway according to Figure 2-24 (the detailed configuration on the server is omitted herein).
CORE	The configuration of the OSPF and Ethernet interface is omitted herein. The following describes only the VXLAN configuration.

Scenario
Figure 1-24

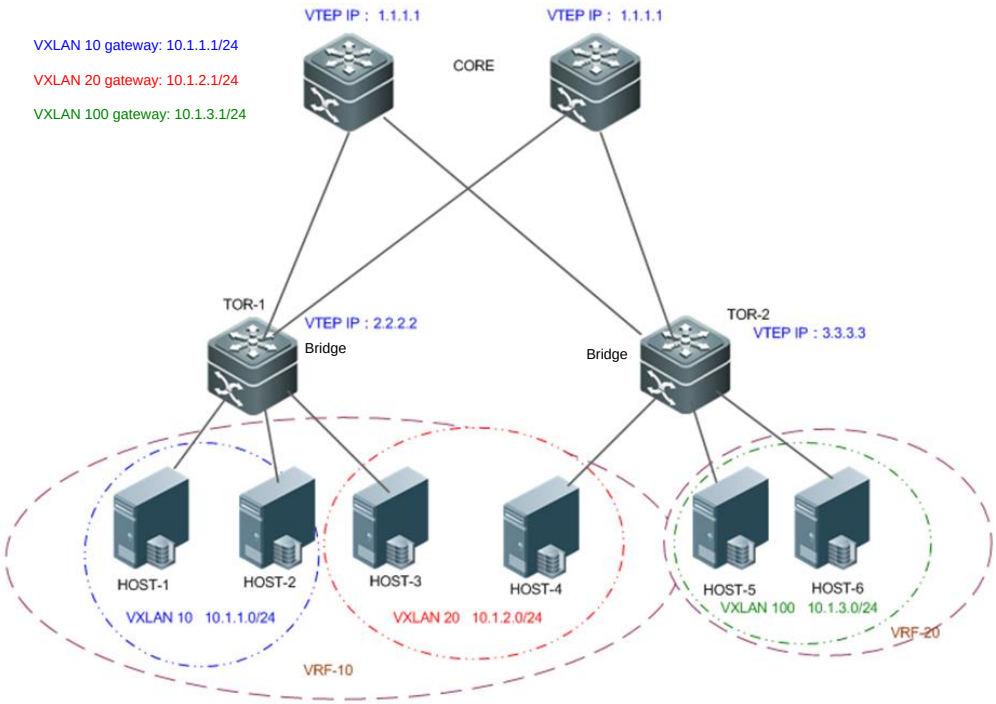


Note:

VXLAN configuration on CORE1 and CORE2 is the same. The following configuration applies to CORE1 and CORE2: Configure loopback 1 on both CORE1 and CORE2 and set the IP address to 1.1.1.2 and 1.1.1.3 for loopback 1. The two core switches establish a BGP neighbor relationship through loopback 1. Configure the L2VPN EVPN address family activation command.

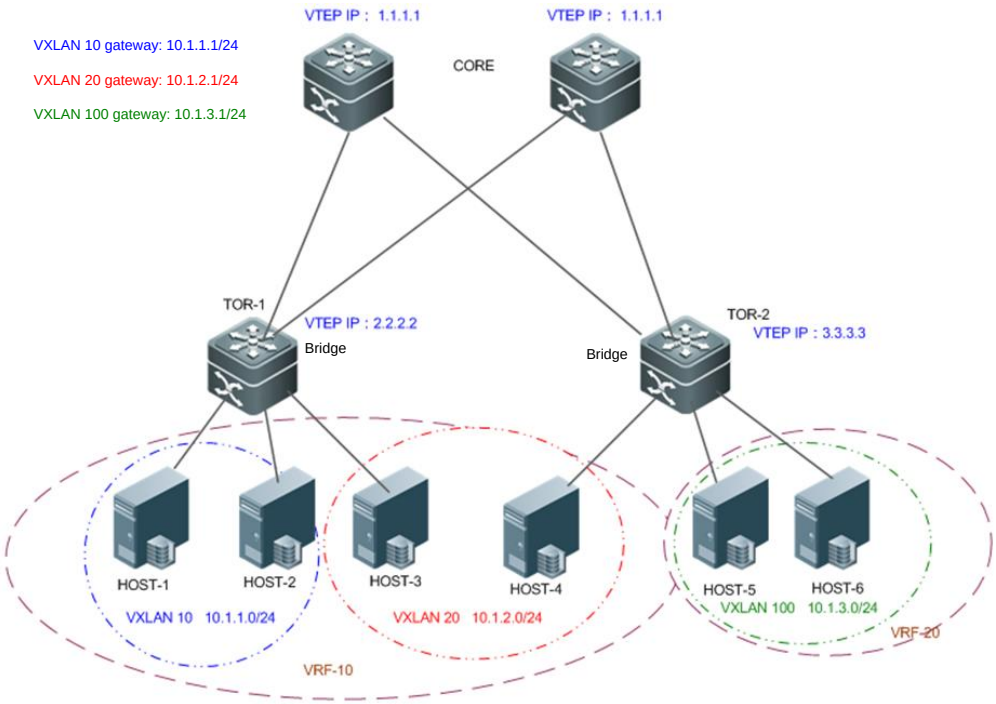
```
CORE# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
CORE(config)# interface Loopback 0
CORE(config-if- Loopback 0)# ip address 1.1.1.1/32
CORE(config-if- Loopback 0)# exit
CORE(config)#int loopback 1
CORE(config-if-Loopback 1)# ip address 1.1.1.2/32
CORE(config-if- Loopback 1)# exit
CORE(config)# vtep
CORE(config-vtep)# source loopback 0
CORE(config-vtep)# remote arp learn enable
CORE(config-vtep)# exit
CORE(config)# fabric anycast-gateway-mac 0011.2233.2016
CORE(config)# ip vrf vrf-10
CORE(config-vrf)# rd 10:10
CORE(config-vrf)# route-target both 1000:1000
```

Scenario Figure 1-24



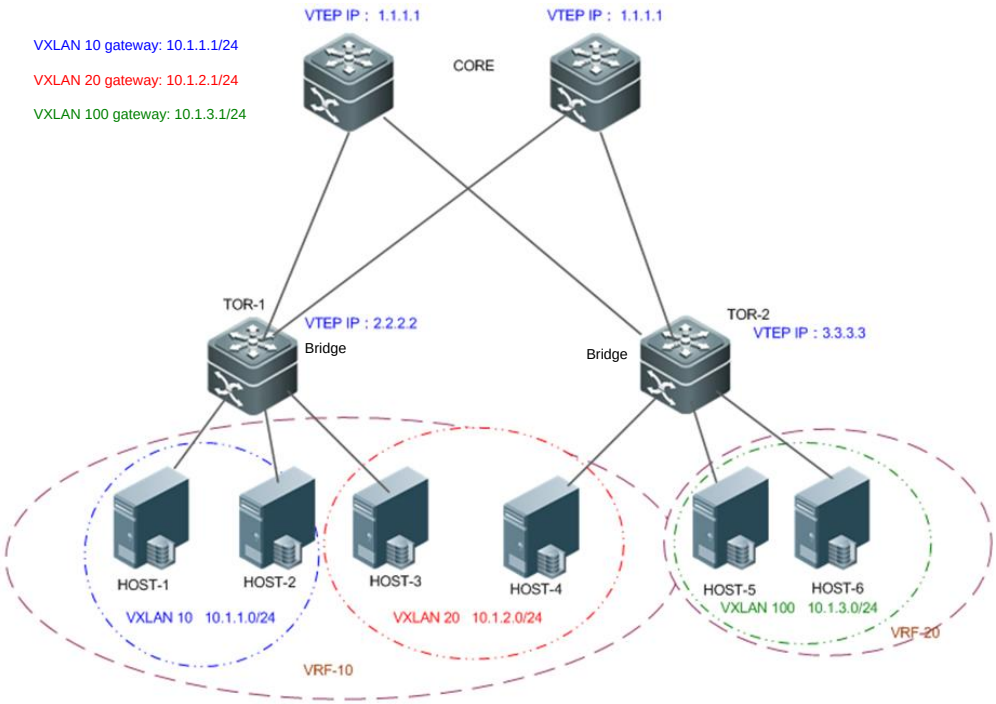

```
CORE(config-vrf)# exit
CORE(config)# ip vrf vrf-20
CORE(config-vrf)# rd 20:20
CORE(config-vrf)# route-target both 2000:2000
CORE(config-vrf)# exit
CORE(config)# int overlayrouter 10
CORE(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10
CORE(config-if-OverlayRouter 10)# ip address 10.1.1.1/24
CORE(config-if-OverlayRouter 10)# anycast-gateway
CORE(config-if-OverlayRouter 10)# exit
CORE(config)# int overlayrouter 20
CORE(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10
CORE(config-if-OverlayRouter 20)# ip address 10.1.2.1/24
CORE(config-if-OverlayRouter 20)# anycast-gateway
CORE(config-if-OverlayRouter 20)# exit
CORE(config)# vxlan 20
CORE(config)# int overlayrouter 100
CORE(config-if-OverlayRouter 100)# ip vrf forwarding vrf-20
CORE(config-if-OverlayRouter 100)# ip address 10.1.3.1/24
CORE(config-if-OverlayRouter 100)# anycast-gateway
CORE(config-if-OverlayRouter 100)# exit
```

Scenario Figure 1-24



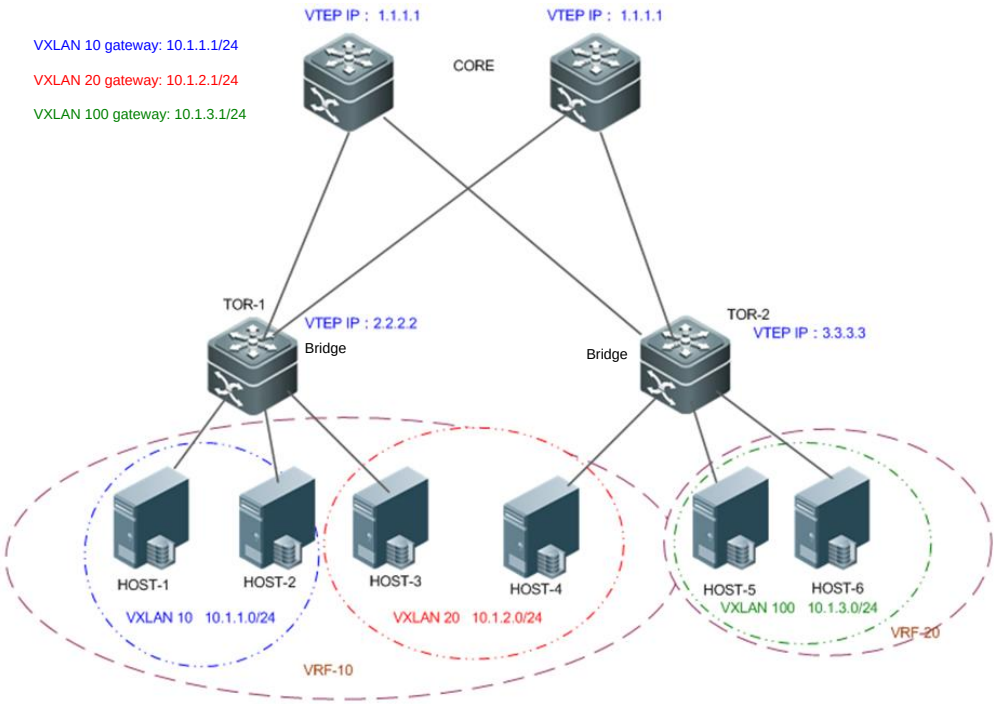

```
CORE(config)# vxlan 10
CORE(config-vxlan)# router-interface OverlayRouter 10
CORE(config-vxlan)# exit
CORE(config)# vxlan 20
CORE(config-vxlan)# router-interface OverlayRouter 20
CORE(config-vxlan)# exit
CORE(config)# vxlan 100
CORE(config-vxlan)# router-interface OverlayRouter 100
CORE(config-vxlan)# exit
CORE(config)# router bgp 64512
CORE(config-router)# neighbor 1.1.1.3 remote-as 64512
CORE(config-router)# neighbor 1.1.1.3 update-source loopback 1
CORE(config-router)# neighbor 2.2.2.2 remote-as 64512
CORE(config-router)# neighbor 2.2.2.2 update-source loopback 1
CORE(config-router)# neighbor 3.3.3.3 remote-as 64512
CORE(config-router)# neighbor 3.3.3.3 update-source loopback 1
CORE(config-router)# address-family l2vpn evpn
CORE(config-router)# neighbor 1.1.1.3 activate
CORE(config-router)# neighbor 2.2.2.2 activate
CORE(config-router)# neighbor 3.3.3.3 activate
CORE(config-router-af)# exit
```

Scenario
Figure 1-24




```
CORE(config-router)# address-family ipv4 vrf vrf-10
CORE(config-router-af)# network 10.1.1.0 mask 255.255.255.0
CORE(config-router-af)# network 10.1.2.0 mask 255.255.255.0
CORE(config-router-af)# exit
CORE(config-router)# address-family ipv4 vrf vrf-20
CORE(config-router-af)# network 10.1.3.0 mask 255.255.255.0
CORE(config-router-af)# exit
CORE(config-router)# exit
CORE(config)# evpn
CORE(config-evpn)# vni 10
CORE(config-evpn-vni)# rd auto
CORE(config-evpn-vni)# route-target both auto
CORE(config-evpn-vni)# exit
CORE(config-evpn)# vni 20
CORE(config-evpn-vni)# rd auto
CORE(config-evpn-vni)# route-target both auto
CORE(config-evpn-vni)# exit
CORE(config-evpn)# vni 100
CORE(config-evpn-vni)# rd auto
CORE(config-evpn-vni)# route-target both auto
CORE(config-evpn-vni)# exit
```

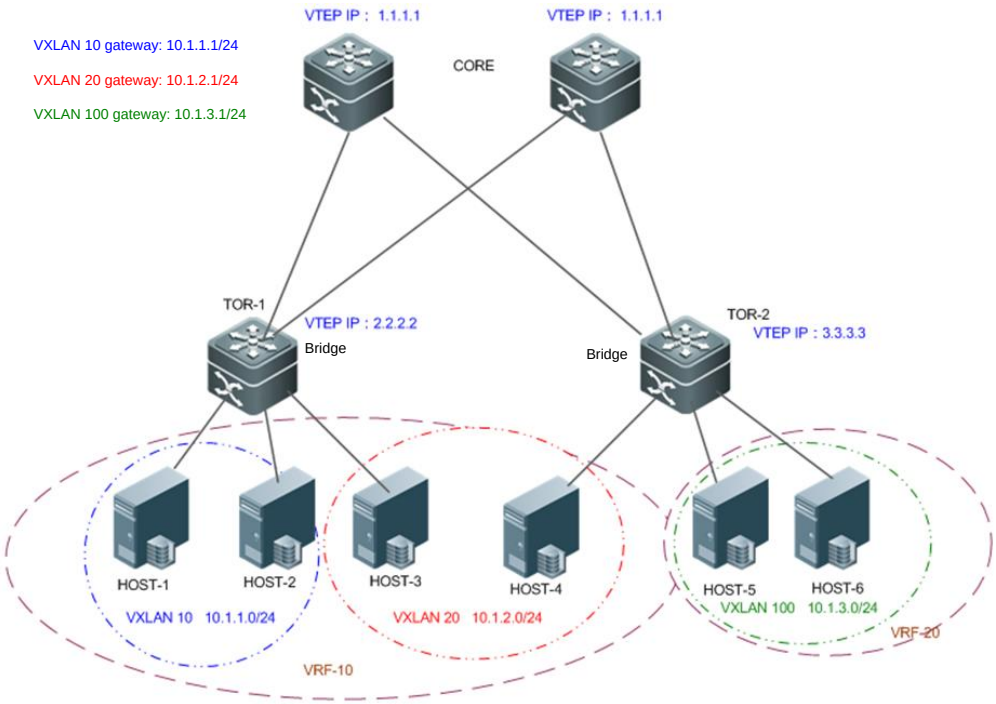
Scenario
Figure 1-24



TOR1

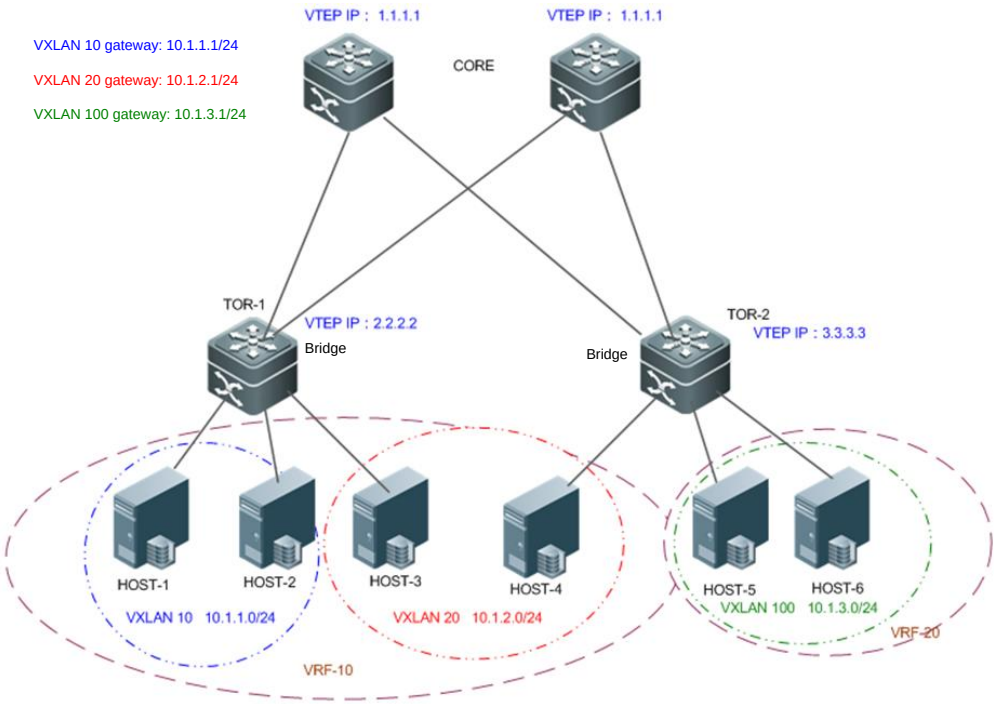
```
TOR1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
TOR1(config)# interface Loopback 1
TOR1(config-if- Loopback 1)# ip address 2.2.2.2/32
TOR1(config-if- Loopback 1)# exit
TOR1(config)# vtep
TOR1(config-vtep)# source loopback 1
TOR1(config-vtep)# arp suppress enable
TOR1(config-vtep)# exit
TOR1(config)# vxlan 10
TOR1(config-vxlan)# extend-vlan 10
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 20
TOR1(config-vxlan)# extend-vlan 20
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# router bgp 64512
TOR1 (config-router)# neighbor 1.1.1.2 remote-as 64512
TOR1 (config-router)# neighbor 1.1.1.2  update-source loopback 1
TOR1 (config-router)# neighbor 1.1.1.3 remote-as 64512
```

Scenario Figure 1-24



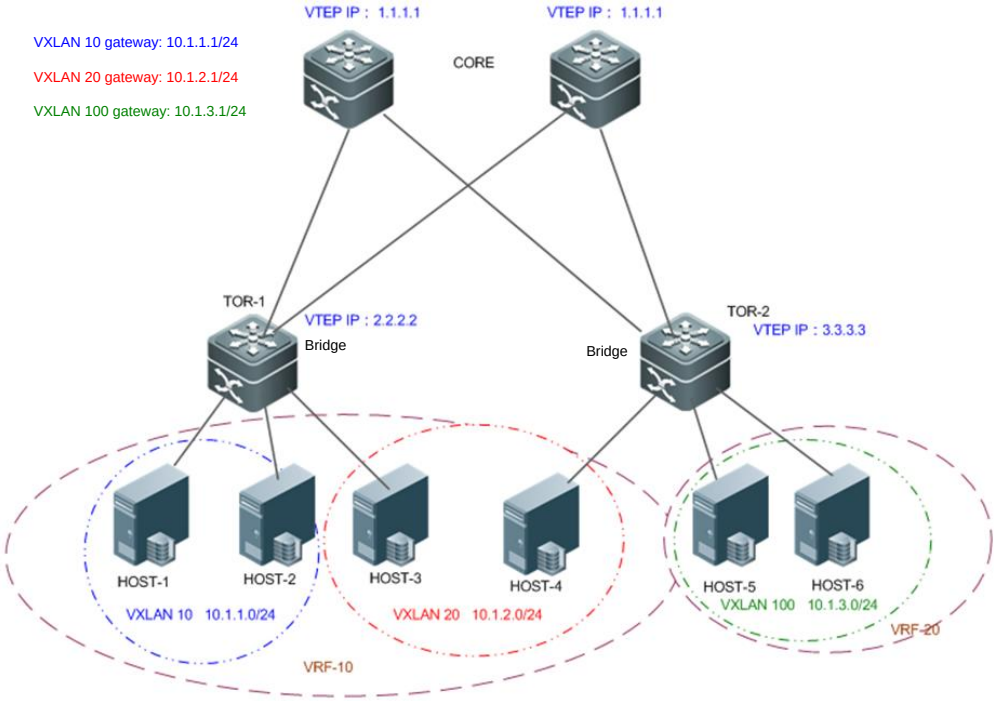
	TOR1 (config-router)# neighbor 1.1.1.3 update-source loopback 1 TOR1 (config-router)# neighbor 3.3.3.3 remote-as 64512 TOR1 (config-router)# neighbor 3.3.3.3 update-source loopback 1 TOR1(config-router)# address-family l2vpn evpn TOR1(config-router-af)# neighbor 1.1.1.2 activate TOR1(config-router-af)# neighbor 1.1.1.3 activate TOR1(config-router-af)# neighbor 3.3.3.3 activate TOR1(config-router-af)# exit TOR1(config-router)# exit TOR1(config)# evpn TOR1(config-evpn)# vni 10 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 20 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit
TOR2	TOR2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR2(config)# interface Loopback 1

Scenario Figure 1-24




```
TOR2(config-if- Loopback 1)# ip address 3.3.3.3/32
TOR2(config-if- Loopback 1)# exit
TOR2(config)# vtep
TOR2(config-vtep)# source loopback 1
TOR2(config-vtep)# arp suppress enable
TOR2(config-vtep)# exit
TOR2(config)# vxlan 100
TOR2(config-vxlan)# extend-vlan 100
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# vxlan 20
TOR2(config-vxlan)# extend-vlan 20
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# router bgp 64512
TOR2 (config-router)# neighbor 1.1.1.2 remote-as 64512
TOR2 (config-router)# neighbor 1.1.1.2  update-source loopback 1
TOR2 (config-router)# neighbor 1.1.1.3 remote-as 64512
TOR2 (config-router)# neighbor 1.1.1.3  update-source loopback 1
TOR2 (config-router)# neighbor 2.2.2.2 remote-as 64512
TOR2 (config-router)# neighbor 2.2.2.2  update-source loopback 1
```

Scenario Figure 1-24

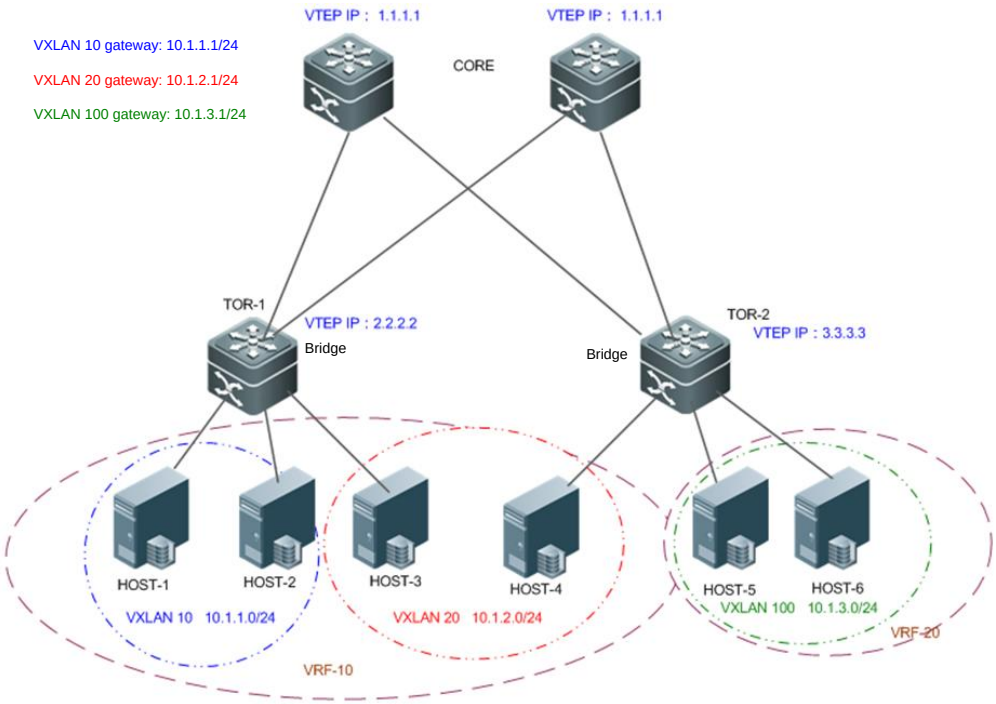



```
TOR2(config-router)# address-family l2vpn evpn
TOR2(config-router-af)# neighbor 1.1.1.2 activate
TOR2(config-router-af)# neighbor 1.1.1.3 activate
TOR2(config-router-af)# neighbor 2.2.2.2 activate
TOR2(config-router-af)# exit
TOR2(config-router)# exit
TOR2(config)# evpn
TOR2(config-evpn)# vni 20
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 100
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
```

Verification

- Verify that HOST-1, HOST-2, HOST-3, and HOST-4 can ping each other.
- Verify that HOST-5 and HOST-6 can ping each other.
- Verify that HOST-1, HOST-2, HOST-3, and HOST-4 cannot ping HOST-5 and HOST-6.
- Verify that virtual machines can be migrated between the hosts on the same VXLAN and can access the network normally after

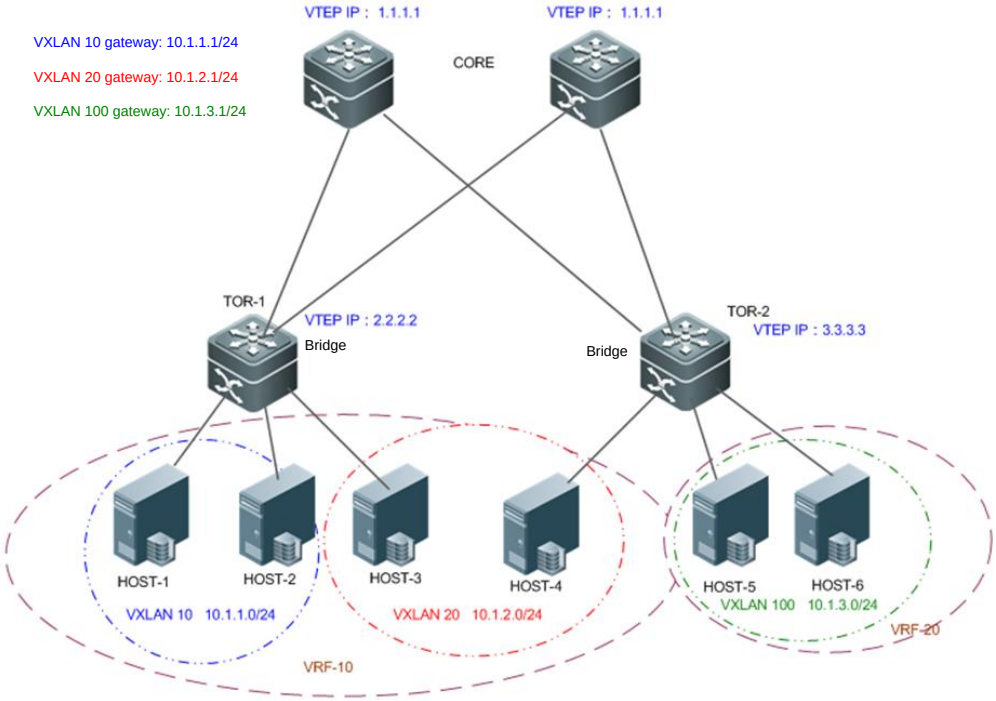
Scenario Figure 1-24



migration without modifying the configuration.

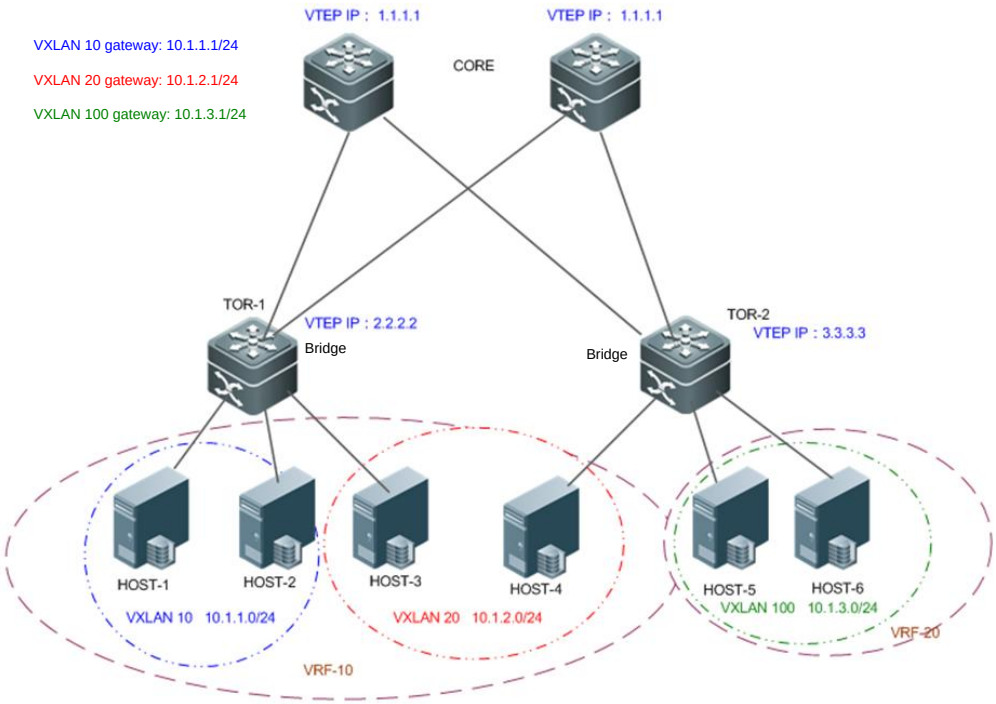
```
TOR1#sho vxlan
VXLAN Total Count: 2
VXLAN Capacity   : 8000


VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : -
   Extend VLAN         : 10
   VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6145    2.2.2.2        1.1.1.1        dynamic


VXLAN 20
   Symmetric property  : FALSE
   Router Interface    : -
   Extend VLAN         : 20
VTEP Adjacency Count: 2
```

Scenario
Figure 1-24




```
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6145    2.2.2.2        1.1.1.1        dynamic
OverlayTunnel 6146    2.2.2.2        3.3.3.3        dynamic


CORE#sho vxlan
VXLAN Total Count: 3
VXLAN Capacity   : 8000


VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 10 (anycast)
   Extend VLAN         : 10
   VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6147    1.1.1.1        2.2.2.2        dynamic
```

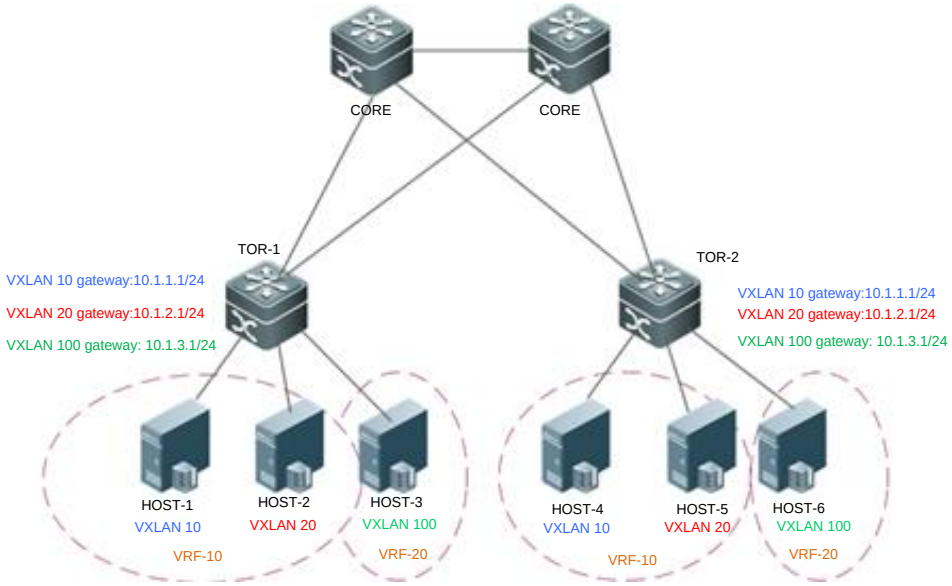
Scenario Figure 1-24




```
VXLAN 20
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 20 (anycast)
   Extend VLAN         : 20
VTEP Adjacency Count: 2
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6147    1.1.1.1        2.2.2.2        dynamic
OverlayTunnel 6148    1.1.1.1        3.3.3.3        dynamic


VXLAN 100
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 100 (anycast)
   Extend VLAN         : 100
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
OverlayTunnel 6148    1.1.1.1        3.3.3.3        dynamic
```

1.4.2.3. Configuring EVPN-based Multi-tenant Distributed Scenario (Enabling Anycast Gateway)

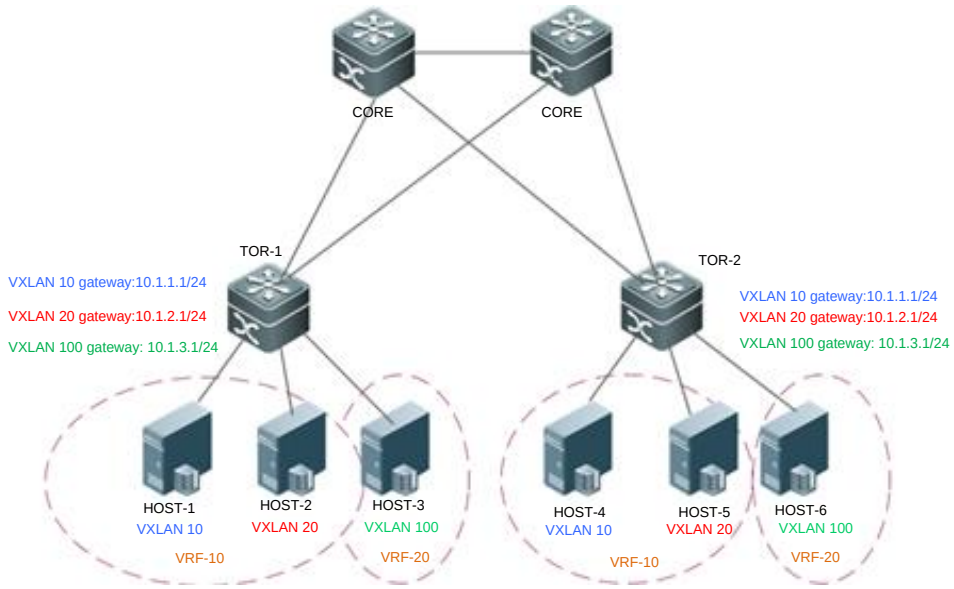
Scenario
Figure 1-25



Configuration Steps

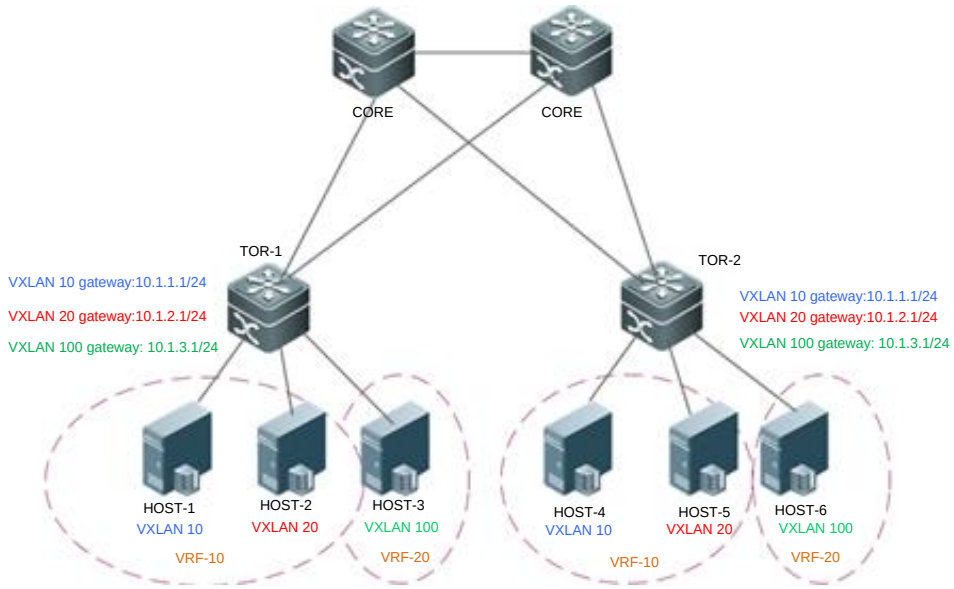
- Configure an IPv4 unicast routing protocol such as the OSPF protocol on CORE, TOR-1, and TOR-2 to ensure that unicast routes are reachable.
- Configure the BGP-EVPN routing protocol on CORE, TOR-1, and TOR-2 to establish BGP neighbor relationships between the four devices and to support the EVPN protocol family.
- Configure the EVI for BGP-EVPN on TOR-1 and TOR-2. For details, see *BGP-EVPN Configuration Guide*.
- Configure a VXLAN on the virtual server and designate the gateway address of the virtual machine.
- Associate the VTEP with loopback interface on TOR-1 and TOR-2 to establish tunnels.
- Configure the anycast gateway MAC address on TOR-1 and TOR-2 to ensure that all VXLAN anycast gateways on the network use the same MAC address.
- Create VXLAN instances on TOR-1 and TOR-2 and associate the VXLAN instances with VLANs.
- Create overlay router interfaces on TOR-1 and TOR-2 and configure the VXLAN gateway IP address for the interfaces. Configure different VRF networks for different overlay router interfaces to determine their respective tenants. Configure the anycast gateway to ensure that all VXLAN gateways on the network use the same IP address and MAC address. As the anycast gateway function is enabled, the overlay router interfaces associated with the same VXLAN on TOR-1 and TOR-2 must be configured with the same VXLAN gateway IP address.
- Associate VXLAN instances with overlay router interfaces on TOR-1 and TOR-2 to realize VXLAN routing.
- (Optional) Configure ARP suppression on TOR-1 and TOR-2 to

Scenario
Figure 1-25



	reduce the ARP packets entering the VXLAN. • (Optional) Configure ARP proxy on the overlay router interfaces of TOR-1 and TOR-2 so that the VXLAN gateway uses the gateway MAC address to respond as a proxy and VXLAN network traffic is forwarded only at L3.
HOST	Configuring the IP address and gateway according to Figure 2-25 (the detailed configuration on the server is omitted herein).
CORE	VXLAN may be not configured on the core switches. The configuration of the OSPF and BGP is omitted herein.
TOR1	TOR1# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR1(config)# interface Loopback 1 TOR1(config-if- Loopback 1)# ip address 2.2.2.2/32 TOR1(config-if- Loopback 1)# exit TOR1(config)# vtep TOR1(config-vtep)# source loopback 1 TOR1(config-vtep)# arp suppress enable TOR1(config-vtep)# exit TOR1(config)# fabric anycast-gateway-mac 0011.2233.2016 TOR1(config)# ip vrf vrf-10 TOR1(config-vrf)# rd 10:10 TOR1(config-vrf)# route-target both 1000:1000 TOR1(config-vrf)# exit

Scenario
Figure 1-25



TOR1(config)# ip vrf vrf-20

TOR1(config-vrf)# rd 20:20

TOR1(config-vrf)# route-target both 2000:2000

TOR1(config-vrf)# exit

TOR1(config)# int overlayrouter 10

TOR1(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10

TOR1(config-if-OverlayRouter 10)# ip address 10.1.1.1/24

TOR1(config-if-OverlayRouter 10)# anycast-gateway

TOR1(config-if-OverlayRouter 10)# route-in-vni //Optional. It needs to be used in combination with the arp suppress enable command.

TOR1(config-if-OverlayRouter 10)# exit

TOR1(config)# int overlayrouter 20

TOR1(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10

TOR1(config-if-OverlayRouter 20)# ip address 10.1.2.1/24

TOR1(config-if-OverlayRouter 20)# anycast-gateway

TOR1(config-if-OverlayRouter 20)# route-in-vni //Optional. It needs to be used in combination with the arp suppress enable command.

TOR1(config-if-OverlayRouter 20)# exit

TOR1(config)# int overlayrouter 100

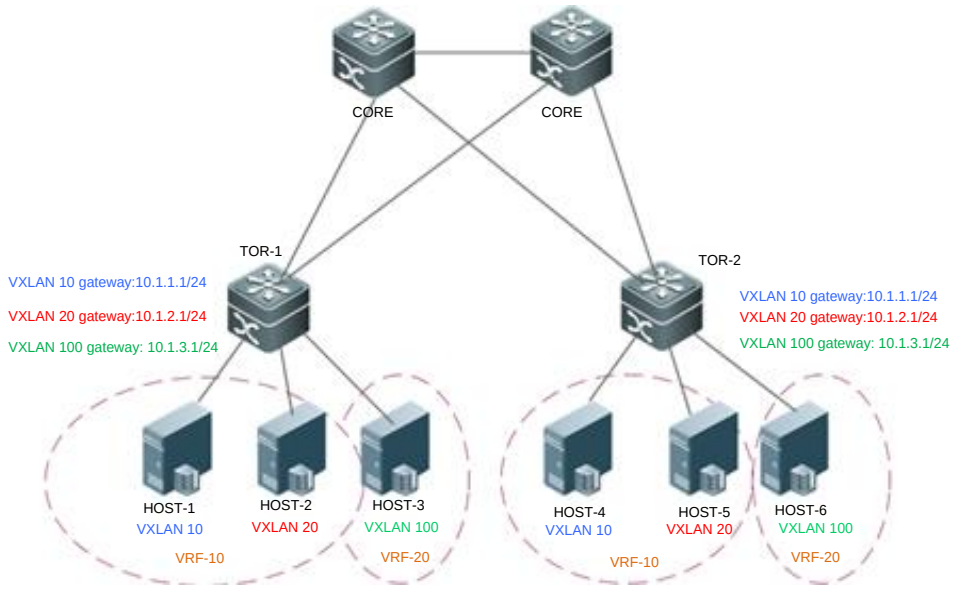
TOR1(config-if-OverlayRouter 100)# ip vrf forwarding vrf-20

TOR1(config-if-OverlayRouter 100)# ip address 10.1.3.1/24

TOR1(config-if-OverlayRouter 100)# anycast-gateway

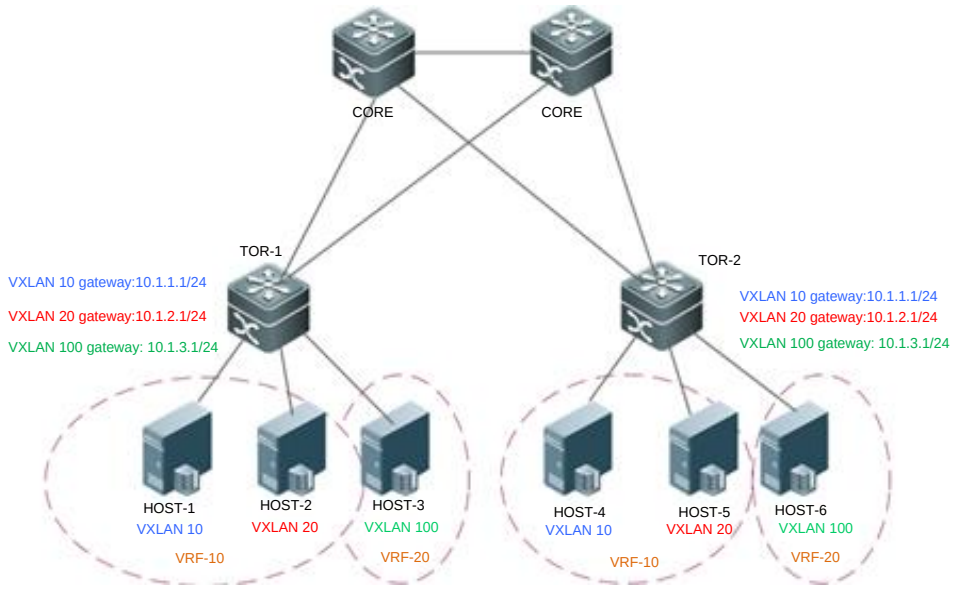
TOR1(config-if-OverlayRouter 100)# route-in-vni //Optional. It needs to be used in combination with the arp suppress enable command.

Scenario
Figure 1-25



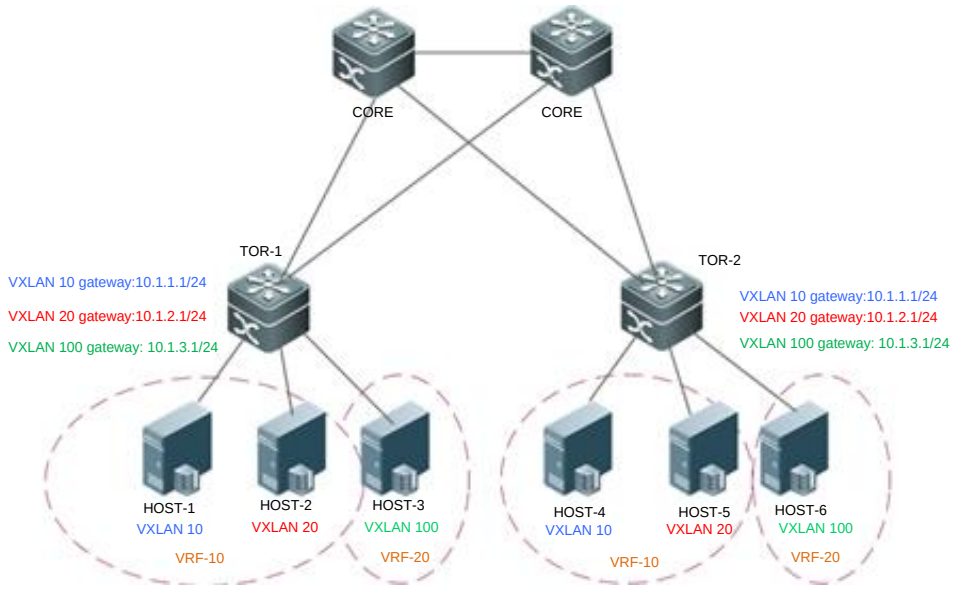

```
TOR1(config-if-OverlayRouter 100)# exit
TOR1(config)# vxlan 10
TOR1(config-vxlan)# extend-vlan 10
TOR1(config-vxlan)# router-interface OverlayRouter 10
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 20
TOR1(config-vxlan)# extend-vlan 20
TOR1(config-vxlan)# router-interface OverlayRouter 20
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 100
TOR1(config-vxlan)# extend-vlan 100
TOR1(config-vxlan)# router-interface OverlayRouter 100
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# router bgp 64512
TOR1(config-router)# neighbor 3.3.3.3 remote-as 64512
TOR1(config-router)# neighbor 3.3.3.3 update-source loopback 1
TOR1(config-router)# address-family l2vpn evpn
TOR1(config-router-af)# neighbor 3.3.3.3 activate
TOR1(config-router-af)# exit
TOR1(config-router)# exit
TOR1(config)# evpn
```

Scenario
Figure 1-25



	TOR1(config-evpn)# vni 10 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 20 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 100 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit
TOR2	TOR2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR2(config)# interface Loopback 1 TOR2(config-if- Loopback 1)# ip address 3.3.3.3/32 TOR2(config-if- Loopback 1)# exit TOR2(config)# vtep TOR2(config-vtep)# source loopback 1 TOR2(config-vtep)# arp suppress enable TOR2(config-vtep)# exit TOR1(config)# fabric anycast-gateway-mac 0011.2233.2016 TOR2(config)# ip vrf vrf-10

Scenario
Figure 1-25



TOR2(config-vrf)# rd 10:10

TOR2(config-vrf)# route-target both 1000:1000

TOR2(config-vrf)# exit

TOR2(config)# ip vrf vrf-20

TOR2(config-vrf)# rd 20:20

TOR2(config-vrf)# route-target both 2000:2000

TOR2(config-vrf)# exit

TOR2(config)# int overlayrouter 10

TOR2(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10

TOR2(config-if-OverlayRouter 10)# ip address 10.1.1.1/24

TOR2(config-if-OverlayRouter 10)# anycast-gateway

TOR2(config-if-OverlayRouter 10)# route-in-vni //Optional. It needs to be used in combination with the arp suppress enable command.

TOR2(config-if-OverlayRouter 10)# exit

TOR2(config)# int overlayrouter 20

TOR2(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10

TOR2(config-if-OverlayRouter 20)# ip address 10.1.2.1/24

TOR2(config-if-OverlayRouter 20)# anycast-gateway

TOR2(config-if-OverlayRouter 20)# route-in-vni //Optional. It needs to be used in combination with the arp suppress enable command.

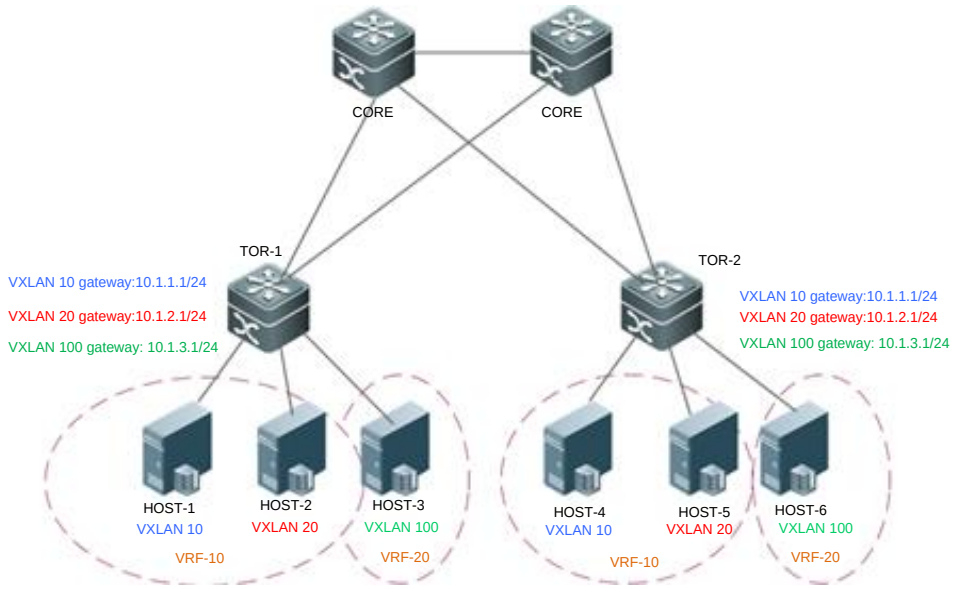
TOR2(config-if-OverlayRouter 20)# exit

TOR2(config)# int overlayrouter 100

TOR2(config-if-OverlayRouter 100)# ip vrf forwarding vrf-20

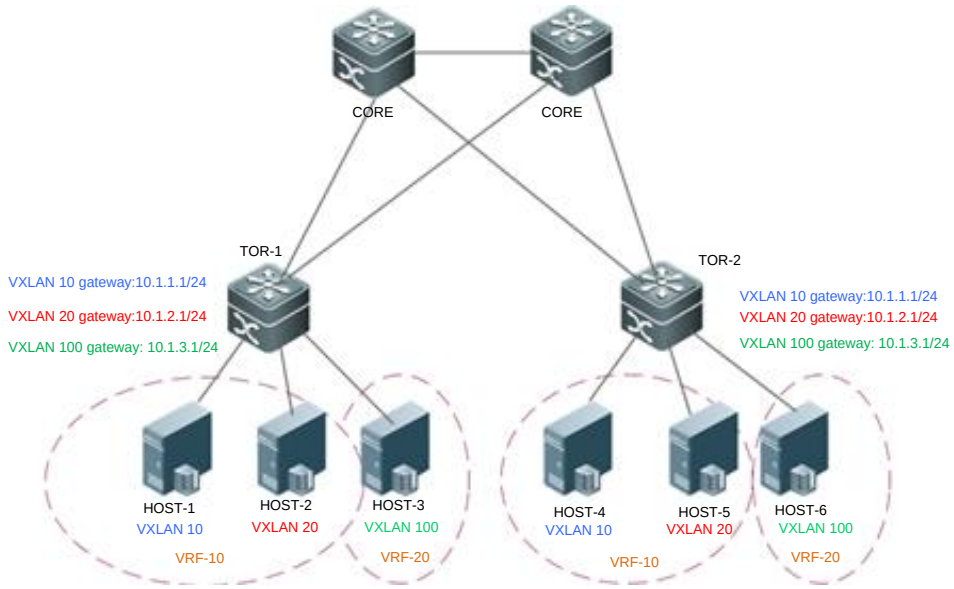
TOR2(config-if-OverlayRouter 100)# ip address 10.1.3.1/24

Scenario
Figure 1-25



	TOR2(config-if-OverlayRouter 100)# anycast-gateway
	TOR2(config-if-OverlayRouter 100)# route-in-vni //Optional. It needs to be used in combination with the arp suppress enable command.
	TOR2(config-if-OverlayRouter 100)# exit
	TOR2(config)# vxlan 10
	TOR2(config-vxlan)# extend-vlan 10
	TOR2(config-vxlan)# router-interface OverlayRouter 10
	TOR2(config-vxlan)# arp suppress enable
	TOR2(config-vxlan)# exit
	TOR2(config)# vxlan 20
	TOR2(config-vxlan)# extend-vlan 20
	TOR2(config-vxlan)# router-interface OverlayRouter 20
	TOR2(config-vxlan)# arp suppress enable
	TOR2(config-vxlan)# exit
	TOR2(config)# vxlan 100
	TOR2(config-vxlan)# extend-vlan 100
	TOR2(config-vxlan)# router-interface OverlayRouter 100
	TOR2(config-vxlan)# arp suppress enable
	TOR2(config-vxlan)# exit
	TOR2(config)# router bgp 64512
	TOR2(config-router)# neighbor 2.2.2.2 remote-as 64512
	TOR2(config-router)# neighbor 2.2.2.2 update-source loopback 1
	TOR2(config-router)# address-family l2vpn evpn
	TOR2(config-router-af)# neighbor 2.2.2.2 activate

Scenario
Figure 1-25



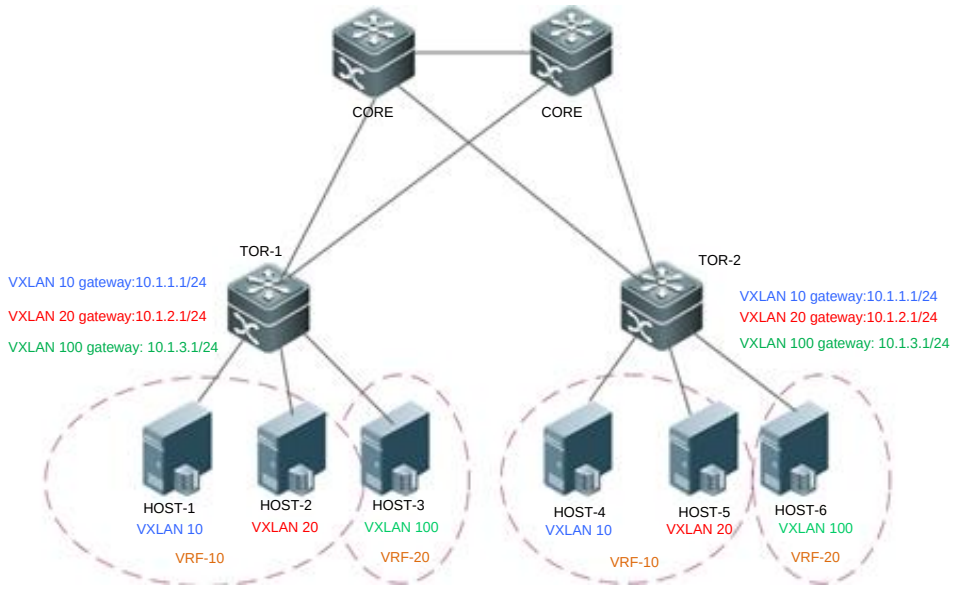

```
TOR2(config-router-af)# exit
TOR2(config-router)# exit
TOR2(config)# evpn
TOR2(config-evpn)# vni 10
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 20
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 100
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
```

Verification

- Verify that HOST-1, HOST-2, HOST-4, and HOST-5 can ping each other.
- Verify that HOST-3 and HOST-6 can ping each other.
- Verify that HOST-1, HOST-2, HOST-4, and HOST-5 cannot ping HOST-3 and HOST-6.
- Verify that the virtual machines can be migrated between the hosts on the same VXLAN and can access the network normally after migration without modifying the configuration.

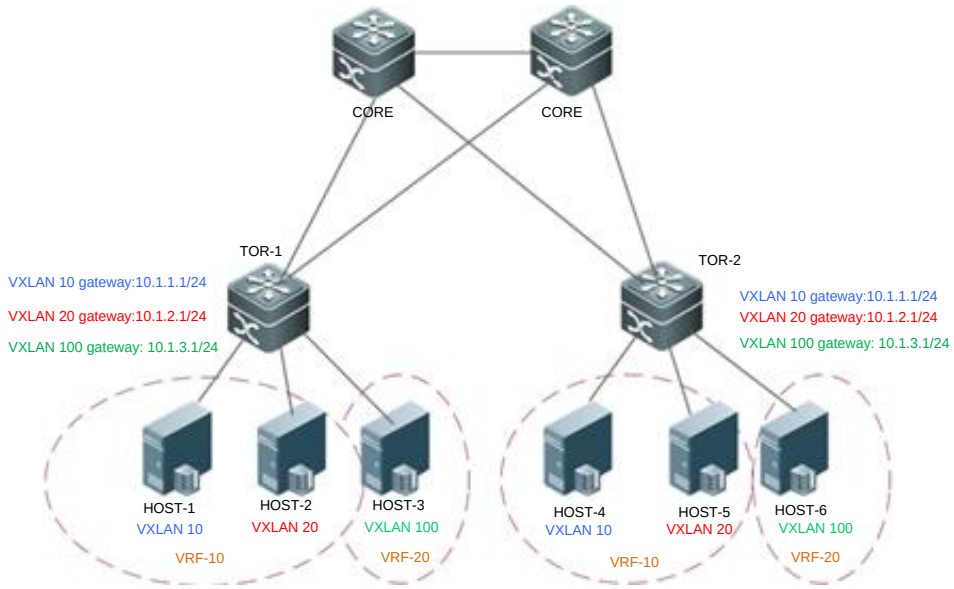
```
TOR1#sho vxlan
```

Scenario
Figure 1-25




```
VXLAN Total Count: 3
VXLAN Capacity   : 8000


VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 10 (anycast)
   Extend VLAN         : 10
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- -------------- -------------- -------
   OverlayTunnel 6145   2.2.2.2       3.3.3.3        dynamic


VXLAN 20
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 20 (anycast)
   Extend VLAN         : 20
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- -------------- -------------- -------
   OverlayTunnel 6145   2.2.2.2       3.3.3.3        dynamic
```

Scenario
Figure 1-25




```
VXLAN 100
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 100 (anycast)
   Extend VLAN         : 100
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6145   2.2.2.2        3.3.3.3         dynamic
```

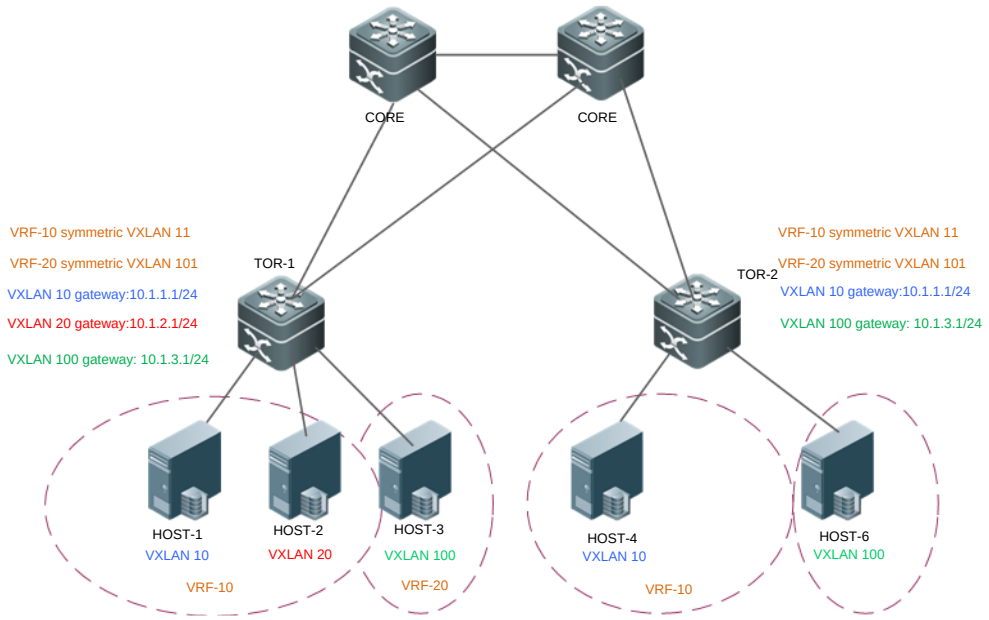
Common Errors

Note:

- When symmetric deployment is disabled, all TOR switches of the same VRF network must have all the VXLAN gateways of the VRF network configured on them. For example, VRF-10 includes VXLAN 10 and VXLAN 20, and therefore all gateways of VXLAN 10 and VXLAN 20 must be configured on TOR-1 and TOR-2. Otherwise, VXLAN 10 and VXLAN 20 cannot communicate with each other If you expect to deploy only required gateways instead of deploying all gateways on all TOR switches, apply symmetric deployment. For details, see section “Configuring EVPN-based Multi-tenant Distributed Scenario (Symmetric Deployment)."
- Make sure that the global anycast MAC address is not the same as that of any device on the VXLAN.

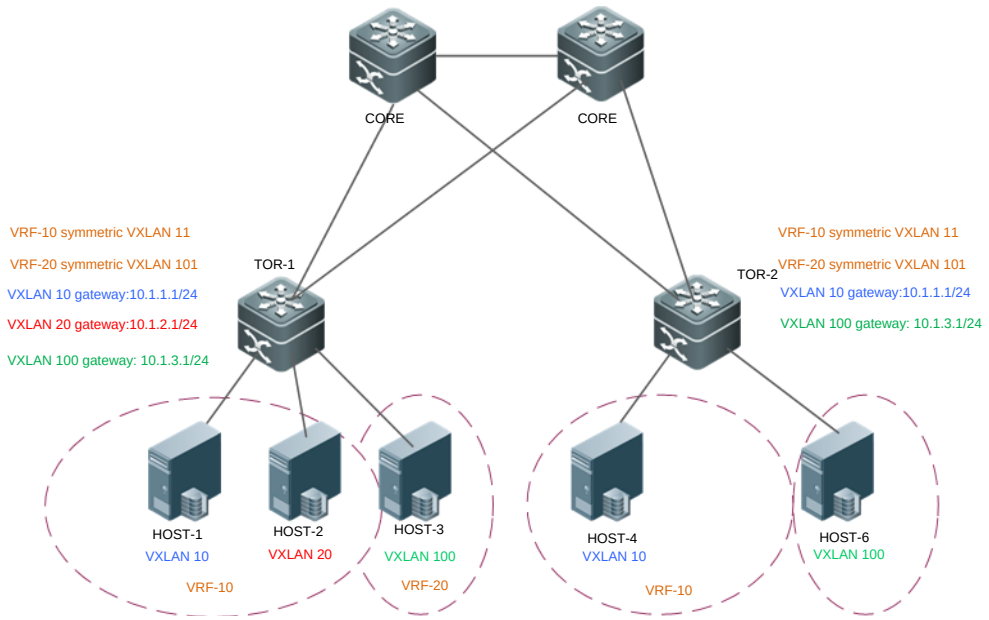
1.4.2.4. Configuring EVPN-based Multi-tenant Distributed Scenario (Symmetric Deployment)

Scenario
Figure 1-26



Configuration Steps	• Configure an IPv4 unicast routing protocol such as the OSPF protocol on CORE, TOR-1, and TOR-2 to ensure that unicast routes are reachable. • Configure the BGP-EVPN routing protocol on CORE, TOR-1, and TOR-2 to establish BGP neighbor relationships between the four devices and to support the EVPN protocol family. • Configure the EVI for BGP-EVPN on TOR-1 and TOR-2. For details, see *BGP-EVPN Configuration Guide*. • Configure a VXLAN on the virtual server and designate the gateway address of the virtual machine. • Associate the VTEP with loopback interface on TOR-1 and TOR-2 to establish tunnels. • Configure the anycast gateway MAC address on TOR-1 and TOR-2 to ensure that all VXLAN anycast gateways on the network use the same MAC address. • Create VXLAN 10, VXLAN20, and VXLAN 100 on TOR-1 and associate them with VLANs. • Create VXLAN 10 and VXLAN 100 on TOR-2 and associate them with VLANs. • Create overlay router interfaces for VXLAN 10, VXLAN 20, and VXLAN 100 on TOR-1 and TOR-2 (TOR-2 do not have VXLAN 20), and configure the VXLAN gateway IP address for them. Configure different VRF networks for different overlay router interfaces to determine their respective tenants. Configure the anycast gateway to ensure that all VXLAN gateways on the network use the same IP address and MAC address. As the anycast gateway function is enabled, the overlay router interfaces

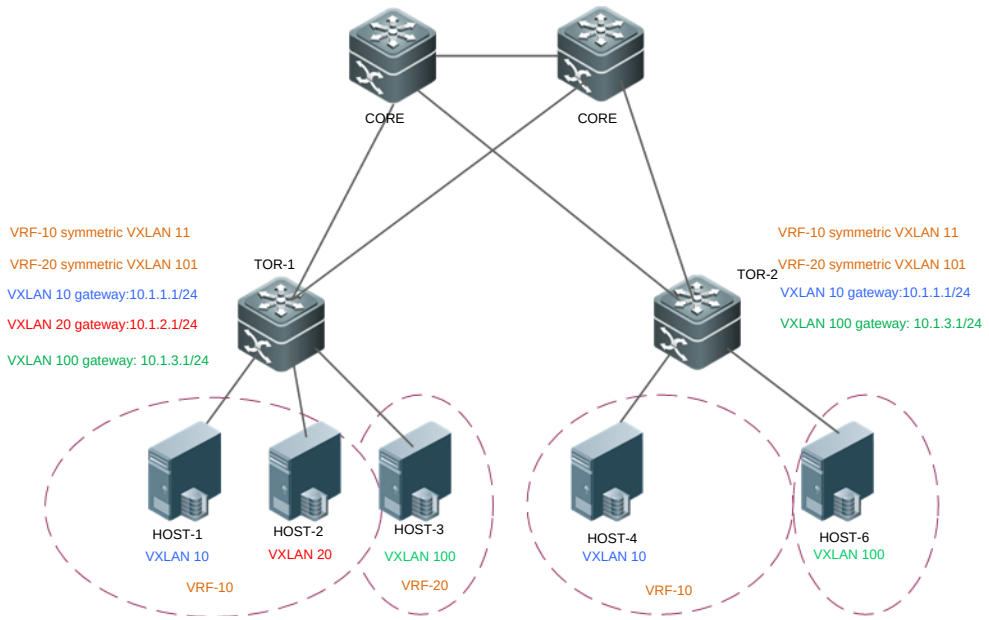
Scenario
Figure 1-26



associated with the same VXLAN on TOR-1 and TOR-2 must be configured with the same VXLAN gateway IP address.

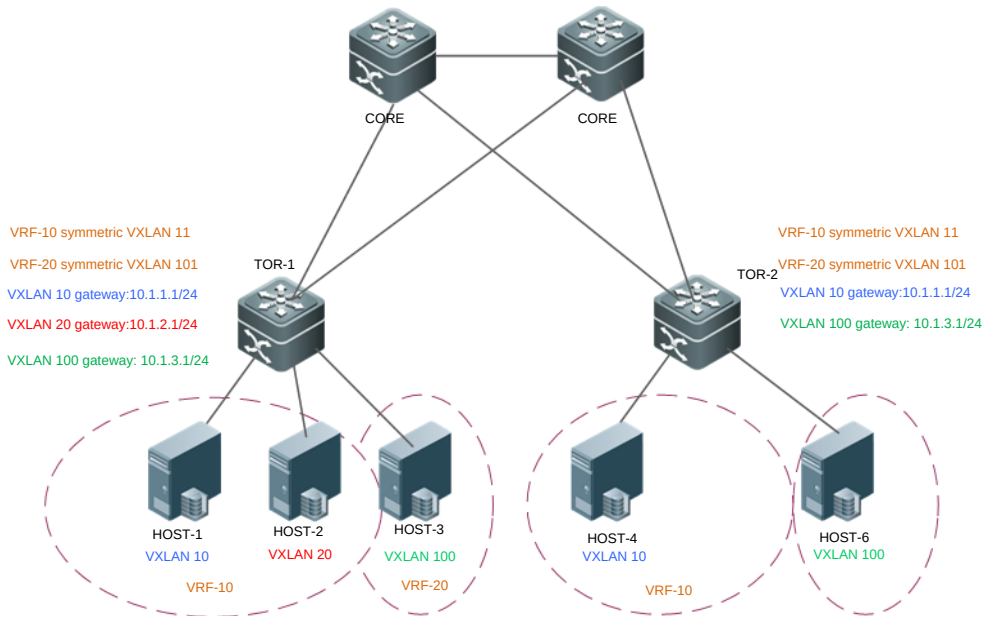
- Create VXLAN 11 and VXLAN 101 on TOR-1 and TOR-2 and configure them as symmetric VXLANs to serve as L3 routing VXLAN of the corresponding VRF networks. L3 routes between all VXLANs of the same VRF network are advertised via the symmetric VXLANs. In addition, the symmetric VXLANs are also used for L3 routing and forwarding.
- Create overlay router interfaces for VXLAN 11 and VXLAN 101 on TOR-1 and TOR-2. Configure different VRF networks for the overlay router interfaces. VXLAN 11 and VXLAN 101 serve as the symmetric VXLANs of the corresponding VRF networks.
- Associate VXLAN instances with overlay router interfaces on TOR-1 and TOR-2 to realize VXLAN routing.
- (Optional) Configure ARP suppression on TOR-1 and TOR-2 to reduce the ARP packets entering the VXLAN.
- (Optional) Configure ARP proxy on the overlay router interfaces belonging to the L2-VNIs (VXLANs 10, 20, and 100) on TOR-1 and TOR-2 so that the traffic of hosts in the same VXLAN is forwarded at L3. This function needs to be enabled together with ARP suppression. The configuration command is **route-in-vni**.
- (Optional) Configure no synchronization of EVPN entries in ARP proxy deployment scenario: On the L2-VNI VXLAN instances (VXLANs 10, 20, and 100) on TOR-1 and TOR-2, configure not to advertise or receive MAC-only EVPN type-2 routes.
  - Configure not to advertise EVPN MAC-only type-2 routes to reduce EVPN route synchronization between devices. The configuration command is **evpn mac advertise disable**.
  - Configure not to deliver MAC entries synchronized by EVPNs to

Scenario
Figure 1-26



	reduce the occupancy of hardware entry resources. The configuration command is **evpn mac inactive**. The two commands above can be configured independently and they do not affect each other. • (Optional) Configure no synchronization of EVPN entries in scenarios where ARP proxy is not deployed: On the L2-VNI VXLAN instances (VXLANs 10, 20, and 100) on TOR-1 and TOR-2, configure not to advertise EVPN MAC-only type-2 routes and configure to extract MAC addresses from MAC-IP type-2 routes. ○ Configure not to advertise EVPN MAC-only type-2 routes to reduce EVPN route synchronization between devices. The configuration command is **evpn mac advertise disable**. ○ Configure to extract MAC addresses from MAC-IP type-2 routes so that the device can learn MAC entries even if neighbors do not adverse MAC-only type-2 routes. The configuration command is **evpn arp mac-learning enable**. • The two commands above must be used in combination. Otherwise, the device cannot learn MAC entries from neighbors and L2 forwarding cannot implemented in VXLANs.
HOST	Configuring the IP address and gateway according to Figure 2-26 (the detailed configuration on the server is omitted herein).
CORE	VXLAN may be not configured on the core switches. The configuration of the OSPF and BGP is omitted herein.
TOR1	TOR1# configure terminal Enter configuration commands, one per line. End with CNTL/Z.

Scenario
Figure 1-26



TOR1(config)# interface Loopback 1

TOR1(config-if- Loopback 1)# ip address 2.2.2.2/32

TOR1(config-if- Loopback 1)# exit

TOR1(config)# vtep

TOR1(config-vtep)# source loopback 1

TOR1(config-vtep)# arp suppress enable

TOR1(config-vtep)# exit

TOR1(config)# fabric anycast-gateway-mac 0011.2233.2016

TOR1(config)# ip vrf vrf-10

TOR1(config-vrf)# rd 10:10

TOR1(config-vrf)# route-target both 1000:1000

TOR1(config-vrf)# exit

TOR1(config)# ip vrf vrf-20

TOR1(config-vrf)# rd 20:20

TOR1(config-vrf)# route-target both 2000:2000

TOR1(config-vrf)# exit

TOR1(config)# int overlayrouter 10

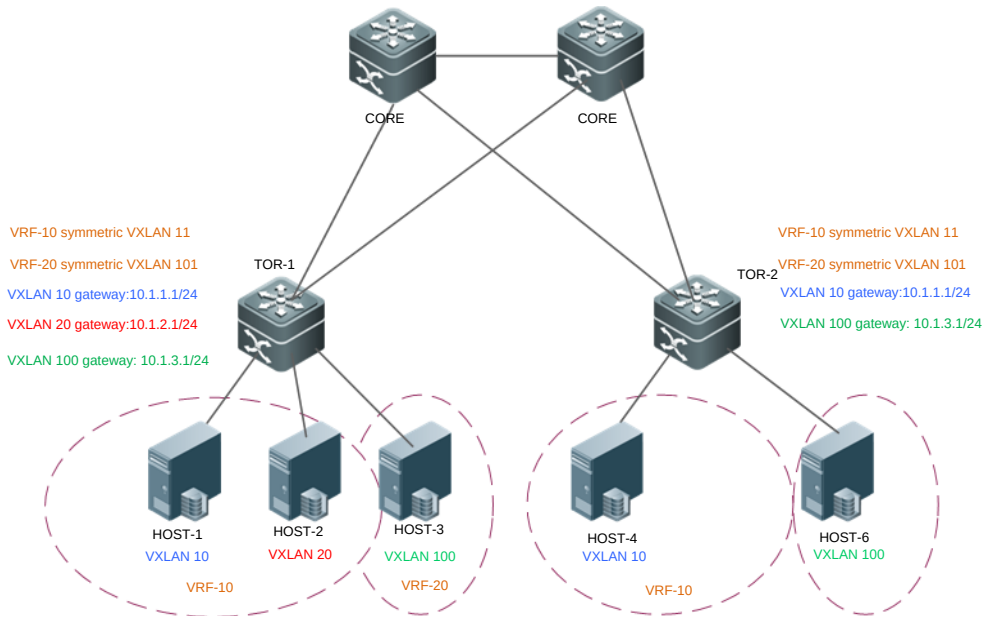
TOR1(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10

TOR1(config-if-OverlayRouter 10)# ip address 10.1.1.1/24

TOR1(config-if-OverlayRouter 10)# anycast-gateway

TOR1(config-if-OverlayRouter 10)# route-in-vni //Optional. It is used to enable ARP proxy and needs to be used in combination with the arp suppress enable command.

Scenario
Figure 1-26



TOR1(config-if-OverlayRouter 10)# exit

TOR1(config)# int overlayrouter 20

TOR1(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10

TOR1(config-if-OverlayRouter 20)# ip address 10.1.2.1/24

TOR1(config-if-OverlayRouter 20)# anycast-gateway

TOR1(config-if-OverlayRouter 20)# route-in-vni //Optional. It is used to enable ARP proxy and needs to be used in combination with the arp suppress enable command.

TOR1(config-if-OverlayRouter 20)# exit

TOR1(config)# int overlayrouter 11

TOR1(config-if-OverlayRouter 11)# ip vrf forwarding vrf-10

TOR1(config-if-OverlayRouter 11)# exit

TOR1(config)# int overlayrouter 100

TOR1(config-if-OverlayRouter 100)# ip vrf forwarding vrf-20

TOR1(config-if-OverlayRouter 100)# ip address 10.1.3.1/24

TOR1(config-if-OverlayRouter 100)# anycast-gateway

TOR1(config-if-OverlayRouter 100)# route-in-vni

TOR1(config-if-OverlayRouter 100)# exit

TOR1(config)# int overlayrouter 101

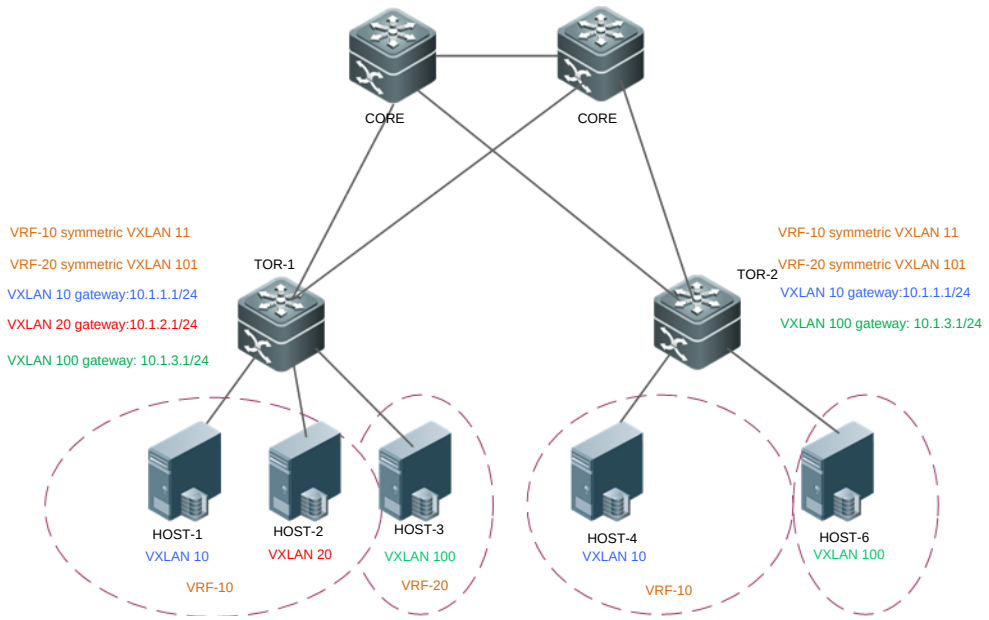
TOR1(config-if-OverlayRouter 101)# ip vrf forwarding vrf-20

TOR1(config-if-OverlayRouter 101)# exit

TOR1(config)# vxlan 10

TOR1(config-vxlan)# extend-vlan 10

Scenario
Figure 1-26



TOR1(config-vxlan)# router-interface OverlayRouter 10

TOR1(config-vxlan)# arp suppress enable

TOR1(config-vxlan)#evpn mac advertise disable

TOR1(config-vxlan)#evpn mac inactive

TOR1(config-vxlan)#evpn arp mac-learning enable

TOR1(config-vxlan)# exit

TOR1(config)# vxlan 20

TOR1(config-vxlan)# extend-vlan 20

TOR1(config-vxlan)# router-interface OverlayRouter 20

TOR1(config-vxlan)# arp suppress enable

TOR1(config-vxlan)#evpn mac advertise disable //Optional.

TOR1(config-vxlan)#evpn mac inactive //Optional. In ARP proxy deployment scenarios, it is used to reduce the synchronization of EVPN entries.

TOR1(config-vxlan)#evpn arp mac-learning enable //Optional. In scenarios in which ARP proxy is not deployed, it is used to reduce the synchronization of EVPN entries.

TOR1(config-vxlan)# exit

TOR1(config)# vxlan 11

TOR1(config-vxlan)# symmetric

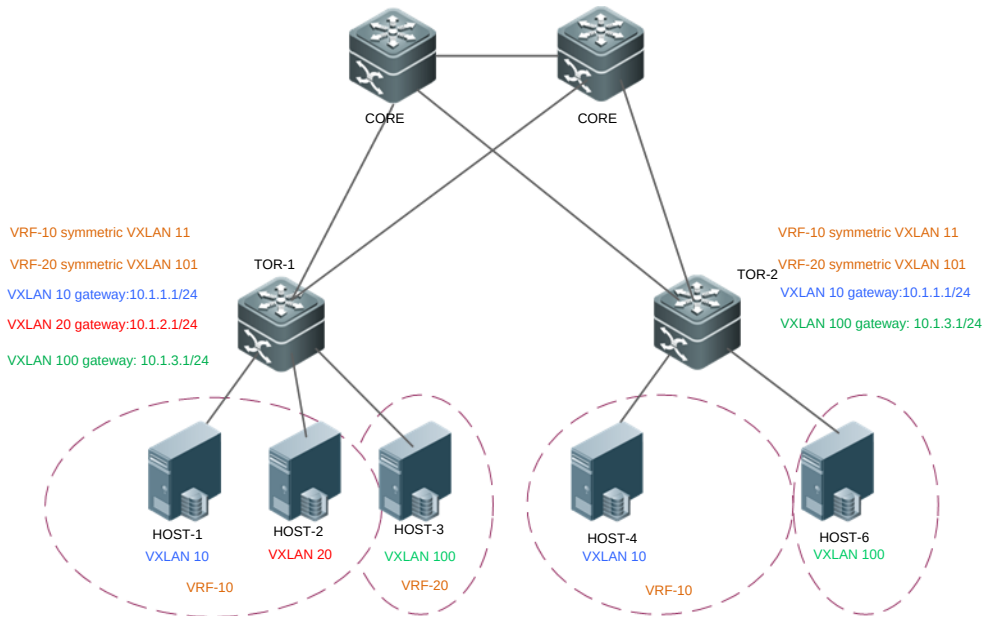
TOR1(config-vxlan)# router-interface OverlayRouter 11

TOR1(config-vxlan)# exit

TOR1(config)# vxlan 100

TOR1(config-vxlan)# extend-vlan 100

Scenario
Figure 1-26



TOR1(config-vxlan)# router-interface OverlayRouter 100

TOR1(config-vxlan)# arp suppress enable

TOR1(config-vxlan)#evpn mac advertise disable //Optional.

TOR1(config-vxlan)#evpn mac inactive //Optional. In ARP proxy deployment scenarios, it is used to reduce the synchronization of EVPN entries.

TOR1(config-vxlan)#evpn arp mac-learning enable //Optional. In scenarios in which ARP proxy is not deployed, it is used to reduce the synchronization of EVPN entries.

TOR1(config-vxlan)# exit

TOR1(config)# vxlan 101

TOR1(config-vxlan)# symmetric

TOR1(config-vxlan)# router-interface OverlayRouter 101

TOR1(config-vxlan)# exit

TOR1(config)# router bgp 64512

TOR1(config-router)# neighbor 3.3.3.3 remote-as 64512

TOR1(config-router)# neighbor 3.3.3.3 update-source loopback 1

TOR1(config-router)# address-family l2vpn evpn

TOR1(config-router-af)# neighbor 3.3.3.3 activate

TOR1(config-router-af)# exit

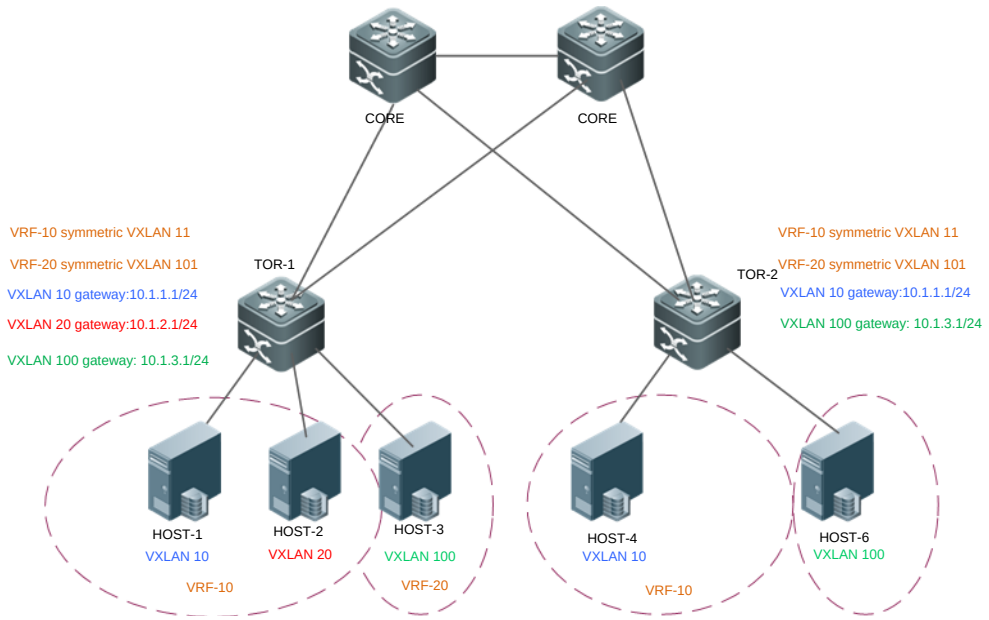
TOR1(config-router)# exit

TOR1(config)# evpn

TOR1(config-evpn)# vni 10

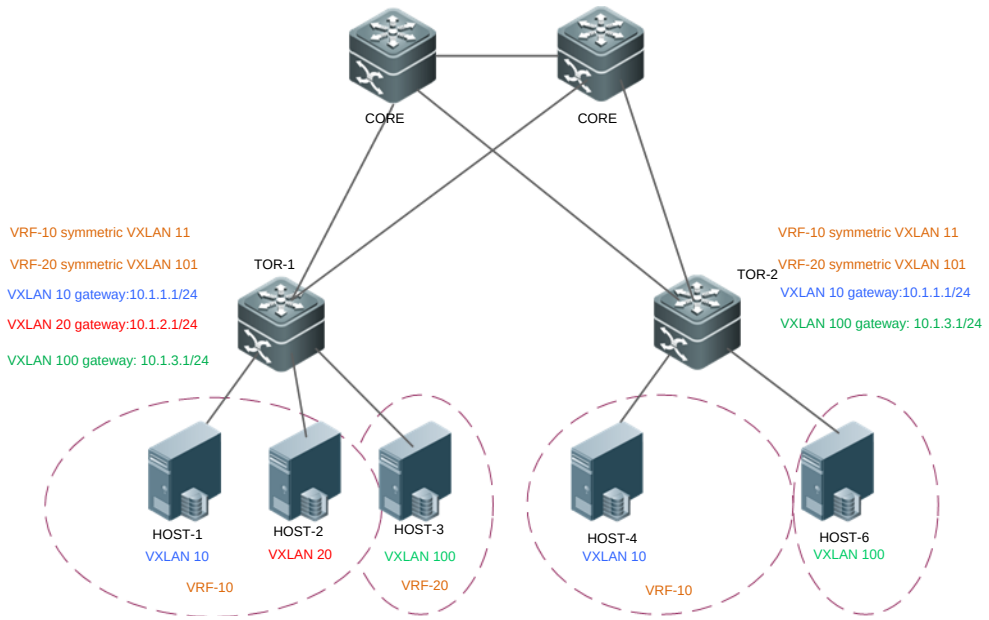
TOR1(config-evpn-vni)# rd auto

Scenario
Figure 1-26



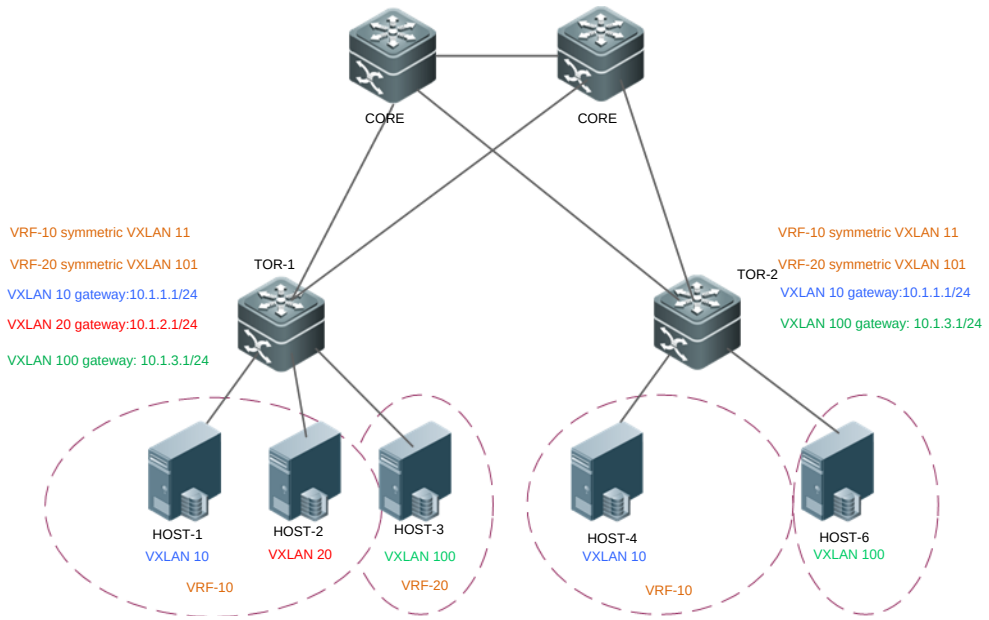
	TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 11 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 100 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 101 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit
TOR2	TOR2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR2(config)# interface Loopback 1 TOR2(config-if- Loopback 1)# ip address 3.3.3.3/32 TOR2(config-if- Loopback 1)# exit TOR2(config)# vtep TOR2(config-vtep)# source loopback 1 TOR2(config-vtep)# arp suppress enable

Scenario
Figure 1-26



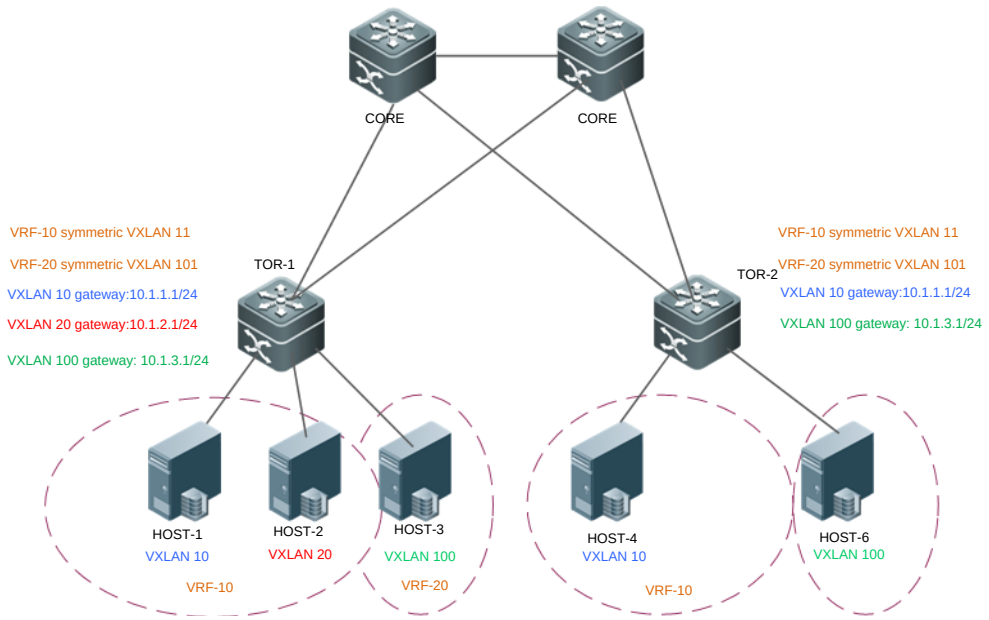
TOR2(config-vtep)# exit
TOR2(config)# fabric anycast-gateway-mac 0011.2233.2016
TOR2(config)# ip vrf vrf-10
TOR2(config-vrf)# rd 10:10
TOR2(config-vrf)# route-target both 1000:1000
TOR2(config-vrf)# exit
TOR2config)# ip vrf vrf-20
TOR2(config-vrf)# rd 20:20
TOR2(config-vrf)# route-target both 2000:2000
TOR2(config-vrf)# exit
TOR2(config)# int overlayrouter 10
TOR2(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10
TOR2(config-if-OverlayRouter 10)# ip address 10.1.1.1/24
TOR2(config-if-OverlayRouter 10)# anycast-gateway
TOR2(config-if-OverlayRouter 10)# route-in-vni //Optional. It is used to enable ARP proxy and needs to be used in combination with the arp suppress enable command.
TOR2(config-if-OverlayRouter 10)# exit
TOR2(config)# int overlayrouter 11
TOR2(config-if-OverlayRouter 11)# ip vrf forwarding vrf-10
TOR2(config-if-OverlayRouter 11)# exit
TOR2(config)# int overlayrouter 100
TOR2(config-if-OverlayRouter 100)# ip vrf forwarding vrf-20

Scenario
Figure 1-26



	TOR2(config-if-OverlayRouter 100)# ip address 10.1.3.1/24
	TOR2(config-if-OverlayRouter 100)# anycast-gateway
	TOR2(config-if-OverlayRouter 100)# route-in-vni
	TOR2(config-if-OverlayRouter 100)# exit
	TOR2(config)# int overlayrouter 101
	TOR2(config-if-OverlayRouter 101)# ip vrf forwarding vrf-20
	TOR2(config-if-OverlayRouter 101)# exit
	TOR2(config)# vxlan 10
	TOR2(config-vxlan)# extend-vlan 10
	TOR2(config-vxlan)# router-interface OverlayRouter 10
	TOR2(config-vxlan)# arp suppress enable
	TOR2(config-vxlan)#evpn mac advertise disable //Optional.
	TOR2(config-vxlan)#evpn mac inactive //Optional. In ARP proxy deployment scenarios, it is used to reduce the synchronization of EVPN entries.
	TOR2(config-vxlan)#evpn arp mac-learning enable //Optional. In scenarios in which ARP proxy is not deployed, it is used to reduce the synchronization of EVPN entries.
	TOR2(config-vxlan)# exit
	TOR2(config)# vxlan 11
	TOR2(config-vxlan)# symmetric
	TOR2(config-vxlan)# router-interface OverlayRouter 11
	TOR2(config-vxlan)# exit
	TOR2(config)# vxlan 100

Scenario
Figure 1-26



TOR2(config-vxlan)# extend-vlan 100

TOR2(config-vxlan)# router-interface OverlayRouter 100

TOR2(config-vxlan)# arp suppress enable

TOR2(config-vxlan)#evpn mac advertise disable //Optional.

TOR2(config-vxlan)#evpn mac inactive //Optional. In ARP proxy deployment scenarios, it is used to reduce the synchronization of EVPN entries.

TOR2(config-vxlan)#evpn arp mac-learning enable //Optional. In scenarios in which ARP proxy is not deployed, it is used to reduce the synchronization of EVPN entries.

TOR2(config-vxlan)# exit

TOR2(config)# int overlayrouter 101

TOR2(config-vxlan)# symmetric

TOR2(config-vxlan)# router-interface OverlayRouter 101

TOR2(config-vxlan)# exit

TOR2(config)# router bgp 64512

TOR2(config-router)# neighbor 2.2.2.2 remote-as 64512

TOR2(config-router)# neighbor 2.2.2.2 update-source loopback 1

TOR2(config-router)# address-family l2vpn evpn

TOR2(config-router-af)# neighbor 2.2.2.2 activate

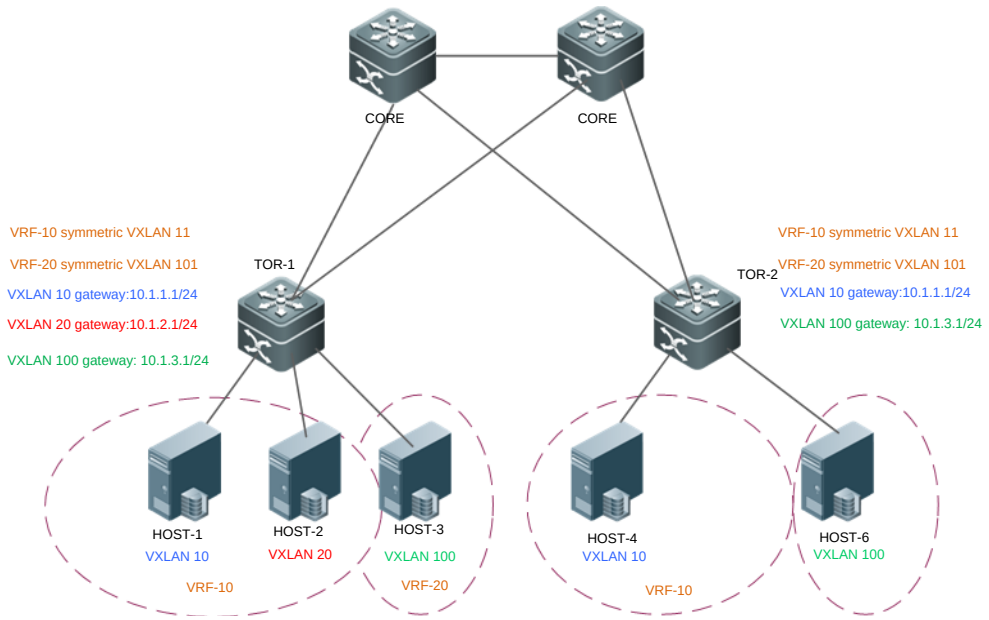
TOR2(config-router-af)# exit

TOR2(config-router)# exit

TOR2(config)# evpn

TOR2(config-evpn)# vni 10

Scenario
Figure 1-26

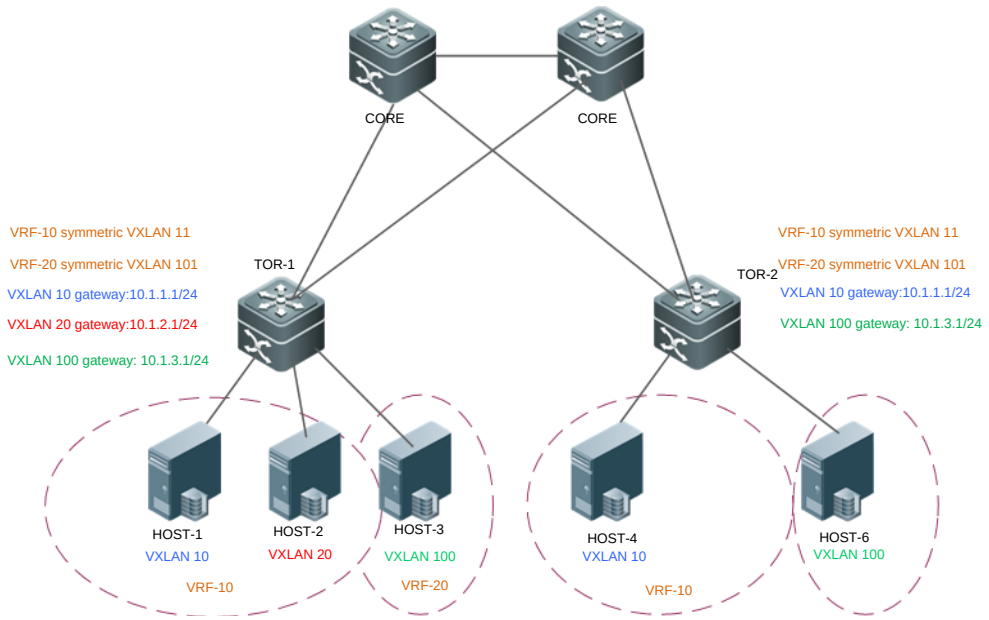



```
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 11
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 100
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 101
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
```

Verification

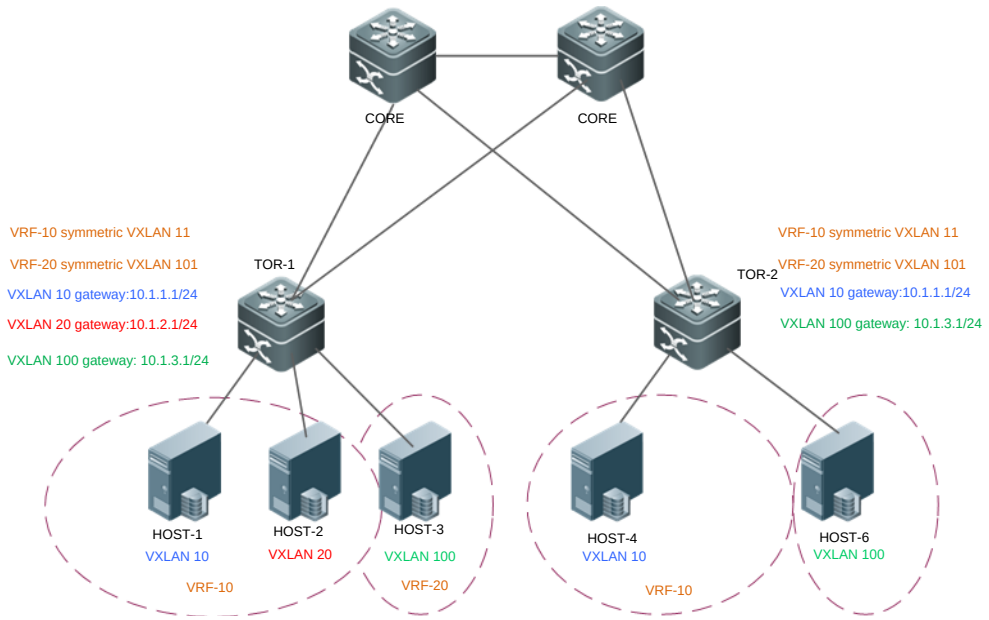
- Verify that HOST-1, HOST-2, and HOST-4 can ping each other.
- Verify that HOST-3 and HOST-6 can ping each other.
- Verify that HOST-1, HOST-2, and HOST-4 cannot ping HOST-3 and HOST-6.
- Verify that the virtual machines can be migrated between the hosts on the same VXLAN and can access the network normally after migration without modifying the configuration.

Scenario
Figure 1-26



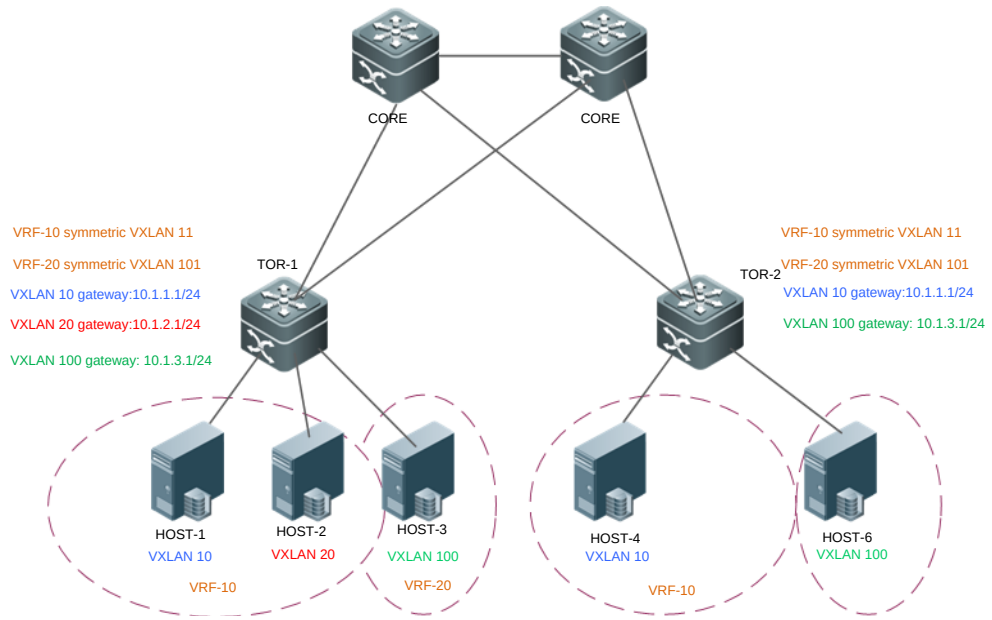

```
TOR1#sho vxlan
VXLAN Total Count: 5
VXLAN Capacity   : 8000


VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : OverlayRouter 10 (anycast)
   Extend VLAN         : 10
   VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   ------------------- -------------- -------------- -------
   OverlayTunnel 6145    2.2.2.2        3.3.3.3        dynamic


VXLAN 11
   Symmetric property  : TRUE
   Router Interface    : OverlayRouter 11 (anycast)
   Extend VLAN         : -
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   ------------------- -------------- -------------- -------
```

Scenario
Figure 1-26




```
    OverlayTunnel 6145    2.2.2.2        3.3.3.3        dynamic


VXLAN 20
    Symmetric property  : FALSE
    Router Interface    : OverlayRouter 20 (anycast)
    Extend VLAN         : 20
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
    ------------------- -------------- -------------- -------
    OverlayTunnel 6145    2.2.2.2        3.3.3.3        dynamic


VXLAN 100
    Symmetric property  : FALSE
    Router Interface    : OverlayRouter 100 (anycast)
    Extend VLAN         : 100
    VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
    ------------------- -------------- -------------- -------
    OverlayTunnel 6145    2.2.2.2        3.3.3.3        dynamic
```

Scenario
Figure 1-26




```
VXLAN 101
   Symmetric property  : TRUE
   Router Interface    : OverlayRouter 101 (anycast)
   Extend VLAN         : -
   VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 6145   2.2.2.2        3.3.3.3         dynamic
```

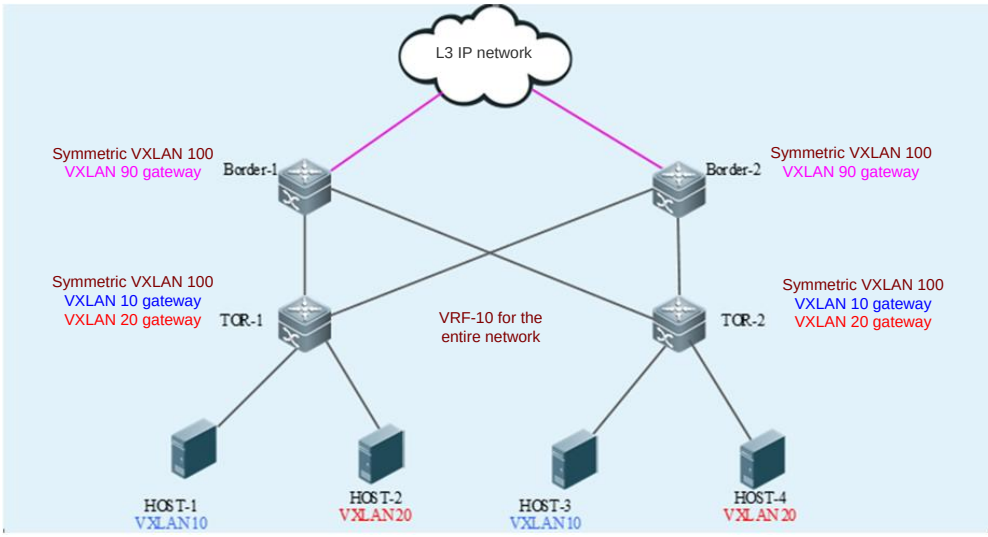
Common Errors

Note:

- Make sure that the global anycast MAC address is not the same as that of any device on the VXLAN.

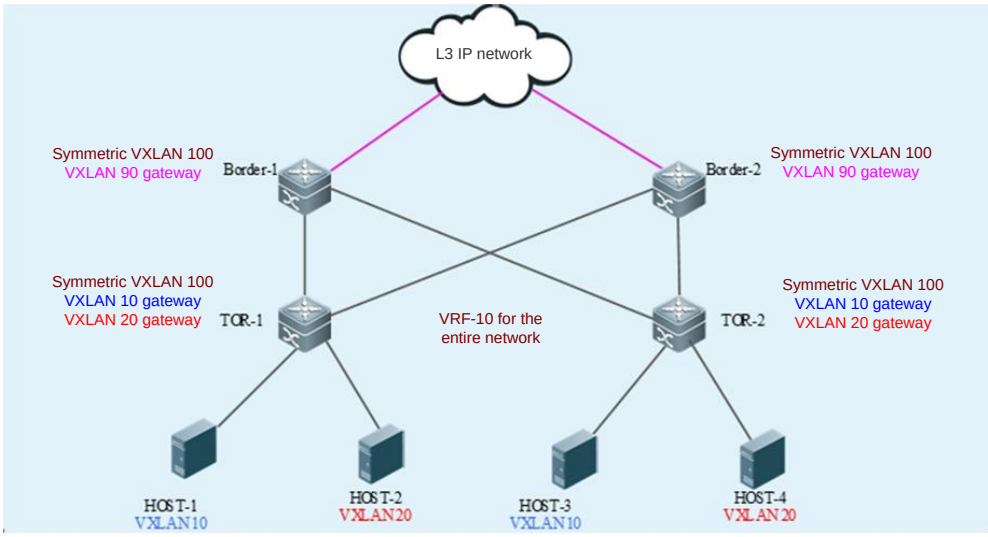
1.4.2.5. Configuring EVPN-based Single-tenant VXLAN Routing Scenario

Scenario Figure 1-27



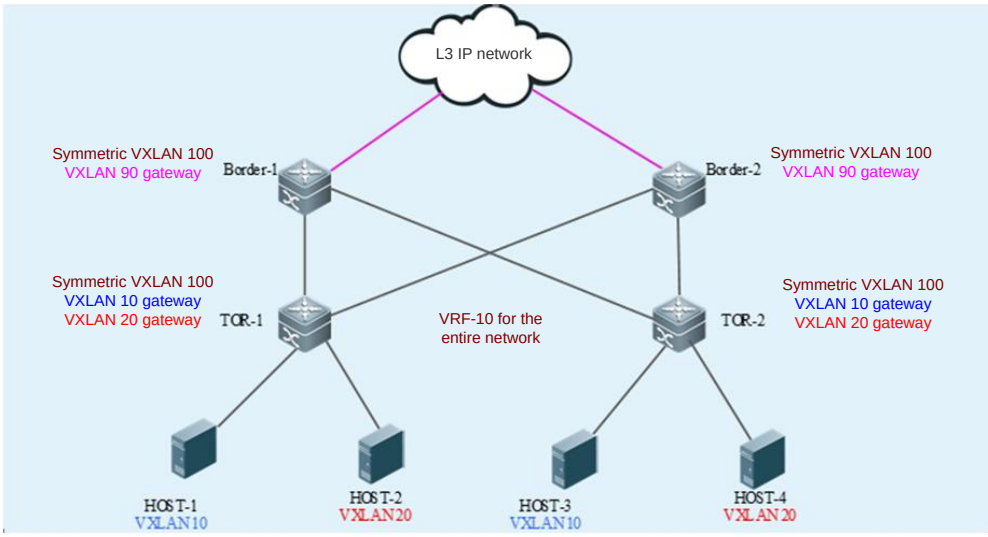
Configuration Steps	• Configure an IPv4 unicast routing protocol such as the OSPF protocol on Border-1, Border-2, TOR-1, and TOR-2 to ensure that unicast routes are reachable. • Configure the BGP-EVPN routing protocol on Border-1, Border-2, TOR-1, and TOR-2 to establish BGP neighbor relationships between the devices (except between Border-1 and Border-2) and to support the EVPN protocol family. • Configure the EVI for BGP-EVPN on TOR-1 and TOR-2. For details, see *BGP-EVPN Configuration Guide*. • Configure a VXLAN on the virtual server and designate the gateway address of the virtual machine. • Associate the VTEP with the loopback interface on TOR-1, TOR-2, Border-1, and Border-2 to establish tunnels. • Configure the anycast gateway MAC address on TOR-1 and TOR-2 to ensure that all VXLAN anycast gateways on the network use the same MAC address. • Create VXLAN 10 and VXLAN 20 on TOR-1 and associate them with VLANs. • Create VXLAN 10 and VXLAN 20 on TOR-2 and associate them with VLANs. • Create VXLAN 90 on Border-1 and associate VXLAN 90 with a VLAN. • Create VXLAN 90 on Border-2 and associate VXLAN 90 with a VLAN. • Create overlay router interfaces for VXLAN 10 and VXLAN 20 on TOR-1 and TOR-2, and configure the VXLAN gateway IP address for them. Configure the same VRF network for the overlay router interfaces to determine their respective tenants. Configure the anycast gateway to ensure that all VXLAN gateways on the network use the same IP address and MAC address. As the anycast gateway function is enabled, the overlay

Scenario Figure 1-27



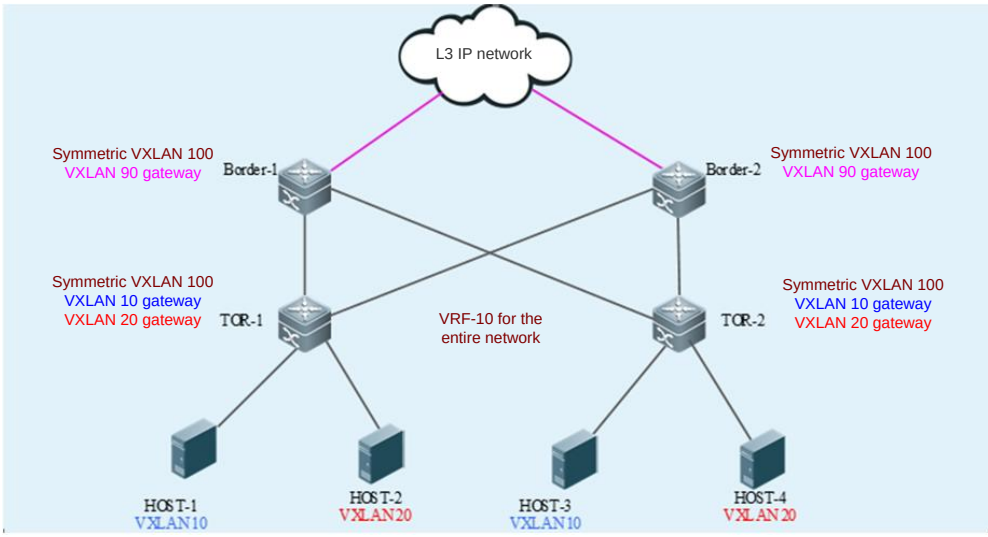
	router interfaces associated with the same VXLAN on TOR-1 and TOR-2 must be configured with the same VXLAN gateway IP address. • Create VXLAN 100 on TOR-1, TOR-2, Border-1, and Border-2. Configure VXLAN 100 as a symmetric VXLAN to serve as the L3 routing VXLAN of the corresponding VRF network. L3 routes between all VXLANs of the same VRF network are advertised via the symmetric VXLAN. In addition, the symmetric VXLAN is also used for L3 routing and forwarding. • Create overlay router interfaces for VXLAN 100 on TOR-1 and TOR-2 and configure the same VRF network for the overlay router interfaces. VXLAN 100 serves as the symmetric VXLAN of the VRF network. • Create overlay router interfaces for VXLAN 100 on Border-1 and Border-2, and configure the same VRF network for the overlay router interfaces, so that VXLAN 100 serves as the symmetric VXLAN of the VRF network. Configure VXLAN gateway IP addresses for Border-1 and Border-2 (different IP addresses for different devices). • Create overlay router interfaces for VXLAN 90 on Border-1 and Border-2. Configure the same VRF network for the overlay router interfaces and configure the VXLAN gateway IP address. • Associate VXLAN instances with overlay router interfaces on TOR-1, TOR-2, Border-1, and Border-2 to realize VXLAN routing. • (Optional) Configure ARP suppression on TOR-1 and TOR-2 to reduce the ARP packets entering the VXLAN.
HOST	Configuring the IP address and gateway according to Figure 2-27 (the detailed configuration on the server is omitted herein).
TOR1	TOR1# configure terminal Enter configuration commands, one per line. End with CNTL/Z.

Scenario Figure 1-27



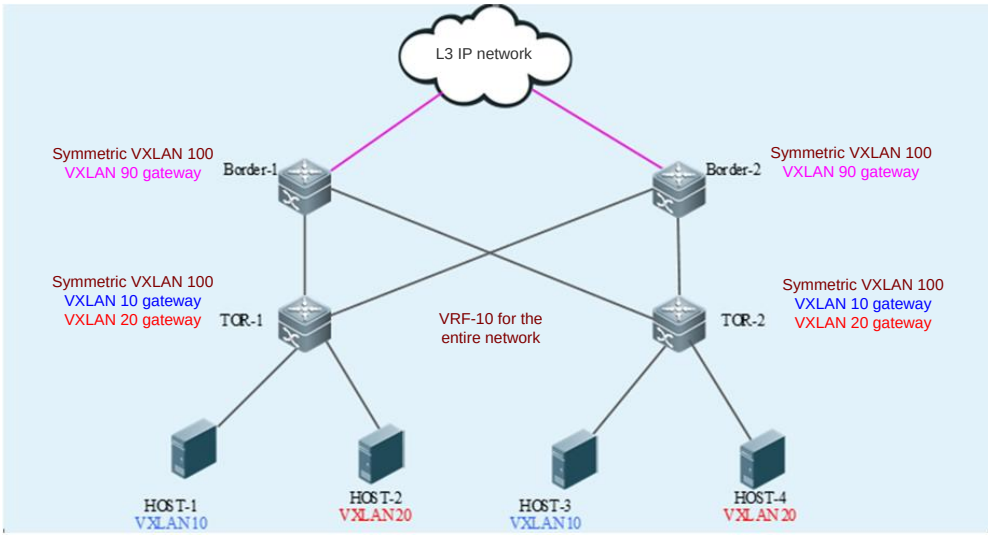

```
TOR1(config)# interface Loopback 1
TOR1(config-if- Loopback 1)# ip address 1.1.1.1/32
TOR1(config-if- Loopback 1)# exit
TOR1(config)# vtep
TOR1(config-vtep)# source loopback 1
TOR1(config-vtep)# arp suppress enable
TOR1(config-vtep)# exit
TOR1(config)# fabric anycast-gateway-mac 0011.2233.2016
TOR1(config)# ip vrf vrf-10
TOR1(config-vrf)# rd 10:10
TOR1(config-vrf)# route-target export 1000:1000
TOR1(config-vrf)# exit
TOR1(config)# int overlayrouter 10
TOR1(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10
TOR1(config-if-OverlayRouter 10)# ip address 10.1.1.1/24
TOR1(config-if-OverlayRouter 10)# anycast-gateway
TOR1(config-if-OverlayRouter 10)# exit
TOR1(config)# int overlayrouter 20
TOR1(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10
TOR1(config-if-OverlayRouter 20)# ip address 20.1.1.1/24
TOR1(config-if-OverlayRouter 20)# anycast-gateway
TOR1(config-if-OverlayRouter 20)# exit
TOR1(config)# int overlayrouter 100
TOR1(config-if-OverlayRouter 100)# ip vrf forwarding vrf-10
TOR1(config-if-OverlayRouter 100)# ip address 100.1.4.1/24
```

Scenario Figure 1-27



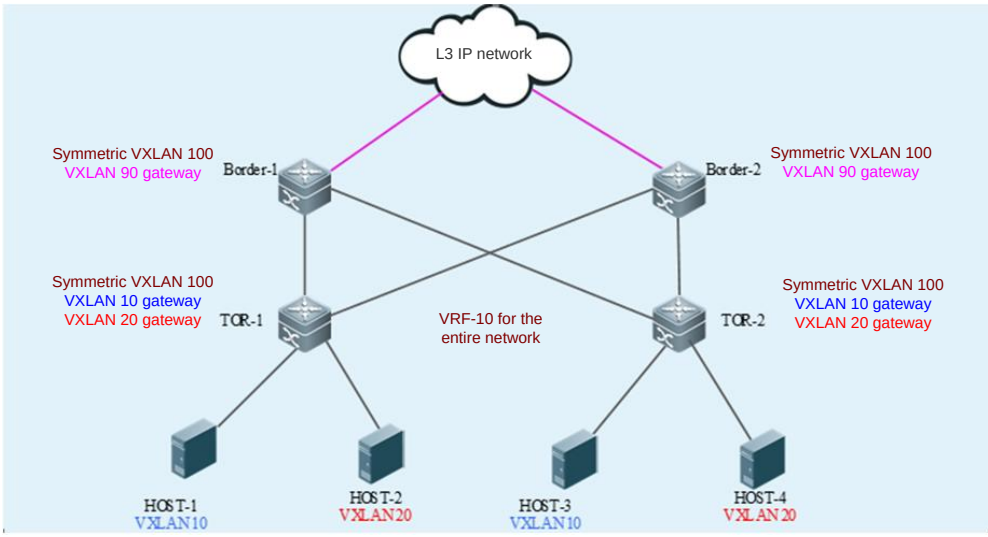

```
TOR1(config-if-OverlayRouter 100)# exit
TOR1(config)# vxlan 10
TOR1(config-vxlan)# extend-vlan 10
TOR1(config-vxlan)# router-interface OverlayRouter 10
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 20
TOR1(config-vxlan)# extend-vlan 20
TOR1(config-vxlan)# router-interface OverlayRouter 20
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 100
TOR1(config-vxlan)# symmetric
TOR1(config-vxlan)# router-interface OverlayRouter 100
TOR1(config-vxlan)# exit
TOR1(config)# router bgp 64512
TOR1(config-router)# neighbor 2.2.2.2 remote-as 64512
TOR1(config-router)# neighbor 2.2.2.2 update-source loopback 1
TOR1(config-router)# neighbor 3.3.3.3 remote-as 64512
TOR1(config-router)# neighbor 3.3.3.3 update-source loopback 1
TOR1(config-router)# neighbor 4.4.4.4 remote-as 64512
TOR1(config-router)# neighbor 4.4.4.4 update-source loopback 1
TOR1(config-router)# address-family l2vpn evpn
TOR1(config-router-af)# neighbor 2.2.2.2 activate
TOR1(config-router-af)# neighbor 3.3.3.3 activate
```

Scenario Figure 1-27



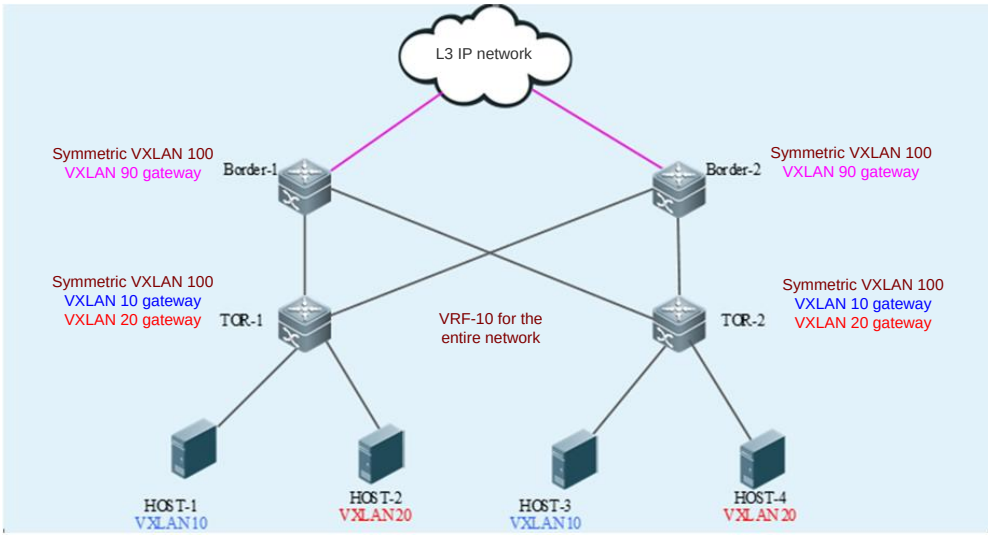
	TOR1(config-router-af)# neighbor 4.4.4.4 activate TOR1(config-router-af)# advertise ipv4 unicast TOR1(config-router-af)# exit TOR1(config-router)# address-family ipv4 vrf vrf-10 TOR1(config-router-af)# redistribute connected TOR1(config-router-af)# exit TOR1(config-router)# exit TOR1(config)# evpn TOR1(config-evpn)# vni 10 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 20 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 100 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# route-target import 1000:1000 TOR1(config-evpn-vni)# exit
TOR2	TOR2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR2(config)# interface Loopback 1

Scenario Figure 1-27



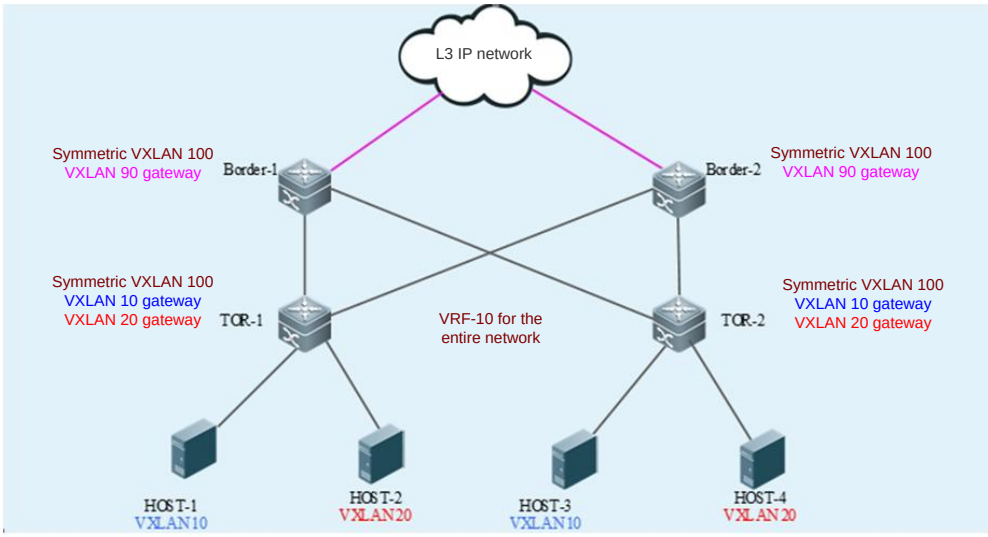

```
TOR2(config-if- Loopback 1)# ip address 2.2.2.2/32
TOR2(config-if- Loopback 1)# exit
TOR2(config)# vtep
TOR2(config-vtep)# source loopback 1
TOR2(config-vtep)# arp suppress enable
TOR2(config-vtep)# exit
TOR2(config)# fabric anycast-gateway-mac 0011.2233.2016
TOR2(config)# ip vrf vrf-10
TOR2(config-vrf)# rd 10:10
TOR2(config-vrf)# route-target export 1000:1000
TOR2(config-vrf)# exit
TOR2(config)# int overlayrouter 10
TOR2(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10
TOR2(config-if-OverlayRouter 10)# ip address 10.1.1.1/24
TOR2(config-if-OverlayRouter 10)# anycast-gateway
TOR2(config-if-OverlayRouter 10)# exit
TOR2(config)# int overlayrouter 20
TOR2(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10
TOR2(config-if-OverlayRouter 20)# ip address 20.1.1.1/24
TOR2(config-if-OverlayRouter 20)# anycast-gateway
TOR2(config-if-OverlayRouter 20)# exit
TOR2(config)# int overlayrouter 100
TOR2(config-if-OverlayRouter 100)# ip vrf forwarding vrf-10
TOR2(config-if-OverlayRouter 100)# ip address 100.1.3.1/24
TOR2(config-if-OverlayRouter 100)# exit
```

Scenario Figure 1-27



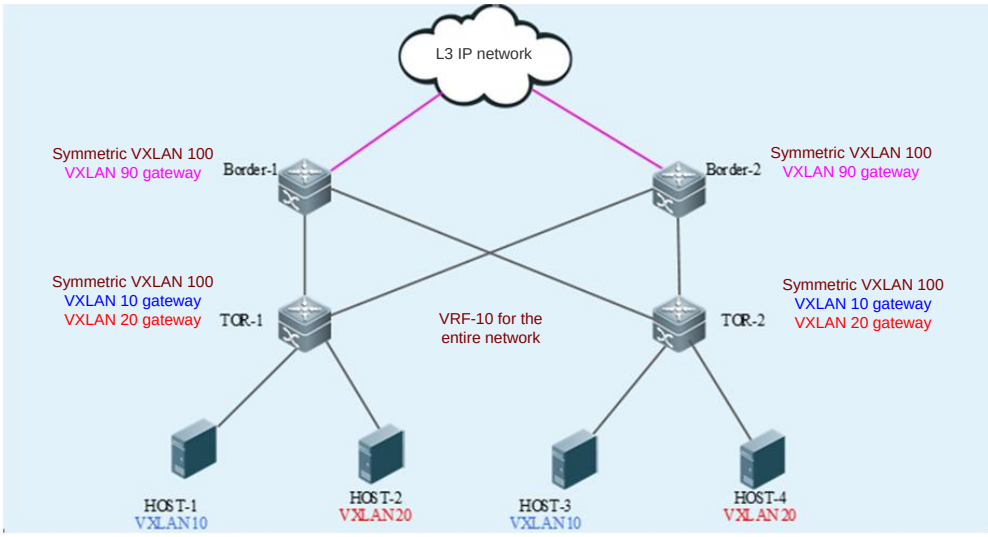

```
TOR2(config)# vxlan 10
TOR2(config-vxlan)# extend-vlan 10
TOR2(config-vxlan)# router-interface OverlayRouter 10
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# vxlan 20
TOR2(config-vxlan)# extend-vlan 20
TOR2(config-vxlan)# router-interface OverlayRouter 20
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# vxlan 100
TOR2(config-vxlan)# symmetric
TOR2(config-vxlan)# router-interface OverlayRouter 100
TOR2(config-vxlan)# exit
TOR2(config)# router bgp 64512
TOR2(config-router)# neighbor 1.1.1.1 remote-as 64512
TOR2(config-router)# neighbor 1.1.1.1 update-source loopback 1
TOR2(config-router)# neighbor 3.3.3.3 remote-as 64512
TOR2(config-router)# neighbor 3.3.3.3 update-source loopback 1
TOR2(config-router)# neighbor 4.4.4.4 remote-as 64512
TOR2(config-router)# neighbor 4.4.4.4 update-source loopback 1
TOR2(config-router)# address-family l2vpn evpn
TOR2(config-router-af)# neighbor 1.1.1.1 activate
TOR2(config-router-af)# neighbor 3.3.3.3 activate
TOR2(config-router-af)# neighbor 4.4.4.4 activate
```

Scenario Figure 1-27



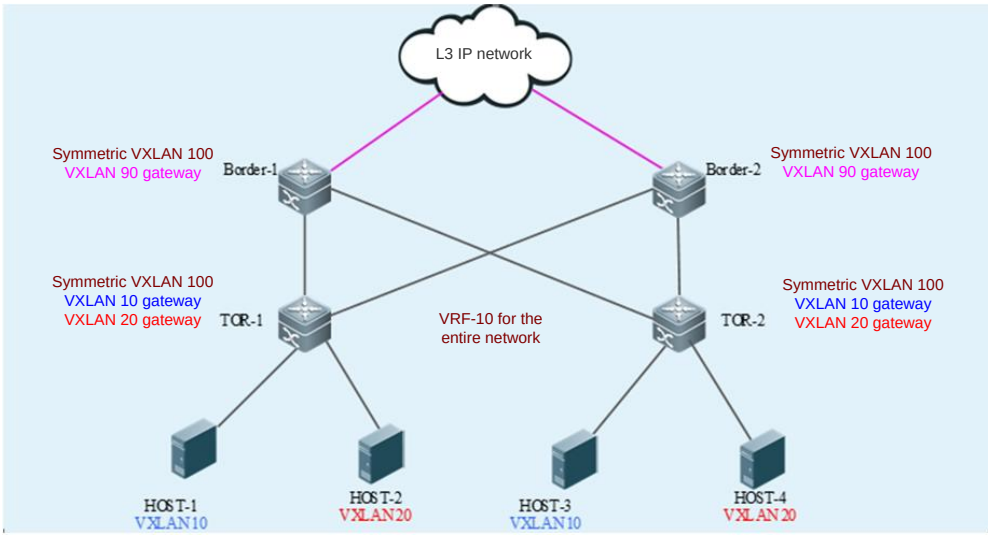
	TOR2(config-router-af)# advertise ipv4 unicast TOR2(config-router-af)# exit TOR2(config-router)# address-family ipv4 vrf vrf-10 TOR2(config-router-af)# redistribute connected TOR2(config-router-af)# exit TOR2(config-router)# exit TOR2(config)# evpn TOR2(config-evpn)# vni 10 TOR2(config-evpn-vni)# rd auto TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# exit TOR2(config-evpn)# vni 20 TOR2(config-evpn-vni)# rd auto TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# exit TOR2(config-evpn)# vni 100 TOR2(config-evpn-vni)# rd auto TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# route-target import 1000:1000 TOR2(config-evpn-vni)# exit
Border1	Border1# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Border1(config)# interface Loopback 1 Border1(config-if- Loopback 1)# ip address 3.3.3.3/32

Scenario Figure 1-27



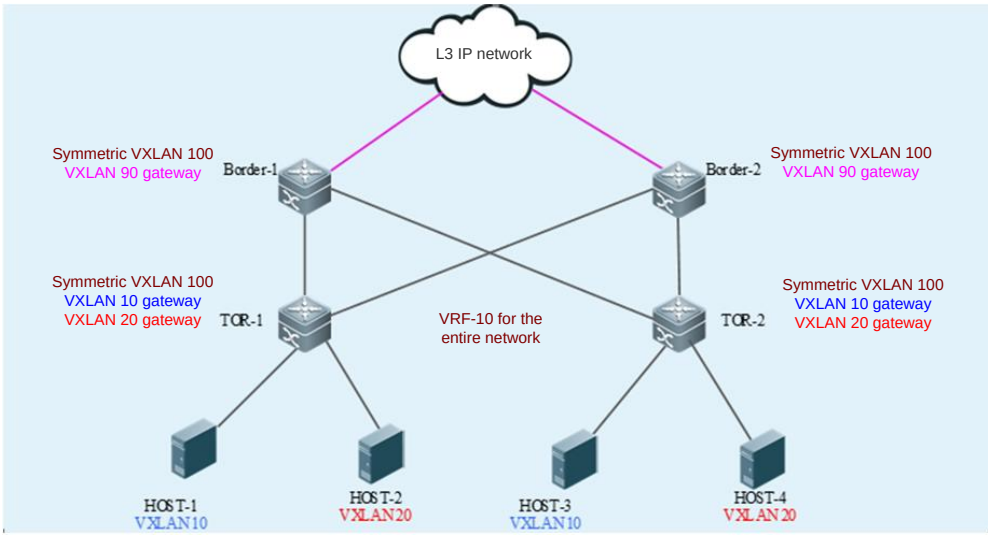

```
Border1(config-if- Loopback 1)# exit
Border1(config)# vtep
Border1(config-vtep)# source loopback 1
Border1(config-vtep)# arp suppress enable
Border1(config-vtep)# exit
Border1(config)# fabric anycast-gateway-mac 0011.2233.2016
Border1(config)# ip vrf vrf-10
Border1(config-vrf)# rd 10:10
Border1(config-vrf)# route-target export 1000:1000
Border1(config-vrf)# exit
Border1(config)# int overlayrouter 90
Border1(config-if-OverlayRouter 90)# ip vrf forwarding vrf-10
Border1(config-if-OverlayRouter 90)# ip address 90.1.1.1/24
Border1(config-if-OverlayRouter 90)# anycast-gateway
Border1(config-if-OverlayRouter 90)# exit
Border1(config)# vxlan 90
Border1(config-vxlan)# extend-vlan 90
Border1(config-vxlan)# router-interface OverlayRouter 90
Border1(config-vxlan)# arp suppress enable
Border1(config-vxlan)# exit
Border1(config)# int overlayrouter 100
Border1(config-if-OverlayRouter 100)# ip vrf forwarding vrf-10
Border1(config-if-OverlayRouter 100)# ip address 100.1.1.1/24
Border1(config-if-OverlayRouter 100)# exit
Border1(config)# vxlan 100
```

Scenario Figure 1-27



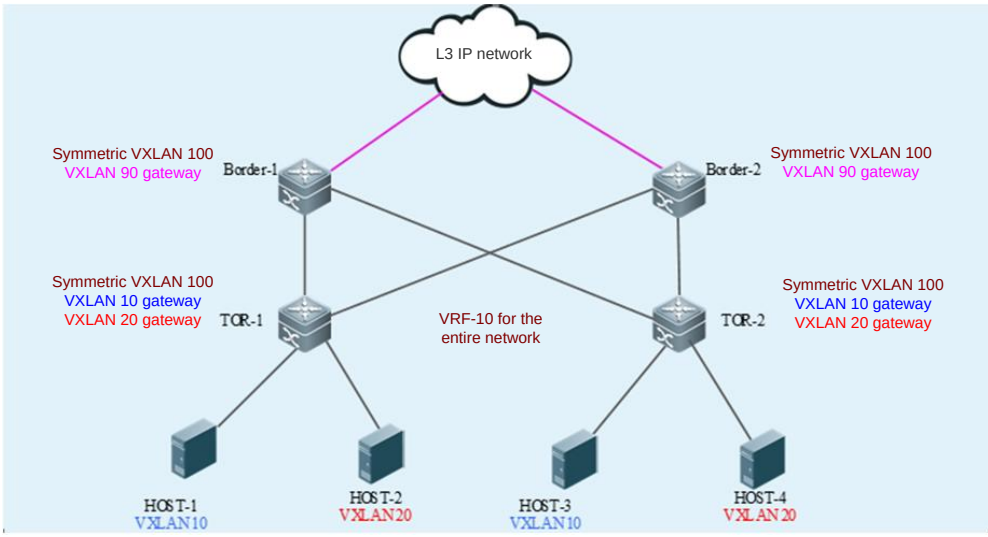

```
Border1(config-vxlan)# symmetric
Border1(config-vxlan)# router-interface OverlayRouter 100
Border1(config-vxlan)# exit
Border1(config)# router bgp 64512
Border1(config-router)# neighbor 1.1.1.1 remote-as 64512
Border1(config-router)# neighbor 1.1.1.1 update-source loopback 1
Border1(config-router)# neighbor 2.2.2.2 remote-as 64512
Border1(config-router)# neighbor 2.2.2.2 update-source loopback 1
Border1(config-router)# neighbor 4.4.4.4 remote-as 64512
Border1(config-router)# neighbor 4.4.4.4 update-source loopback 1
Border1(config-router)# address-family l2vpn evpn
Border1(config-router-af)# neighbor 1.1.1.1 activate
Border1(config-router-af)# neighbor 2.2.2.2 activate
Border1(config-router-af)# neighbor 4.4.4.4 activate
Border1(config-router-af)# advertise ipv4 unicast
Border1(config-router-af)# exit
Border1(config-router)# address-family ipv4 vrf vrf-10
Border1(config-router-af)# redistribute static
Border1(config-router-af)# exit
Border1(config-router)# exit
Border1(config)# evpn
Border1(config-evpn)# vni 100
Border1(config-evpn-vni)# rd auto
Border1(config-evpn-vni)# route-target both auto
Border1(config-evpn-vni)# route-target import 1000:1000
```

Scenario Figure 1-27



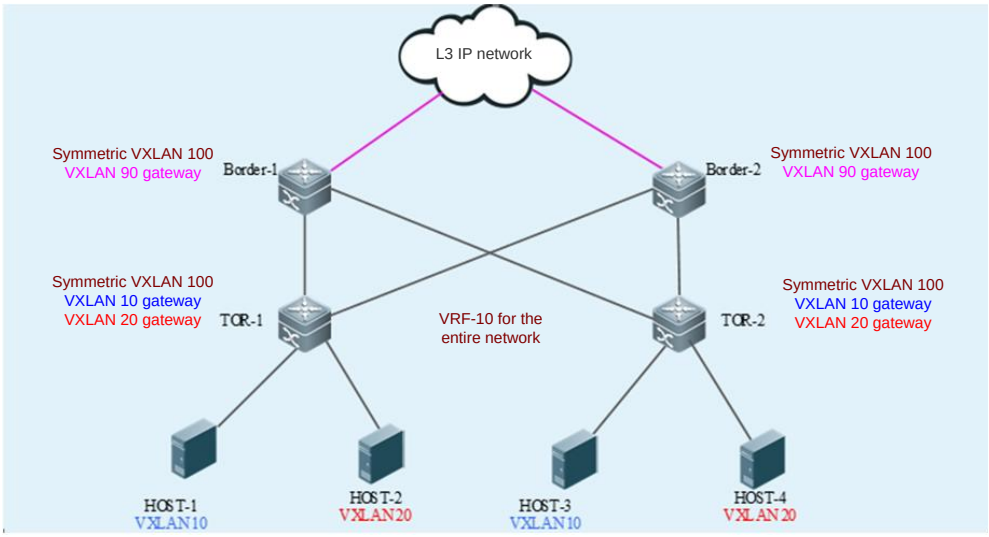
	Border1(config-evpn-vni)# exit
Border2	Border2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Border2(config)# interface Loopback 1 Border2(config-if- Loopback 1)# ip address 4.4.4.4/32 Border2(config-if- Loopback 1)# exit Border2(config)# vtep Border2(config-vtep)# source loopback 1 Border2(config-vtep)# arp suppress enable Border2(config-vtep)# exit Border2(config)# fabric anycast-gateway-mac 0011.2233.2016 Border2(config)# ip vrf vrf-10 Border2(config-vrf)# rd 10:10 Border2(config-vrf)# route-target export 1000:1000 Border2(config-vrf)# exit Border2(config)# int overlayrouter 90 Border2(config-if-OverlayRouter 90)# ip vrf forwarding vrf-10 Border2(config-if-OverlayRouter 90)# ip address 90.1.2.1/24 Border2(config-if-OverlayRouter 90)# anycast-gateway Border2(config-if-OverlayRouter 90)# exit Border2(config-vxlan)# arp suppress enable Border2(config-vxlan)# exit Border2(config)# int overlayrouter 100 Border2(config-if-OverlayRouter 100)# ip vrf forwarding vrf-10

Scenario Figure 1-27




```
Border2(config-if-OverlayRouter 100)# ip address 100.1.2.1/24
Border2(config-if-OverlayRouter 100)# exit
Border2(config)# vxlan 90
Border2(config-vxlan)# extend-vlan 90
Border2(config-vxlan)# router-interface OverlayRouter 90
Border2(config-vxlan)# arp suppress enable
Border2(config-vxlan)# exit
Border2(config)# vxlan 100
Border2(config-vxlan)# symmetric
Border2(config-vxlan)# router-interface OverlayRouter 100
Border2(config-vxlan)# exit
Border2(config)# router bgp 64512
Border2(config-router)# neighbor 1.1.1.1 remote-as 64512
Border2(config-router)# neighbor 1.1.1.1 update-source loopback 1
Border2(config-router)# neighbor 2.2.2.2 remote-as 64512
Border2(config-router)# neighbor 2.2.2.2 update-source loopback 1
Border2(config-router)# neighbor 3.3.3.3 remote-as 64512
Border2(config-router)# neighbor 3.3.3.3 update-source loopback 1
Border2(config-router)# address-family l2vpn evpn
Border2(config-router-af)# neighbor 1.1.1.1 activate
Border2(config-router-af)# neighbor 2.2.2.2 activate
Border2(config-router-af)# neighbor 3.3.3.3 activate
Border2(config-router-af)# advertise ipv4 unicast
Border2(config-router-af)# exit
Border2(config-router)# address-family ipv4 vrf vrf-10
```

Scenario Figure 1-27



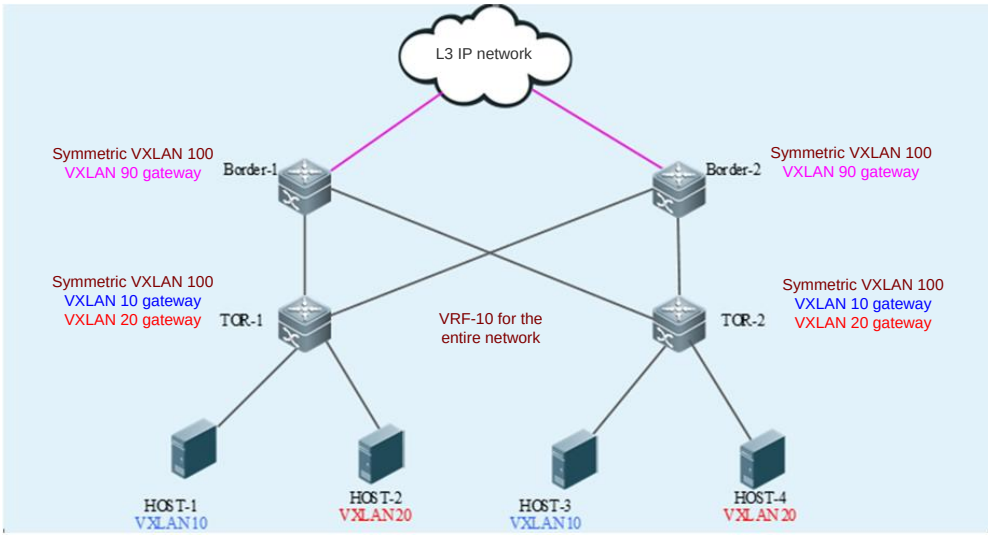

```
Border2(config-router-af)# redistribute static
Border2(config-router-af)# exit
Border2(config-router)# exit
Border2(config)# evpn
Border2(config-evpn)#vni 100
Border2(config-evpn-vni)# rd auto
Border2(config-evpn-vni)# route-target both auto
Border2(config-evpn-vni)# route-target import 1000:1000
Border2(config-evpn-vni)# exit
```

Verification

- Verify that HOST-1, HOST-2, HOST-3, and HOST-4 can ping each other.
- Verify that the virtual machines can be migrated between the HOSTs on the same VXLAN and can access the network normally after migration without modifying the configuration.

```
Border1# sh vxlan
VXLAN Total Count: 3
VXLAN Capacity   : 8000

VXLAN 90
   Symmetric property  : FALSE
   Router Interface    : overlayrouter 90 (anycast)
   Extend VLAN         : 90
VTEP Adjacency Count: 1
Interface          Source IP      Destination IP  Type
```

Scenario Figure 1-27



```
---------------------- --------------- --------------- -------
OverlayTunnel 6146     3.3.3.3         2.2.2.2         dynamic
```

VXLAN 100

Symmetric property : TRUE

Router Interface : overlayrouter 100 (non-anycast)

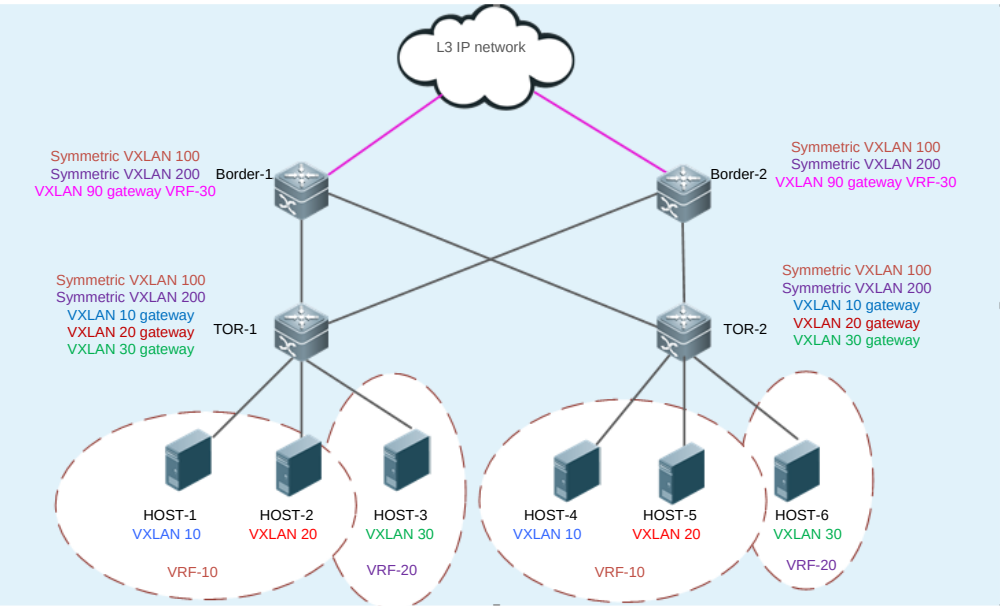
Extend VLAN : -

VTEP Adjacency Count: 1

```
      Interface          Source IP       Destination IP  Type
---------------------- --------------- --------------- -------
OverlayTunnel 6146     3.3.3.3         2.2.2.2         dynamic
```

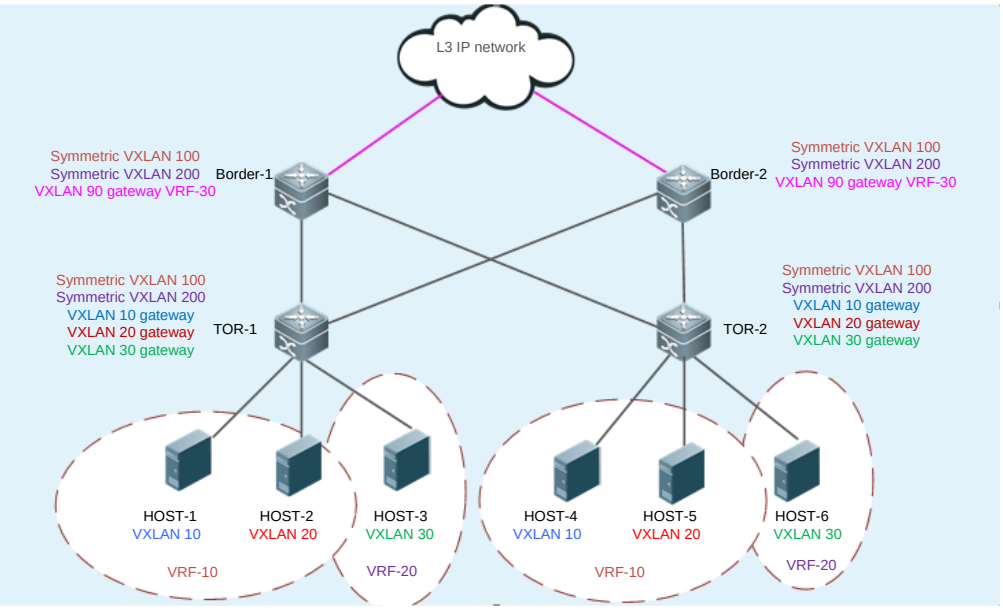
1.4.2.6. Configuring EVPN-based Multi-tenant VXLAN Routing Scenario

Scenario
Figure 1-28



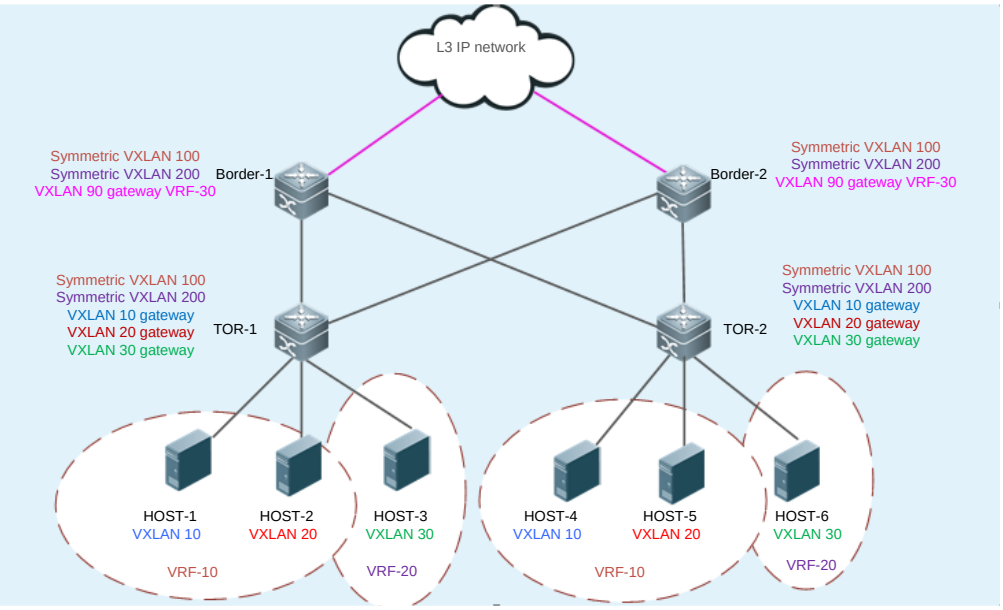
Configuration Steps	• Configure an IPv4 unicast routing protocol such as the OSPF protocol on Border-1, Border-2, TOR-1, and TOR-2 to ensure that unicast routes are reachable. • Configure the BGP-EVPN routing protocol on Border-1, Border-2, TOR-1, and TOR-2 to establish BGP neighbor relationships between the devices (except between Border-1 and Border-2) and to support the EVPN protocol family. • Configure the EVI for BGP-EVPN on TOR-1 and TOR-2. For details, see *BGP-EVPN Configuration Guide*. • Configure a VXLAN on the virtual server and designate the gateway address of the virtual machine. • Associate the VTEP with the loopback interface on TOR-1, TOR-2, Border-1, and Border-2 to establish tunnels. • Configure the anycast gateway MAC address on TOR-1 and TOR-2 to ensure that all VXLAN anycast gateways on the network use the same MAC address. • Create VXLAN 10, VXLAN 20, and VXLAN 30 on TOR-1 and associate them with VLANs. • Create VXLAN 10, VXLAN 20, and VXLAN 30 on TOR-2 and associate them with VLANs. • Create VXLAN 90 on Border-1 and associate VXLAN 90 with a VLAN. • Create VXLAN 90 on Border-2 and associate VXLAN 90 with a VLAN. • Create overlay router interfaces for VXLAN 10, VXLAN 20, and VXLAN 30 on TOR-1 and TOR-2 and configure the VXLAN gateway IP address for them. Configure different VRF networks for different overlay router interfaces to determine their respective

Scenario
Figure 1-28



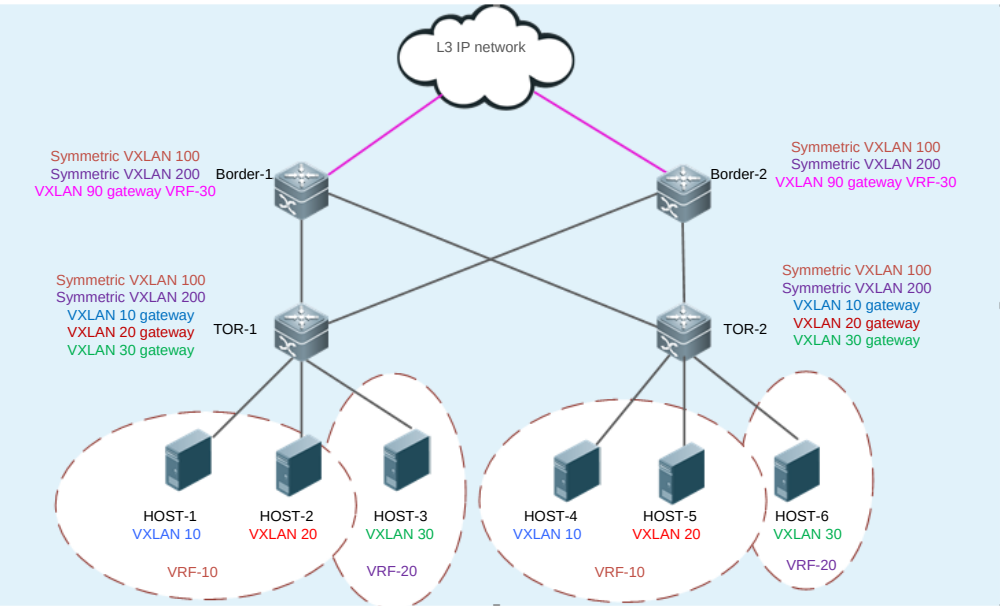
	tenants. Configure the anycast gateway to ensure that all VXLAN gateways on the network use the same IP address and MAC address. As the anycast gateway function is enabled, the overlay router interfaces associated with the same VXLAN on TOR-1 and TOR-2 must be configured with the same VXLAN gateway IP address. • Create overlay router interfaces for VXLAN 100 and VXLAN 200 on TOR-1 and TOR-2 and configure different VRF networks for the overlay router interfaces. VXLAN 100 and VXLAN 200 serve as the symmetric VXLANs of the corresponding VRF networks. • Create overlay router interfaces for VXLAN 100 and VXLAN 200 on Border-1 and Border-2. Configure different VRF networks for the overlay router interfaces so that VXLAN 100 and VXLAN 200 serve as the symmetric VXLANs of the corresponding VRF networks. Configure VXLAN gateway IP addresses for Border-1 and Border-2 (different IP addresses for different devices). • Create overlay router interfaces for VXLAN 90 on Border-1 and Border-2. Configure different VRF networks for the OverlayRouter interfaces and configure the VXLAN gateway IP address. • Associate VXLAN instances with overlay router interfaces on TOR-1, TOR-2, Border-1, and Border-2 to realize VXLAN routing. • (Optional) Configure ARP suppression on TOR-1 and TOR-2 to reduce the ARP packets entering the VXLAN.
HOST	Configuring the IP address and gateway according to Figure 1-28 (the detailed configuration on the server is omitted herein).
	TOR1# configure terminal Enter configuration commands, one per line. End with CNTL/Z.

Scenario
Figure 1-28



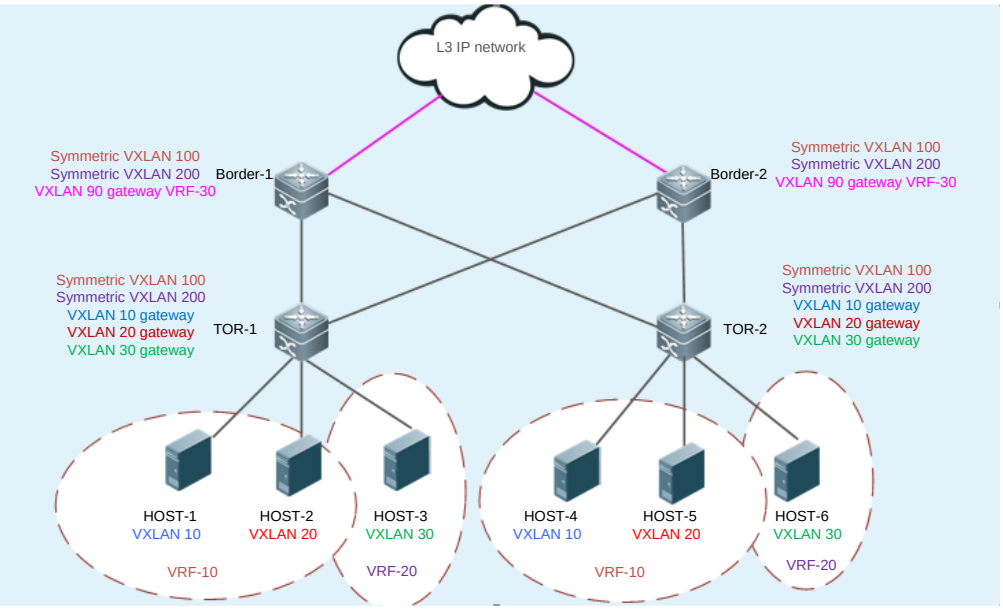

```
TOR1(config)# interface Loopback 1
TOR1(config-if- Loopback 1)# ip address 1.1.1.1/32
TOR1(config-if- Loopback 1)# exit
TOR1(config)# vtep
TOR1(config-vtep)# source loopback 1
TOR1(config-vtep)# arp suppress enable
TOR1(config)# fabric anycast-gateway-mac 0011.2233.2016
TOR1(config-vtep)# exit
TOR1(config)# ip vrf vrf-10
TOR1(config-vrf)# rd 10:10
TOR1(config-vrf)# route-target export 1000:1000
TOR1(config-vrf)# exit
TOR1(config)# ip vrf vrf-20
TOR1(config-vrf)# rd 20:20
TOR1(config-vrf)# route-target export 2000:2000
TOR1(config-vrf)# exit
TOR1(config)# int overlayrouter 10
TOR1(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10
TOR1(config-if-OverlayRouter 10)# ip address 10.1.1.1/24
TOR1(config-if-OverlayRouter 10)# anycast-gateway
TOR1(config-if-OverlayRouter 10)# exit
TOR1(config-vxlan)# exit
TOR1(config)# int overlayrouter 20
```

Scenario
Figure 1-28



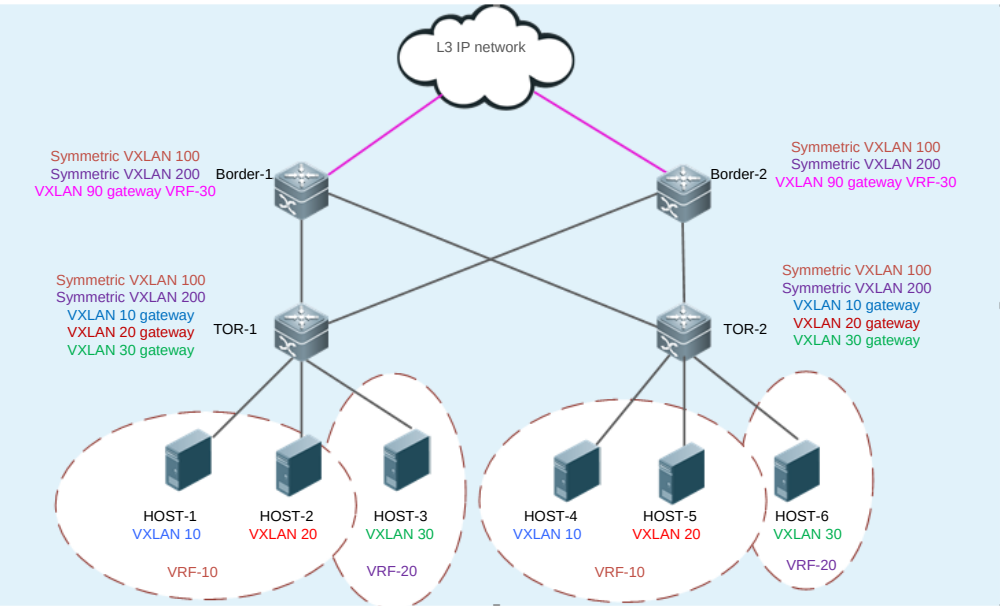

```
TOR1(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10
TOR1(config-if-OverlayRouter 20)# ip address 20.1.1.1/24
TOR1(config-if-OverlayRouter 20)# anycast-gateway
TOR1(config-if-OverlayRouter 20)# exit
TOR1(config)# int overlayrouter 30
TOR1(config-if-OverlayRouter 30)# ip vrf forwarding vrf-20
TOR1(config-if-OverlayRouter 30)# ip address 30.1.1.1/24
TOR1(config-if-OverlayRouter 30)# anycast-gateway
TOR1(config-if-OverlayRouter 30)# exit
TOR1(config)# int overlayrouter 100
TOR1(config-if-OverlayRouter 100)# ip vrf forwarding vrf-10
TOR1(config-if-OverlayRouter 100)# ip address 100.1.4.1/24
TOR1(config-if-OverlayRouter 100)# exit
TOR1(config)# int overlayrouter 200
TOR1(config-if-OverlayRouter 200)# ip vrf forwarding vrf-20
TOR1(config-if-OverlayRouter 200)# ip address 200.1.4.1/24
TOR1(config-if-OverlayRouter 200)# exit
TOR1(config)# vxlan 10
TOR1(config-vxlan)# extend-vlan 10
TOR1(config-vxlan)# router-interface OverlayRouter 10
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 20
```

Scenario
Figure 1-28



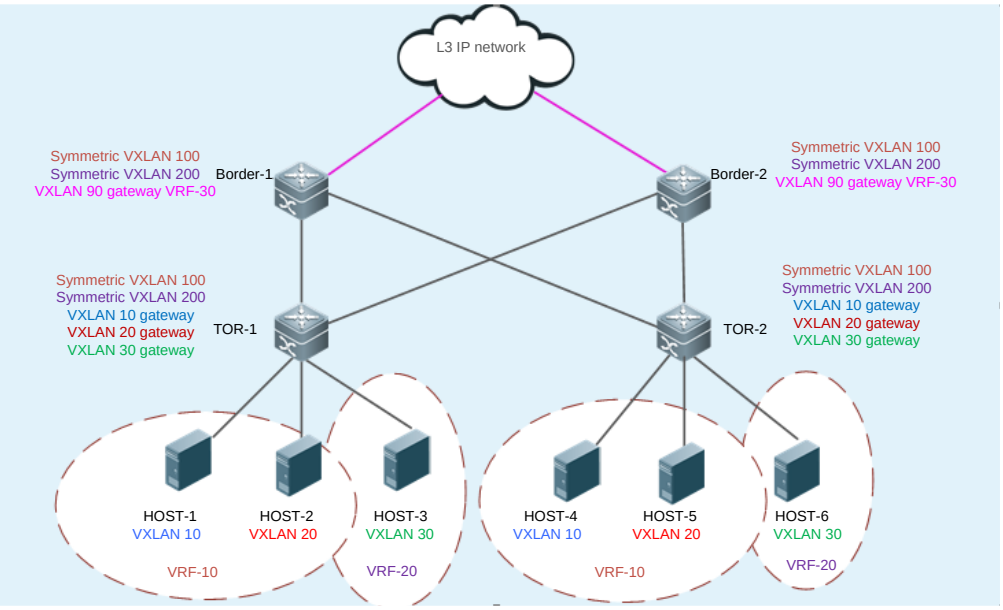

```
TOR1(config-vxlan)# extend-vlan 20
TOR1(config-vxlan)# router-interface OverlayRouter 20
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 100
TOR1(config-vxlan)# symmetric
TOR1(config-vxlan)# router-interface OverlayRouter 100
TOR1(config)# vxlan 30
TOR1(config-vxlan)# extend-vlan 30
TOR1(config-vxlan)# router-interface OverlayRouter 30
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 200
TOR1(config-vxlan)# symmetric
TOR1(config-vxlan)# router-interface OverlayRouter 200
TOR1(config-vxlan)# exit
TOR1(config)# router bgp 64512
TOR1(config-router)# neighbor 2.2.2.2 remote-as 64512
TOR1(config-router)# neighbor 2.2.2.2 update-source loopback 1
TOR1(config-router)# neighbor 3.3.3.3 remote-as 64512
TOR1(config-router)# neighbor 3.3.3.3 update-source loopback 1
TOR1(config-router)# neighbor 4.4.4.4 remote-as 64512
TOR1(config-router)# neighbor 4.4.4.4 update-source loopback 1
```

Scenario
Figure 1-28



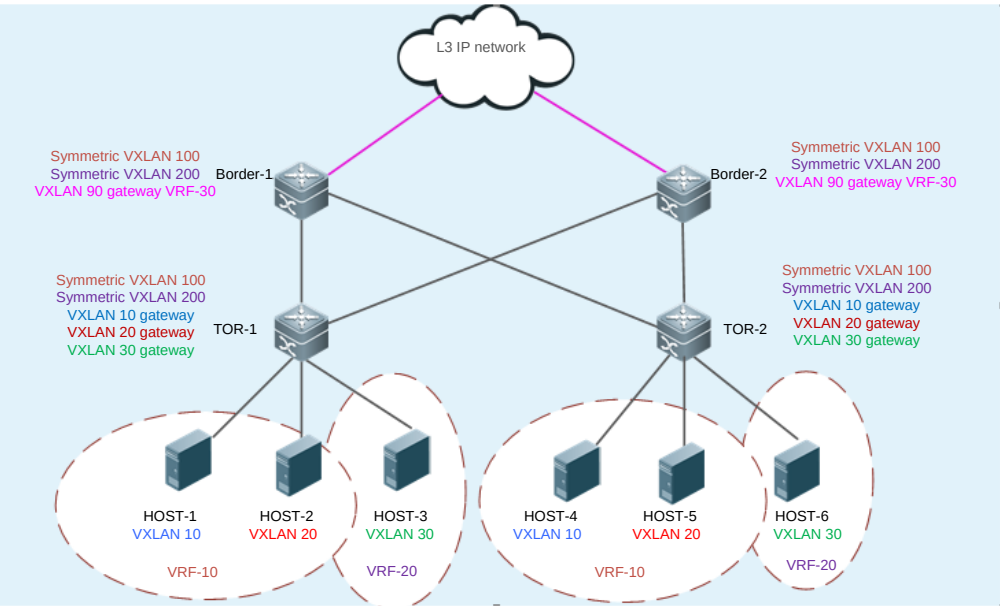

```
TOR1(config-router)# address-family l2vpn evpn
TOR1(config-router-af)# neighbor 2.2.2.2 activate
TOR1(config-router-af)# neighbor 3.3.3.3 activate
TOR1(config-router-af)# neighbor 4.4.4.4 activate
TOR1(config-router-af)# advertise ipv4 unicast
TOR1(config-router-af)# exit
TOR1(config-router)# address-family ipv4 vrf vrf-10
TOR1(config-router-af)# redistribute connected
TOR1(config-router-af)# exit
TOR1(config-router)# address-family ipv4 vrf vrf-20
TOR1(config-router-af)# redistribute connected
TOR1(config-router-af)# exit
TOR1(config-router)# exit
TOR1(config)# evpn
TOR1(config-evpn)# vni 10
TOR1(config-evpn-vni)# rd auto
TOR1(config-evpn-vni)# route-target both auto
TOR1(config-evpn-vni)# exit
TOR1(config-evpn)# vni 20
TOR1(config-evpn-vni)# rd auto
TOR1(config-evpn-vni)# route-target both auto
TOR1(config-evpn-vni)# exit
TOR1(config-evpn)# vni 30
```

Scenario
Figure 1-28



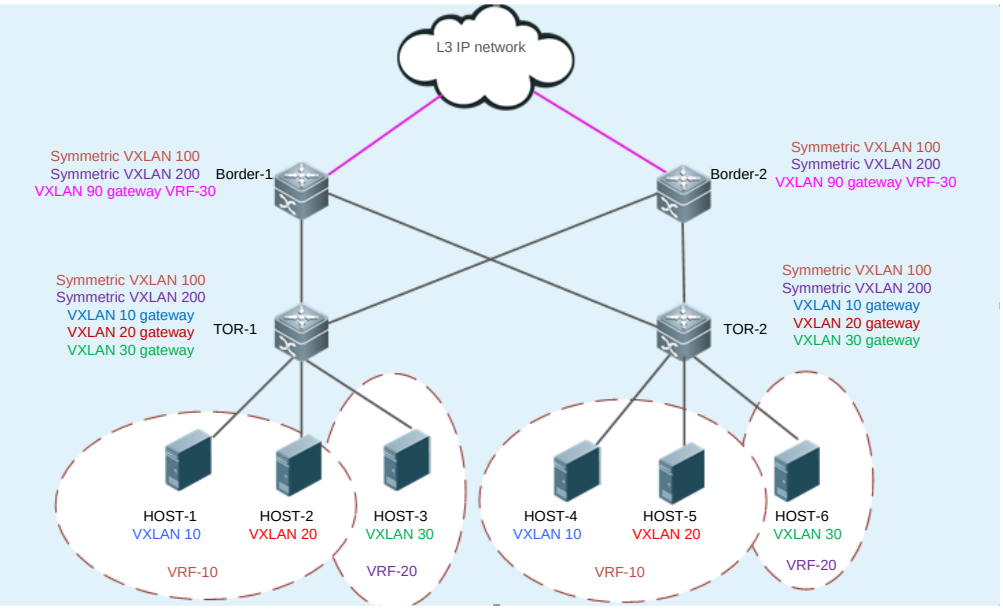
	TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 100 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# route-target import 1000:1000 TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 200 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# route-target import 2000:2000 TOR1(config-evpn-vni)# exit
TOR2	TOR2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR2(config)# interface Loopback 1 TOR2(config-if- Loopback 1)# ip address 2.2.2.2/32 TOR2(config-if- Loopback 1)# exit TOR2(config)# vtep TOR2(config-vtep)# source loopback 1 TOR2(config-vtep)# arp suppress enable TOR2(config-vtep)# exit

Scenario
Figure 1-28



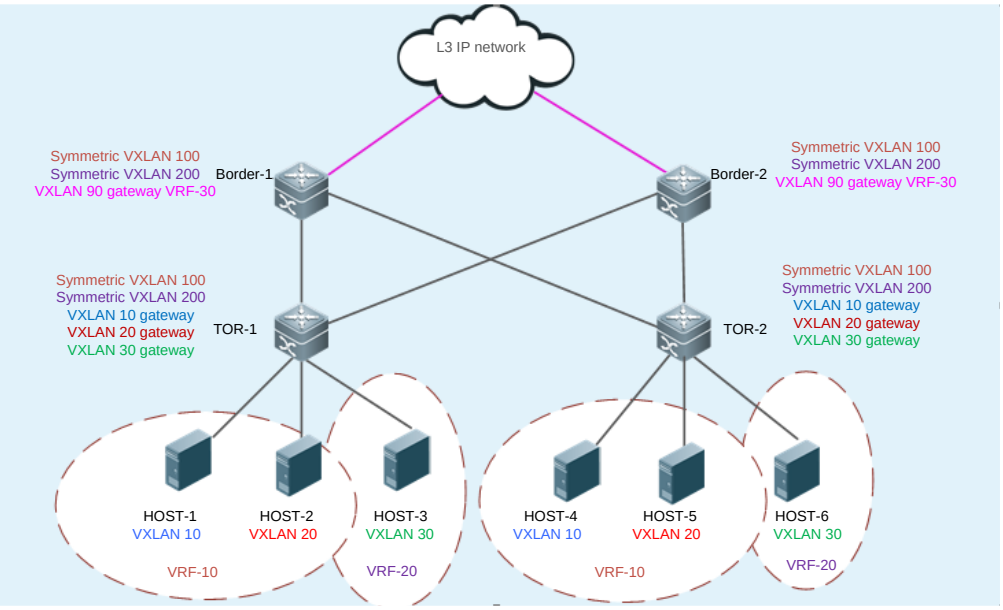

```
TOR2(config)# fabric anycast-gateway-mac 0011.2233.2016
TOR2(config)# ip vrf vrf-10
TOR2(config-vrf)# rd 10:10
TOR2(config-vrf)# route-target export 1000:1000
TOR2(config-vrf)# exit
TOR2(config)# ip vrf vrf-20
TOR2(config-vrf)# rd 20:20
TOR2(config-vrf)# route-target export 2000:2000
TOR2(config-vrf)# exit
TOR2(config)# int overlayrouter 10
TOR2(config-if-OverlayRouter 10)# ip vrf forwarding vrf-10
TOR2(config-if-OverlayRouter 10)# ip address 10.1.1.1/24
TOR2(config-if-OverlayRouter 10)# anycast-gateway
TOR2(config-if-OverlayRouter 10)# exit
TOR2(config)# int overlayrouter 20
TOR2(config-if-OverlayRouter 20)# ip vrf forwarding vrf-10
TOR2(config-if-OverlayRouter 20)# ip address 20.1.1.1/24
TOR2(config-if-OverlayRouter 20)# anycast-gateway
TOR2(config-if-OverlayRouter 20)# exit
TOR2(config)# int overlayrouter 100
TOR2(config-if-OverlayRouter 100)# ip vrf forwarding vrf-10
TOR2(config-if-OverlayRouter 100)# ip address 100.1.3.1/24
TOR2(config-if-OverlayRouter 100)# exit
```

Scenario
Figure 1-28



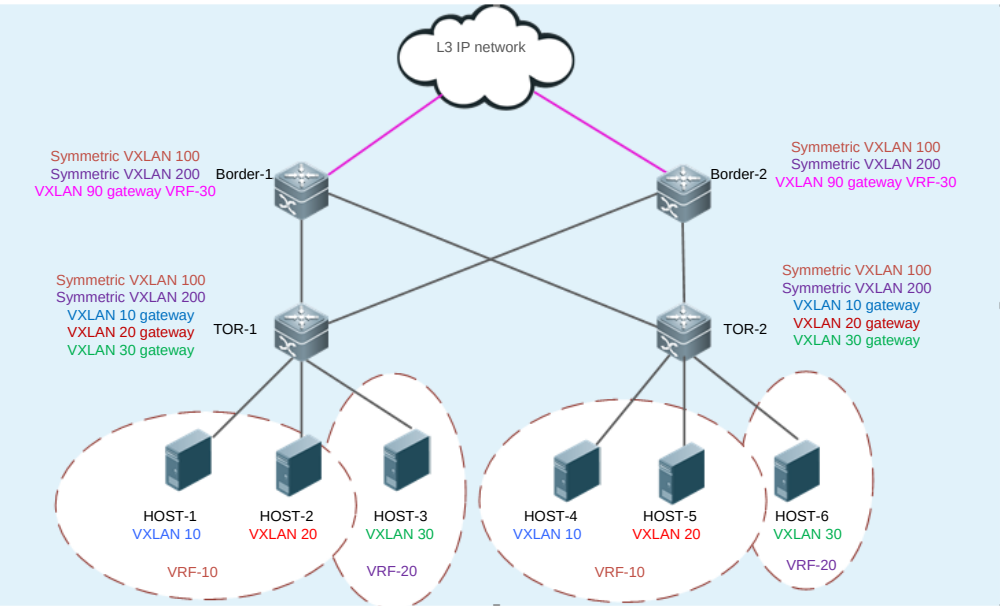

```
TOR2(config)# int overlayrouter 30
TOR2(config-if-OverlayRouter 30)# ip vrf forwarding vrf-20
TOR2(config-if-OverlayRouter 30)# ip address 30.1.1.1/24
TOR2(config-if-OverlayRouter 30)# anycast-gateway
TOR2(config-if-OverlayRouter 30)# exit
TOR2(config)# vxlan 30
TOR2(config-vxlan)# router-interface OverlayRouter 30
TOR2(config)# int overlayrouter 200
TOR2(config-if-OverlayRouter 200)# ip vrf forwarding vrf-20
TOR2(config-if-OverlayRouter 200)# ip address 200.1.3.1/24
TOR2(config-if-OverlayRouter 200)# exit
TOR2(config)# vxlan 10
TOR2(config-vxlan)# extend-vlan 10
TOR2(config-vxlan)# router-interface OverlayRouter 10
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# vxlan 20
TOR2(config-vxlan)# extend-vlan 20
TOR2(config-vxlan)# router-interface OverlayRouter 20
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# vxlan 100
TOR2(config-vxlan)# symmetric
```

Scenario
Figure 1-28



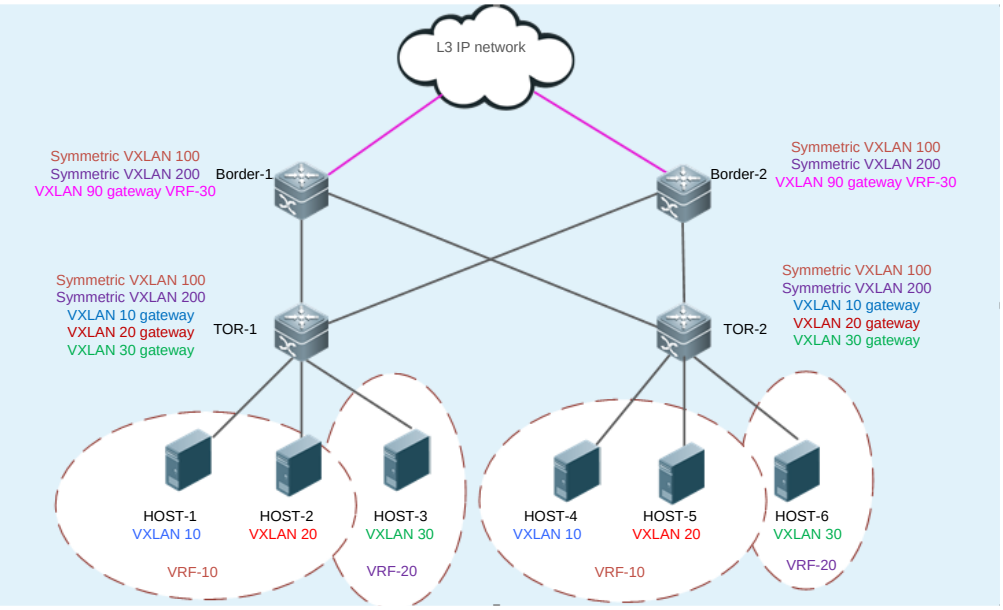

```
TOR2(config-vxlan)# router-interface OverlayRouter 100
TOR2(config-vxlan)# exit
TOR2(config)# vxlan 30
TOR2(config-vxlan)# extend-vlan 30
TOR2(config-vxlan)# router-interface OverlayRouter 30
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# vxlan 200
TOR2(config-vxlan)# symmetric
TOR2(config-vxlan)# router-interface OverlayRouter 200
TOR2(config-vxlan)# exit
TOR2(config)# router bgp 64512
TOR2(config-router)# neighbor 1.1.1.1 remote-as 64512
TOR2(config-router)# neighbor 1.1.1.1 update-source loopback 1
TOR2(config-router)# neighbor 3.3.3.3 remote-as 64512
TOR2(config-router)# neighbor 3.3.3.3 update-source loopback 1
TOR2(config-router)# neighbor 4.4.4.4 remote-as 64512
TOR2(config-router)# neighbor 4.4.4.4 update-source loopback 1
TOR2(config-router)# address-family l2vpn evpn
TOR2(config-router-af)# neighbor 1.1.1.1 activate
TOR2(config-router-af)# neighbor 3.3.3.3 activate
TOR2(config-router-af)# neighbor 4.4.4.4 activate
TOR2(config-router-af)# advertise ipv4 unicast
```

Scenario
Figure 1-28



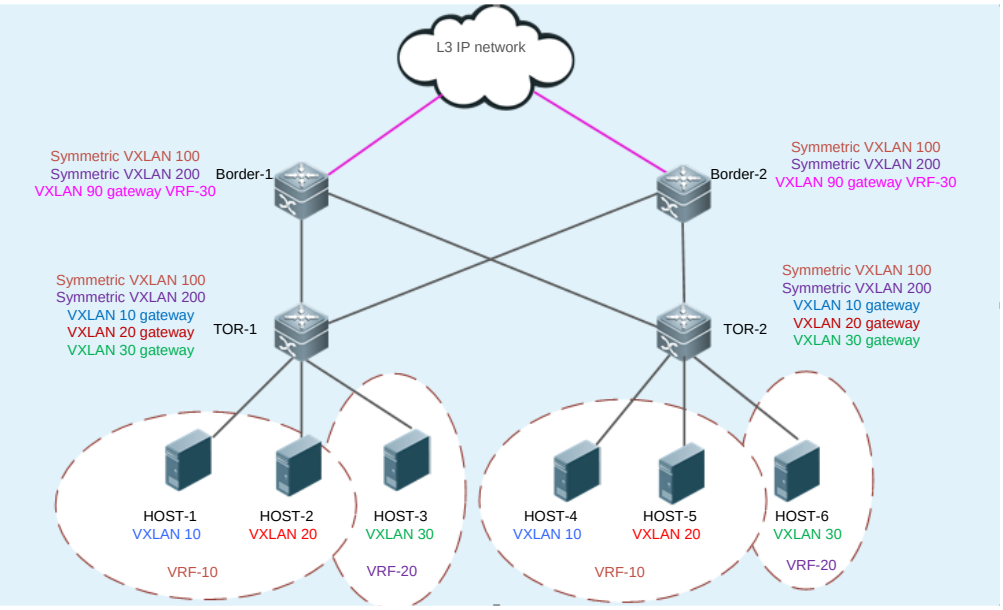

```
TOR2(config-router-af)# exit
TOR2(config-router)# address-family ipv4 vrf vrf-10
TOR2(config-router-af)# redistribute connected
TOR2(config-router-af)# exit
TOR2(config-router)# address-family ipv4 vrf vrf-20
TOR2(config-router-af)# redistribute connected
TOR2(config-router-af)# exit
TOR2(config-router)# exit
TOR2(config)# evpn
TOR2(config-evpn)# vni 10
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 20
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 30
TOR2(config-evpn-vni)# rd auto
TOR2(config-evpn-vni)# route-target both auto
TOR2(config-evpn-vni)# exit
TOR2(config-evpn)# vni 100
TOR2(config-evpn-vni)# rd auto
```

Scenario
Figure 1-28



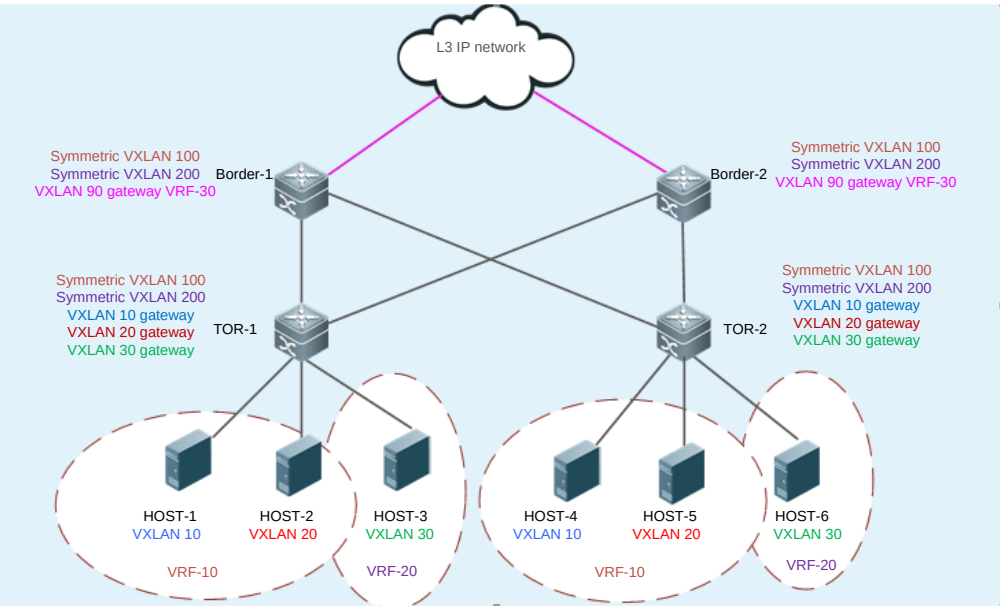
	TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# route-target import 1000:1000 TOR2(config-evpn-vni)# exit TOR2(config-evpn)# vni 200 TOR2(config-evpn-vni)# rd auto TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# route-target import 2000:2000 TOR2(config-evpn-vni)# exit
Border1	Border1# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Border1(config)# interface Loopback 1 Border1(config-if- Loopback 1)# ip address 3.3.3.3/32 Border1(config-if- Loopback 1)# exit Border1(config)# vtep Border1(config-vtep)# source loopback 1 Border1(config-vtep)# arp suppress enable Border1(config-vtep)# exit Border1(config)# fabric anycast-gateway-mac 0011.2233.2016 Border1(config)# ip vrf vrf-10 Border1(config-vrf)# rd 10:10 Border1(config-vrf)# route-target export 1000:1000 Border1(config-vrf)# route-target import 3000:3000

Scenario
Figure 1-28



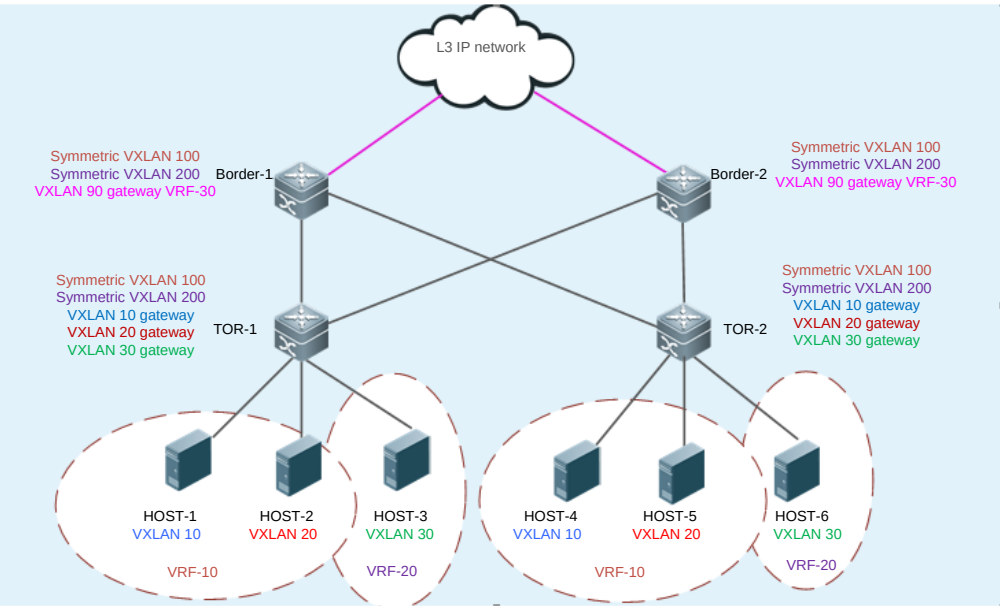

```
Border1(config-vrf)# exit
Border1(config)# ip vrf vrf-20
Border1(config-vrf)# rd 20:20
Border1(config-vrf)# route-target export 2000:2000
Border1(config-vrf)# route-target import 3000:3000
Border1(config-vrf)# exit
Border1(config)# ip vrf vrf-30
Border1(config-vrf)# rd 30:30
Border1(config-vrf)# route-target export 3000:3000
Border1(config-vrf)# route-target import 1000:1000
Border1(config-vrf)# route-target import 2000:2000
Border1(config-vrf)# exit
Border1(config)# int overlayrouter 90
Border1(config-if-OverlayRouter 90)# ip vrf forwarding vrf-30
Border1(config-if-OverlayRouter 90)# ip address 90.1.1.1/24
Border1(config-if-OverlayRouter 90)# anycast-gateway
Border1(config-if-OverlayRouter 90)# exit
Border1(config)# int overlayrouter 100
Border1(config-if-OverlayRouter 100)# ip vrf forwarding vrf-10
Border1(config-if-OverlayRouter 100)# ip address 100.1.1.1/24
Border1(config-if-OverlayRouter 100)# exit
Border1(config)# int overlayrouter 200
Border1(config-if-OverlayRouter 200)# ip vrf forwarding vrf-20
```

Scenario
Figure 1-28



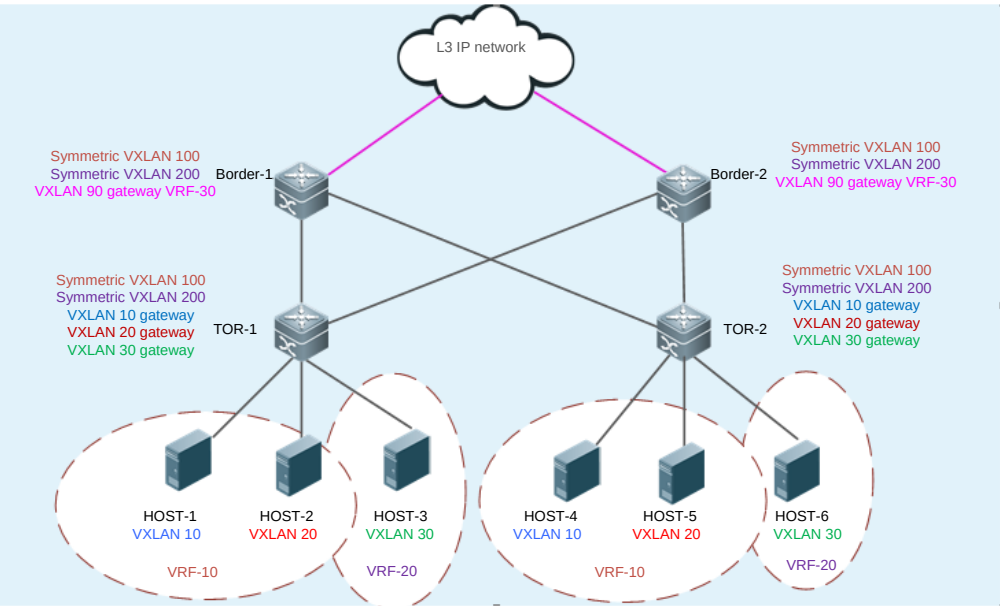

```
Border1(config-if-OverlayRouter 200)# ip address 200.1.1.1/24
Border1(config-if-OverlayRouter 200)# exit
Border1(config)# vxlan 90
Border1(config-vxlan)# extend-vlan 90
Border1(config-vxlan)# router-interface OverlayRouter 90
Border1(config-vxlan)# arp suppress enable
Border1(config-vxlan)# exit
Border1(config)# vxlan 100
Border1(config-vxlan)# symmetric
Border1(config-vxlan)# router-interface OverlayRouter 100
Border1(config-vxlan)# exit
Border1(config)# vxlan 200
Border1(config-vxlan)# symmetric
Border1(config-vxlan)# router-interface OverlayRouter 200
Border1(config-vxlan)# exit
Border1(config)# router bgp 64512
Border1(config-router)# neighbor 1.1.1.1 remote-as 64512
Border1(config-router)# neighbor 1.1.1.1 update-source loopback 1
Border1(config-router)# neighbor 2.2.2.2 remote-as 64512
Border1(config-router)# neighbor 2.2.2.2 update-source loopback 1
Border1(config-router)# neighbor 4.4.4.4 remote-as 64512
Border1(config-router)# neighbor 4.4.4.4 update-source loopback 1
Border1(config-router)# address-family l2vpn evpn
```

Scenario
Figure 1-28



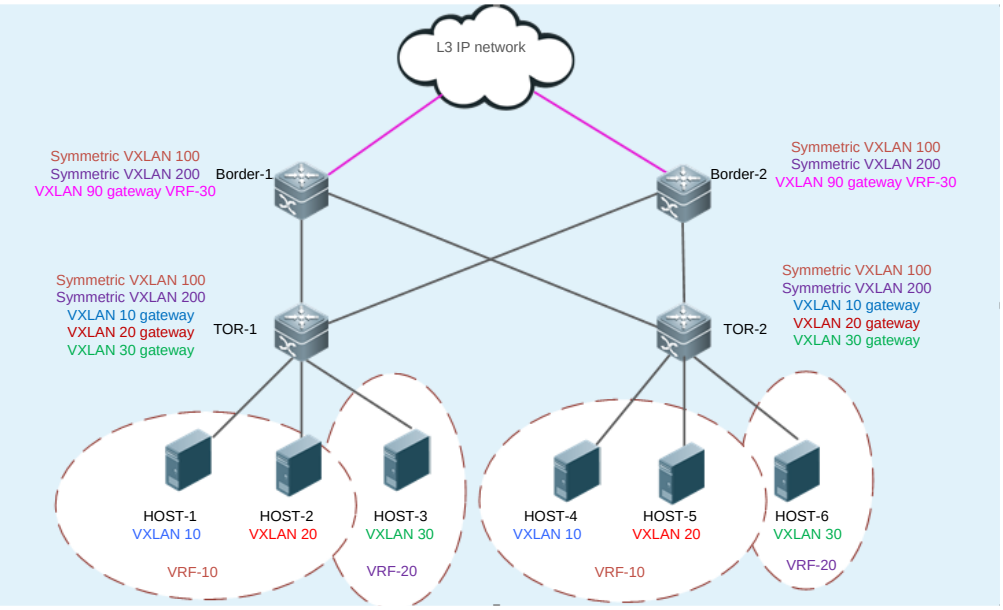

```
Border1(config-router-af)# neighbor 1.1.1.1 activate
Border1(config-router-af)# neighbor 2.2.2.2 activate
Border1(config-router-af)# neighbor 4.4.4.4 activate
Border1(config-router-af)# advertise ipv4 unicast
Border1(config-router-af)# exit
Border1(config-router)# address-family ipv4 vrf vrf-10
Border1(config-router-af)# exit
Border1(config-router)# address-family ipv4 vrf vrf-20
Border1(config-router-af)# exit
Border1(config-router)# address-family ipv4 vrf vrf-30
Border1(config-router-af)# redistribute static
Border1(config-router-af)# exit
Border1(config-router)# exit
Border1(config)# evpn
Border1(config-evpn)# vni 100
Border1(config-evpn-vni)# rd auto
Border1(config-evpn-vni)# route-target both auto
Border1(config-evpn-vni)# route-target import 3000:3000
Border1(config-evpn-vni)# exit
Border1(config-evpn)# vni 200
Border1(config-evpn-vni)# rd auto
Border1(config-evpn-vni)# route-target both auto
Border1(config-evpn-vni)# route-target import 3000:3000
```

Scenario
Figure 1-28



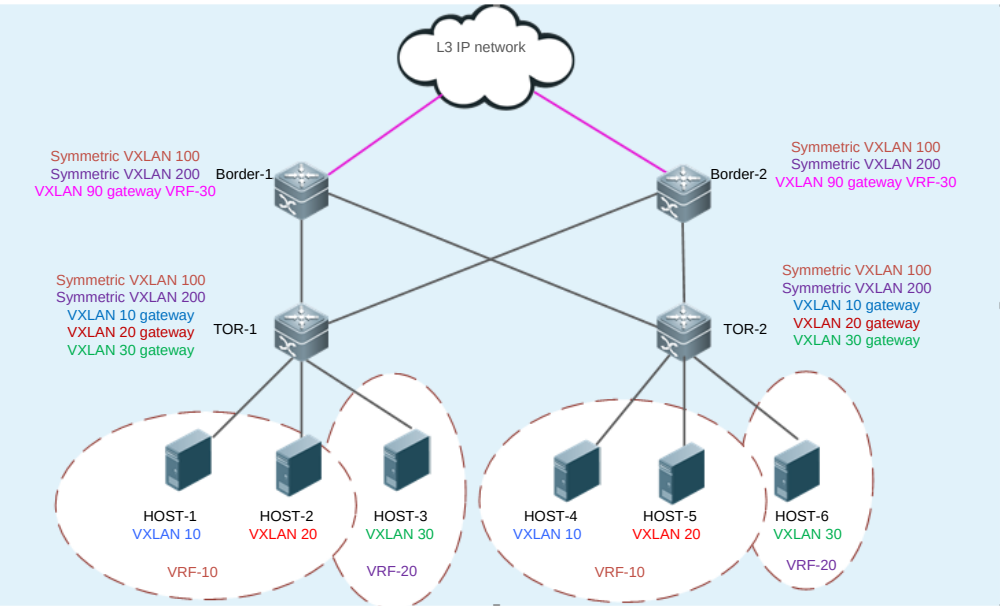
	Border1(config-evpn-vni)# exit
Border2	Border2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Border2(config)# interface Loopback 1 Border2(config-if- Loopback 1)# ip address 4.4.4.4/32 Border2(config-if- Loopback 1)# exit Border2(config)# vtep Border2(config-vtep)# source loopback 1 Border2(config-vtep)# arp suppress enable Border2(config-vtep)# exit Border2(config)# fabric anycast-gateway-mac 0011.2233.2016 Border2(config)# ip vrf vrf-10 Border2(config-vrf)# rd 10:10 Border2(config-vrf)# route-target export 1000:1000 Border2(config-vrf)# route-target import 3000:3000 Border2(config-vrf)# exit Border2(config)# ip vrf vrf-20 Border2(config-vrf)# rd 20:20 Border2(config-vrf)# route-target export 2000:2000 Border2(config-vrf)# route-target import 3000:3000 Border2(config-vrf)# exit Border2(config)# ip vrf vrf-30

Scenario
Figure 1-28



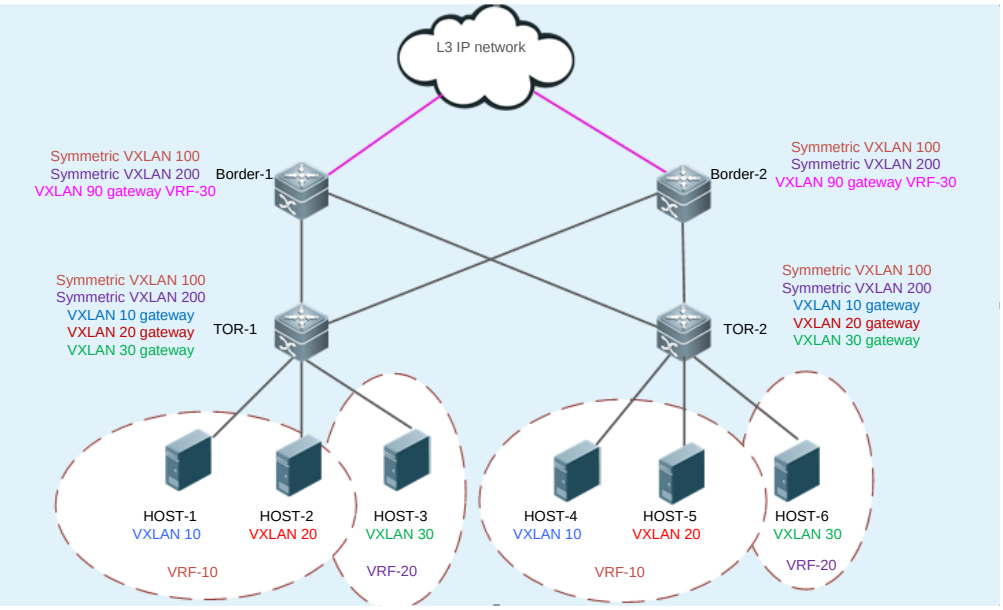

```
Border2(config-vrf)# rd 30:30
Border2(config-vrf)# route-target export 3000:3000
Border2(config-vrf)# route-target import 1000:1000
Border2(config-vrf)# route-target import 2000:2000
Border2(config-vrf)# exit
Border2(config)# int overlayrouter 90
Border2(config-if-OverlayRouter 90)# ip vrf forwarding vrf-30
Border2(config-if-OverlayRouter 90)# ip address 90.1.2.1/24
Border2(config-if-OverlayRouter 90)# anycast-gateway
Border2(config-if-OverlayRouter 90)# exit
Border2(config)# int overlayrouter 100
Border2(config-if-OverlayRouter 100)# ip vrf forwarding vrf-10
Border2(config-if-OverlayRouter 100)# ip address 100.1.2.1/24
Border2(config-if-OverlayRouter 100)# exit
Border2(config)# int overlayrouter 200
Border2(config-if-OverlayRouter 200)# ip vrf forwarding vrf-20
Border2(config-if-OverlayRouter 200)# ip address 200.1.2.1/24
Border2(config-if-OverlayRouter 200)# exit
Border2(config)# vxlan 90
Border2(config-vxlan)# extend-vlan 90
Border2(config-vxlan)# router-interface OverlayRouter 90
Border2(config-vxlan)# arp suppress enable
Border2(config-vxlan)# exit
```

Scenario
Figure 1-28




```
Border2(config)# vxlan 100
Border2(config-vxlan)# symmetric
Border2(config-vxlan)# router-interface OverlayRouter 100
Border2(config-vxlan)# exit
Border2(config)# vxlan 200
Border2(config-vxlan)# symmetric
Border2(config-vxlan)# router-interface OverlayRouter 200
Border2(config-vxlan)# exit
Border2(config)# router bgp 64512
Border2(config-router)# neighbor 1.1.1.1 remote-as 64512
Border2(config-router)# neighbor 1.1.1.1 update-source loopback 1
Border2(config-router)# neighbor 2.2.2.2 remote-as 64512
Border2(config-router)# neighbor 2.2.2.2 update-source loopback 1
Border2(config-router)# neighbor 3.3.3.3 remote-as 64512
Border2(config-router)# neighbor 3.3.3.3 update-source loopback 1
Border2(config-router)# address-family l2vpn evpn
Border2(config-router-af)# neighbor 1.1.1.1 activate
Border2(config-router-af)# neighbor 2.2.2.2 activate
Border2(config-router-af)# neighbor 3.3.3.3 activate
Border2(config-router-af)# advertise ipv4 unicast
Border2(config-router-af)# exit
Border2(config-router)# address-family ipv4 vrf vrf-10
Border2(config-router-af)# exit
```

Scenario
Figure 1-28

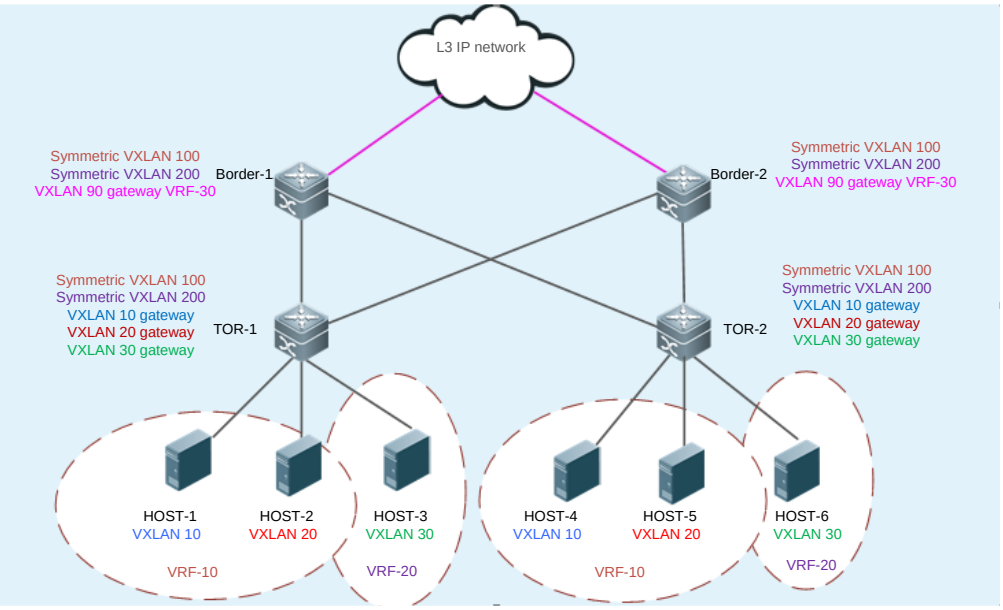



```
Border2(config-router)# address-family ipv4 vrf vrf-20
Border2(config-router-af)# exit
Border2(config-router)# address-family ipv4 vrf vrf-30
Border2(config-router-af)# redistribute static
Border2(config-router-af)# exit
Border2(config-router)# exit
Border2(config)# evpn
Border2(config-evpn)# vni 100
Border2(config-evpn-vni)# rd auto
Border2(config-evpn-vni)# route-target both auto
Border2(config-evpn-vni)# route-target import 3000:3000
Border2(config-evpn-vni)# exit
Border2(config-evpn)# vni 200
Border2(config-evpn-vni)# rd auto
Border2(config-evpn-vni)# route-target both auto
Border2(config-evpn-vni)# route-target import 3000:3000
Border2(config-evpn-vni)# exit
```

Verification

- Verify that HOST-1, HOST-2, and HOST-4 can ping each other.
- Verify that HOST-3 and HOST-6 can ping each other.
- Verify that HOST-1, HOST-2, and HOST-4 cannot ping HOST-3 and HOST-6.
- Verify that the virtual machines can be migrated between the HOSTs on the same VXLAN and can access the network normally

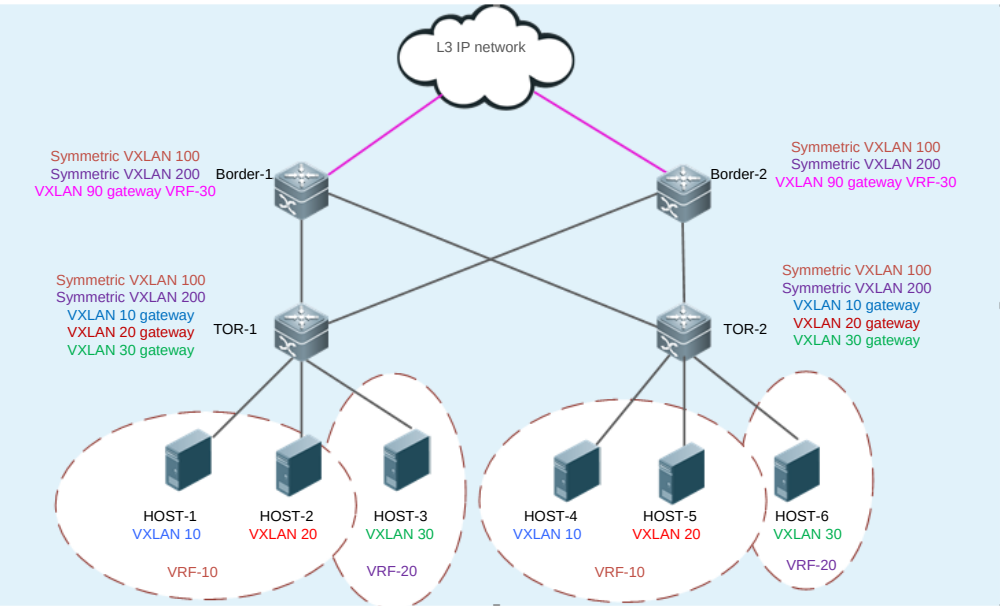
Scenario
Figure 1-28



	after migration without modifying the configuration.

```
Border1# sh vxlan
VXLAN Total Count: 3
VXLAN Capacity   : 8000

VXLAN 90
   Symmetric property  : FALSE
   Router Interface    : overlayrouter 90 (anycast)
   Extend VLAN         : 90
VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- -------------- --------------- -------
   OverlayTunnel 6146   3.3.3.3        2.2.2.2         dynamic

VXLAN 100
   Symmetric property  : TRUE
   Router Interface    : overlayrouter 100 (non-anycast)
   Extend VLAN         : -
VTEP Adjacency Count: 1
```

Scenario
Figure 1-28




```
        Interface            Source IP       Destination IP  Type
  ---------------------- --------------- --------------- -------
  OverlayTunnel 6146     3.3.3.3         2.2.2.2         dynamic


VXLAN 200
  Symmetric property  : TRUE
  Router Interface    : overlayrouter 200 (non-anycast)
  Extend VLAN         : -
  VTEP Adjacency Count: 1
  VTEP Adjacency List :
  Interface            Source IP       Destination IP  Type
  ---------------------- --------------- --------------- -------
  OverlayTunnel 6146     3.3.3.3         2.2.2.2         dynamic
```

1.4.3. Configuring an EVPN Distributed Network to Be Compatible with Non-EVPN VTEP Devices

Configuration Effect

- Enable the control plane learning function to implement VXLAN tunnel learning, MAC address learning, and route learning via control plane protocols, thereby finally implementing VXLAN bridging, VXLAN routing, and data communication between VXLANs and between a VXLAN and an external network.
- Support functions such as anycast gateways, symmetric VXLAN instances, ARP suppression, and IPv6 ND suppression in EVPN control plane mode.
- Non-EVPN VTEP devices establish tunnels with EVPN-supported VTEP devices to implement the VXLAN bridging and forwarding function.

Notes

- VXLAN instances require the support from existing unicast routes on the network. Therefore, an IPv4 unicast routing protocol such as OSPF must be configured on the network devices.
- VXLAN needs the MP-BGP-EVPN protocol to implement VXLAN tunnel learning, MAC address learning, and route learning. Therefore, BGP must be configured on EVPN-supported network devices.

Configuration Steps

Configuring a Loopback Interface Associated with the Local End

- Mandatory for devices supporting the EVPN address family.
- Configure the IP address of a loopback interface as the IP address of the local VTEP. One device can be associated with only one loopback interface and the IP address of the loopback interface serves as the VXLAN VTEP IP address.
- If the L3 egress is an overlay router interface during static route configuration, the next-hop IP address cannot be set to the VTEP IP address.

Configuring a Virtual MAC Address for Anycast Gateways

- Optional.
- When anycast gateways are required, a unified virtual MAC address must be configured and used as the MAC address of the anycast gateways. The anycast function can be enabled on the VXLAN overlay router interfaces of the local device only after the virtual MAC address is configured.

Configuring ARP Suppression

- Optional.
- After ARP suppression is enabled, the VTEP device responds to ARP requests from hosts as a proxy, to reduce flooded ARP data on the network.
- ARP suppression is generally enabled on distributed gateways in distributed deployment scenarios.

Configuring ARP Proxy

- Optional.
- After ARP suppression is enabled on a VTEP device, you can enable the ARP proxy function on an overlay router interface.
- After ARP proxy is enabled, the VTEP device responds to ARP requests from hosts as a proxy and the MAC address used for proxy response is the gateway MAC address configured on the VTEP device. In this way, the MAC address in the ARP request responses are the MAC address of the VTEP device, and the traffic between hosts in the same VNI is forwarded at L3.
- ARP proxy can be enabled only on VXLAN gateways and is generally enabled on distributed gateways in distributed deployment scenarios.

Configuring IPv6 ND Suppression

- Optional.
- After IPv6 ND suppression is enabled, the VTEP device responds to NS multicast packets from hosts as a proxy, to reduce flooded NS multicast packets on the network.

- IPv6 ND suppression is generally enabled on distributed gateways in distributed deployment scenarios.

Configuring the EVPN Protocol Packet Control Function

- In symmetric EVPN deployment scenarios, the EVPN protocol packet control function can be configured on TOR switches to reduce the traffic of EVPN packets.
- Currently, the EVPN protocol packet control function includes the following:
- Extracting MAC entries from EVPN MAC-IP type-2 routes (ARP entries) on a L2-VPN VXLAN instance
- Extracting MAC entries from EVPN MAC-IPv6 type-2 routes (IPv6 ND entries) on a L2-VNI VXLAN instance
- Banning synchronization of the local MAC address to the remote VTEP through EVPN messages on an L2-VNI VXLAN instance
- Banning delivery of the MAC addresses remotely synchronized through EVPN messages to the local MAC address table on an L2-VNI VXLAN instance
- Stopping an L2-VNI VXLAN instance from generating EVPN type-2 routes

Creating a VXLAN Instance

- Mandatory.

Associating the VXLAN Instance with an Overlay Router Interface

- Mandatory for VXLAN gateways.
- The device supports the VXLAN routing function and can serve as a VXLAN IP gateway only after the VXLAN is associated with an overlay router interface.

Associating the VXLAN Instance with a VLAN

- Mandatory for VXLAN devices directly connected to hosts.
- Packets of a VLAN are encapsulated into VXLAN packets for forwarding only after the VLAN is associated with a VXLAN instance.
- After a VLAN is associated with a VXLAN, all packets of the VLAN will be encapsulated into VXLAN packets. Therefore, an SVI on the device cannot be used as the IP gateway of the VLAN.

Configuring Overlay Tunnels

- Mandatory for a VTEP not supporting EVPN and devices that directly communicate with the VTEP.
- The type of tunnels mutually established by two VTEP devices must be the same. The tunnels are those delivered by the SDN controller, statically configured on the CLI, or auto-discovered by EVPN.

Configuring the Source and Destination IP Addresses for Overlay Tunnels

- Mandatory for a VTEP not supporting EVPN and devices that directly communicate with the VTEP.

Associating the VXLAN Instance with the Overlay Tunnels

- Mandatory for a VTEP not supporting EVPN and devices that directly communicate with the VTEP.
- This command is used to statically specify VXLAN tunnels.

Configuring Storm Control for the VXLAN Instance

- Optional.
- This function is required only when the storm rate needs to be limited for a VXLAN instance.

Configuring the VXLAN UDP Destination Port

- Optional. The VXLAN UDP destination port used by early devices may not be port 4789. You can run this command to achieve compatibility. In addition, you can also run this command to specify the VXLAN UDP destination port.
- The VXLAN UDP destination port 4789 designated by IANA is used by default.

Configuring Symmetric Instances

- Optional.
- Symmetric instances need to be configured only in symmetric scenarios. Only one symmetric instance can be configured in each VRF instance. After a symmetric instance is configured in a VRF instance, L3 forwarding in other asymmetric instances is switched to the symmetric instance.

Configuring VXLAN Static Routes

- Optional.
- Configure VXLAN static routes for VXLAN instances if required.

Configuring the Synchronization of MAC Entries Whose Egresses Are Static Tunnels

- Optional.
- Configure this function when MAC entries with the egress of static tunnels need to be synchronized externally.

Configuring the Synchronization of ARP Entries Whose Egresses Are Static Tunnels

- Optional.
- Configure this function when ARP entries with the egress of static tunnels need to be synchronized externally.

Verification

The device can establish VXLAN tunnels and obtain VXLAN MAC entries and VXLAN routing entries via EVPN control plane learning. The tunnels and entries can be those delivered by the SDN controller or statically configured on the CLI. They implement the inter-VTEP communication. Run the following commands for verification.

- Run the **show vxlan** *vni-number* command to check whether the local and remote VXLAN devices learn the peer VTEP neighbor relationships.
- Run the **show vxlan mac** to check whether the VXLAN MAC addresses are learned.
- Run the **show arp** command to check whether the ARP entry of the VXLAN IP gateway is learned.
- Run the **show ipv6 neighbors** command to check whether all local/remote IPv6 ND entries are learned.
- Run the **show vxlan udp-port** command to display the VXLAN UDP destination port.

Related Commands

Configuring a Loopback Interface Associated with the Local End

Command	**source loopback** *loopback-port-id*
Parameter Description	*loopback-port-id: Indicates the ID of the loopback interface.*
Command Mode	VTEP configuration mode
Usage Guide	The local VETP IP address is the IP address of the configured loopback interface.

Configuring a Virtual MAC Address for Anycast Gateways

Command	**fabric anycast-gateway-mac** *mac-addr*
Parameter Description	*mac-addr: Indicates the MAC address in the format of xxxx.xxxx.xxxx.*
Command Mode	Global configuration mode
Usage Guide	All gateways on which the anycast function is enabled use this MAC address as the gateway MAC address. The virtual MAC address for anycast gateways cannot be set to the local MAC address or the MAC address of any device on the overlay network.

Configuring Remote ARP Packet Learning

Command	**remote arp learn enable**
Parameter Description	*N/A*
Command Mode	VTEP configuration mode
Usage Guide	Enable or disable the remote ARP packet learning function globally. After this function is enabled, the VXLAN gateways will learn ARP entries from the VXLAN-encapsulated ARP packets received from VXLAN tunnels.

Configuring Remote IPv6 ND Protocol Packet Learning

Command	**remote nd learn enable**
Parameter Description	*N/A*
Command Mode	VTEP configuration mode
Usage Guide	Enable or disable the remote IPv6 ND packet learning function globally. After this function is enabled, the device can learn IPv6 ND entries from the VXLAN-encapsulated IPv6 NS packets received from VXLAN tunnels.

Configuring Global ARP Suppression

Command	**arp suppress enable**
Parameter Description	*N/A*
Command Mode	VTEP configuration mode
Usage Guide	Enable or disable the global ARP suppression function. After ARP suppression is enabled, the switch responds to ARP requests from hosts as a proxy. The VNI-based ARP suppression may be also supported, depending on the product type. You can configure global ARP suppression or VNI-based ARP suppression based on the actual application scenario.

Configuring VNI-based ARP Suppression	
Command	**arp suppress enable**
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	Enable or disable VNI-based ARP suppression. After ARP suppression is enabled, the switch responds to ARP requests from hosts as a proxy. The global ARP suppression may be also supported, depending on the product type. You can configure global ARP suppression or VNI-based ARP suppression based on the actual application scenario.

Configuring ARP Proxy	
Command	**route-in-vni**
Parameter Description	*N/A*
Command Mode	Overlay router interface configuration mode
Usage Guide	After the intra-VNI routing function (ARP proxy) is enabled on an overlay router interface, the VTEP device uses its gateway MAC address to respond to all ARP requests from hosts in the VNI, to which the overlay router interface belongs, when serving as an ARP proxy. In this way, the communication traffic between hosts in the same VNI is forwarded through VXLAN routes.

Configuring Global IPv6 ND Suppression	
Command	**nd suppress enable**
Parameter Description	*N/A*
Command Mode	VTEP configuration mode
Usage Guide	Enable or disable the global IPv6 ND suppression function. After IPv6 ND suppression is enabled, the device responds to IPv6 NS multicast packets from hosts as a proxy. The VNI-based IPv6 ND suppression may be also supported, depending on the product type. You can configure global IPv6 ND suppression or VNI-based IPv6 ND suppression based on the actual application scenario.

Configuring VNI-based IPv6 ND Suppression

Command	nd suppress enable
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	Enable or disable the VNI-based IPv6 ND suppression function. After IPv6 ND suppression is enabled, the device responds to IPv6 NS multicast packets from hosts as a proxy. The global IPv6 ND suppression may be also supported, depending on the product type. You can configure global IPv6 ND suppression or VNI-based IPv6 ND suppression based on the actual application scenario.

Extracting MAC Entries from EVPN MAC-IP Type-2 Routes (ARP Entries)

Command	evpn arp mac-learning enable
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	After this command is configured, the device parses one ARP entry and one MAC entry from a MAC-IP type-2 route synchronized from the VXLAN-EVPN neighbor. This command is disabled by default and the device parses one ARP entry but no MAC entry from a MAC-IP type-2 route synchronized from the VXLAN-EVPN neighbor. This command is configured on a VXLAN instance and affects only the EVPN entry parsing of the VXLAN instance. Other VXLAN instances, for which this command is not configured, are not affected. This command can be used in combination with the **evpn mac advertise disable** command. After they are executed, the network-wide VXLAN-EVPN neighbors synchronize only MAC-IP type-2 routes but no MAC-only type-2 routes. All devices parse and extract MAC entries from MAC-IP type-2 routes. In symmetric deployment scenarios, this command is configured on L3-VNI VXLAN instances (that is, symmetric instances).

Extracting MAC Entries from EVPN MAC-IPv6 Type-2 Routes (IPv6 ND Entries)

Command	**evpn nd mac-learning enable**
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	After this command is configured, the device parses one IPv6 ND entry and one MAC entry from a MAC-IPv6 type-2 route (IPv6 ND entry) synchronized from the VXLAN-EVPN neighbor. This command is disabled by default and the device parses one IPv6 ND entry but no MAC entry from a MAC-IPv6 type-2 route synchronized from the VXLAN-EVPN neighbor. This command is configured on a VXLAN instance and affects only the EVPN entry parsing of the VXLAN instance. Other VXLAN instances, for which this command is not configured, are not affected. This command can be used in combination with the **evpn mac advertise disable** command. After they are executed, the network-wide VXLAN-EVPN neighbors synchronize only MAC-IPv6 type-2 routes but no MAC-only type-2 routes. All devices parse and extract MAC entries from MAC-IPv6 type-2 routes. This command is configured on L2-VNI VXLAN instances.

Configuring an L2-VNI VXLAN Instance Not to Synchronize the Local MAC Address to the Remote VTEP Through EVPN Messages

Command	**evpn mac advertise disable**
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	This command is not configured on a device by default. The device generates one MAC-only type-2 route through the VXLAN-EVPN protocol based on a locally learned MAC entry, and synchronizes the type-2 route to the EVPN neighbor (that is, remote VTEP). Then, the remote VTEP can learn the MAC entry from the MAC-only type-2 route. After this command is configured, the device does not generate VXLAN-EVPN MAC-only type-2 routes based on MAC entries, and therefore, it will not advertise MAC-only type-2 routes to the EVPN neighbor. This command is configured on a VXLAN instance and affects only whether the VXLAN instance generates MAC-only type-2 routes. Other VXLAN instances, for which this command is not configured, can still generate MAC-only type-2 routes. This command can be used in combination with the **evpn arp mac-learning enable** and **evpn nd mac-learning enable** commands. After they are executed, the network-wide VXLAN-EVPN neighbors synchronize only MAC-IP type-2 routes but no MAC-only type-2 routes. All devices parse and extract MAC entries from MAC-IP or MAC-IPv6 type-2 routes. Note: This command can be configured only on L2-VNI VXLAN instances

Command	evpn mac advertise disable
	(that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

Configuring an L2-VNI VXLAN Instance Not to Deliver MAC Addresses Remotely Synchronized Through EVPN Messages to the Local MAC Address Table

Command	evpn mac inactive
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	After this command is configured, the device does not learn MAC entries from VXLAN-EVPN type-2 routes (MAC-IP or MAC-only type-2 routes) synchronized from neighbors. This command is not configured on a device by default. The device learns MAC entries from VXLAN-EVPN type-2 routes synchronized from neighbors. This command is configured on a VXLAN instance and affects only whether the VXLAN instance learns MAC entries from VXLAN-EVPN type-2 routes. Other VXLAN instances, for which this command is not configured, can still learn MAC entries. Note: This command can be configured only on L2-VNI VXLAN instances (that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

Configuring an L2-VNI VXLAN Instance Not to Generate EVPN Type-2 Routes

Command	evpn rt-2 advertise disable
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	This command is not configured on a device by default. The device generates one MAC-only type-2 route through the VXLAN-EVPN protocol based on a locally learned MAC entry, and synchronizes the type-2 route to the EVPN neighbor (that is, remote VTEP). Then, the remote VTEP learns the MAC entry from the MAC-only type-2 route. In addition, the device generates one MAC-IP type-2 route through the VXLAN-EVPN protocol based on a locally learned ARP entry and synchronizes the type-2 route to the EVPN neighbor. Then, the remote VTEP learns the ARP entry and host route from the MAC-IP type-2 route. The device generates one MAC-IPv6 type-2 route through the VXLAN-EVPN protocol based on a locally learned IPv6 ND entry, and synchronizes the type-2 route to the EVPN neighbor. Then, the remote VTEP learns the IPv6 ND entry and host route from the MAC-IPv6 type-2 route. After this command is configured, the MAC entries, ARP entries, and IPv6 ND entries of the device are not used to generate VXLAN-EVPN type-2 routes and therefore, no type-2 route is advertised to the EVPN neighbor. This command is configured on a VXLAN instance and affects only whether

Command	**evpn rt-2 advertise disable**
	the VXLAN instance generates type-2 routes. Other VXLAN instances, for which this command is not configured, can still generate type-2 routes. Note: This command can be configured only on L2-VNI VXLAN instances (that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

Creating an Overlay Router Interface

Command	**interface OverlayRouter** *port-id*
Parameter Description	*port-id*: Indicates the ID of an overlay router interface.
Command Mode	Global configuration mode
Usage Guide	This interface serves as the VXLAN IP gateway in the VXLAN routing environment. It is similar to an SVI interface in a VLAN.

Configuring an IP Address for the Overlay Router Interface

Command	**ip address** *ip-address mask*
Parameter Description	*ip-address*: Indicates the IP address of the overlay router interface. *mask*: Indicates the subnet mask.
Command Mode	Interface configuration mode
Usage Guide	This IP address serves as the VXLAN IP gateway address in the VXLAN routing environment. It is similar to the IP address of an SVI in a VLAN.

Configuring an IPv6 Address for the Overlay Router Interface

Command	**ipv6 address** *ip-address mask*
Parameter Description	*ip-address*: Indicates the IPv6 address of the overlay router interface. *mask*: Indicates the subnet mask.
Command Mode	Overlay router interface configuration mode
Usage Guide	This IPv6 address serves as the VXLAN IPv6 gateway address in the VXLAN routing environment. It is similar to the IP address of an SVI in a VLAN.

Associating the Overlay Router Interface with a VRF Instance	
Command	**vrf forwarding** *table name*
Parameter Description	*table name*: Indicates the VRF instance, with which the overlay router interface is associated.
Command Mode	Interface configuration mode
Usage Guide	This command is used to associate with a VRF instance in the VXLAN routing environment and is used for VXLAN L3 routing isolation.

Creating a VXLAN Instance or Entering the VXLAN Configuration Mode	
Command	**vxlan** *vni-number*
Parameter Description	*vni-number*: Indicates the VNI. The value ranges from 1 to 16,777,215.
Command Mode	Global configuration mode
Usage Guide	N/A

Configuring a Symmetric Instance	
Command	**symmetric**
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	No symmetric instance is configured by default. A symmetric instance is used to manage L3 forwarding entries of all asymmetric instances in the VRF instance associated with the symmetric instance.

Associating the VXLAN Instance with the Overlay Router Interface	
Command	**router-interface** *interface-name*
Parameter Description	*interface-name*: Indicates the name of the overlay router interface.
Command Mode	VXLAN configuration mode
Usage Guide	The **overlay router interfaces** between VXLANs cannot conflict with each other and different VXLANs cannot associate with the same overlay router interface.

Creating an Overlay Tunnel Interface	
Command	**interface OverlayTunnel** *port-id*
Parameter Description	*port-id*: Indicates the ID of an overlay tunnel interface.
Command Mode	Global configuration mode
Usage Guide	This interface is used to statically create an overlay tunnel. You can run the **tunnel-interface** command to associate it with a VXLAN.

Configuring a Tunnel Source IP Address for the Overlay Tunnel Interface	
Command	**tunnel source** *ip-address*
Parameter Description	*ip-address*: Indicates the tunnel source IP address.
Command Mode	Overlay tunnel interface configuration mode
Usage Guide	This command is used to specify the source IP address of an overlay tunnel. When packets are encapsulated and forwarded, the outer source IP address of the packets is the source IP address of an overlay tunnel.

Configuring a Tunnel Destination IP Address for the Overlay Tunnel Interface	
Command	**tunnel destination** *ip-address*
Parameter Description	*ip-address*: Indicates the tunnel destination IP address.
Command Mode	Overlay tunnel interface configuration mode
Usage Guide	This command is used to specify the destination IP address of an overlay tunnel. When packets are encapsulated and forwarded, the outer destination IP address of the packets is the destination IP address of an overlay tunnel. The tunnel destination IP address is unique globally. Different overlay tunnels cannot share the same destination IP address. Otherwise, a configuration conflict will occur.

Associating the VXLAN Instance with the Overlay Tunnel Interface	
Command	**tunnel-interface** *interface-name*
Parameter Description	*interface-name*: Indicates the name of an overlay tunnel interface.
Command Mode	VXLAN configuration mode
Usage Guide	This command is used to statically specify a VXLAN VTEP.

Configuring the VXLAN UDP Destination Port

Command	**vxlan udp-port** *port-number*
Parameter Description	*port-number*: Indicates the UDP destination port ID. The value ranges from 0 to 65535 and the default value is 4789.
Command Mode	Global configuration mode
Usage Guide	The VXLAN UDP destination port cannot be set to a commonly used UDP port.

Configuring Storm Control for the VXLAN Instance

Command	**storm-control {broadcast | multicast | unicast} [***kbps-value* **| pps** *pps-value*]
Parameter Description	*kbps-value*: Indicates the rate limit value, in kbit/s. *pps-value*: Indicates the rate limit value, in packets/second.
Command Mode	VXLAN configuration mode
Usage Guide	Configure this function when the storm rate needs to be limited based on the VNI.

Configuring the Synchronization of MAC Entries Whose Egresses Are Static Tunnels

Command	**evpn mac advertise enable**
Parameter Description	N/A
Command Mode	Overlay tunnel interface configuration mode
Usage Guide	Configure this function when MAC entries with the egress of static tunnels need to be synchronized externally.

Configuring the Synchronization of ARP Entries Whose Egresses Are Static Tunnels

Command	**evpn macip advertise enable**
Parameter Description	N/A
Command Mode	Overlay tunnel interface configuration mode
Usage Guide	Configure this function when ARP entries with the egress of static tunnels need to be synchronized externally.

Configuration Example

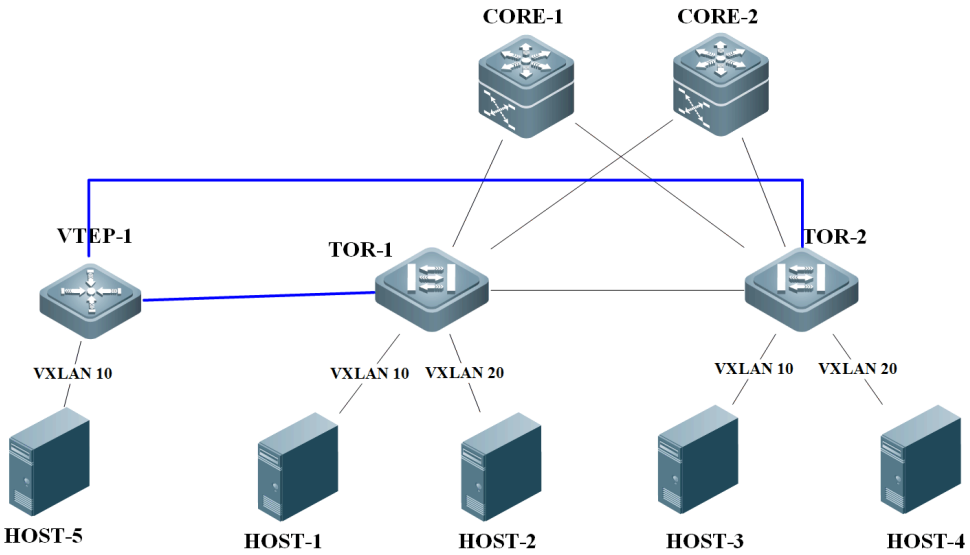
Note:

- Only configuration related to the VXLAN is described below.

- Only IPv4 configuration is used as an example below and the IPv6 scenario configuration is largely the same as the IPv4 scenario configuration.

Detailed configuration of a full mesh network:

Scenario
Figure 1-29

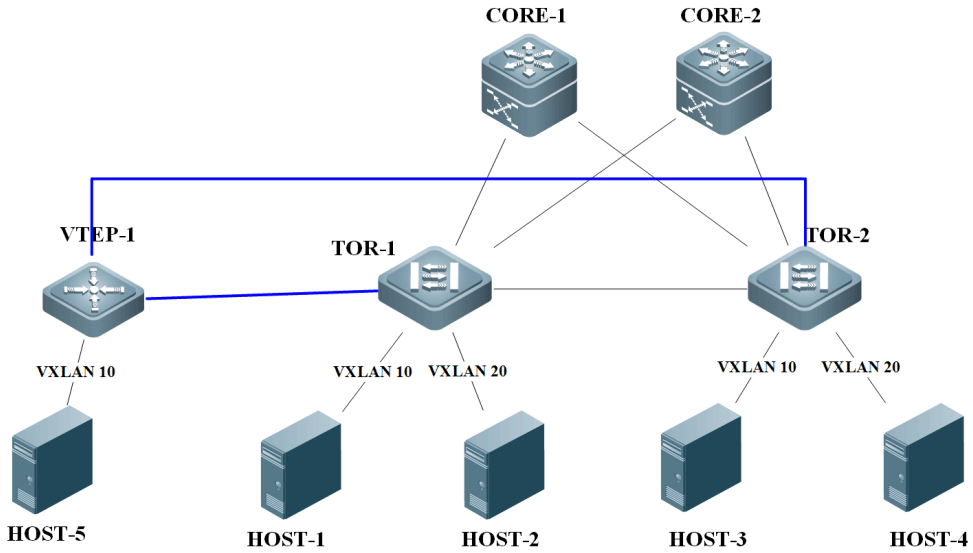


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

Configuration Steps	• Configure an IPv4 unicast routing protocol (such as OSPF) on core switches, TOR switches, and VTEP to ensure that unicast routes are reachable. • Configure the BGP-EVPN routing protocol on the CORE and TOR switches to ensure that the four switches establish BGP neighbor relationships and support the EVPN protocol family. • Configure EVI for BGP-EVPN on the core and TOR switches. For details, see the *BGP-EVPN Configuration Guide*. • Configure VXLANs on the virtual servers and specify the gateway address for the virtual machines. (Omitted) • Associate the VTEP with the loopback interface on TOR-1 and TOR-2 for the establishment of tunnels. • Create VXLAN instances on TOR-1, TOR-2, and VTEP-1 and associate the VXLAN instances with VLANs. • Configure the same anycast gateway MAC address on TOR-1 and TOR-2 so that the VXLAN anycast gateways on the TOR switches use the same virtual MAC address. • Create overlay router interfaces on TOR-1 and TOR-2 and configure the VXLAN gateway IP address. Configure different VRF instances for the overlay router interfaces and determine their respective tenants. • Note that the overlay router interface configuration on TOR-1 and TOR-2 must be the same. That is, on all devices, the IP address and mask configured for the overlay router interfaces associated

Scenario
Figure 1-29

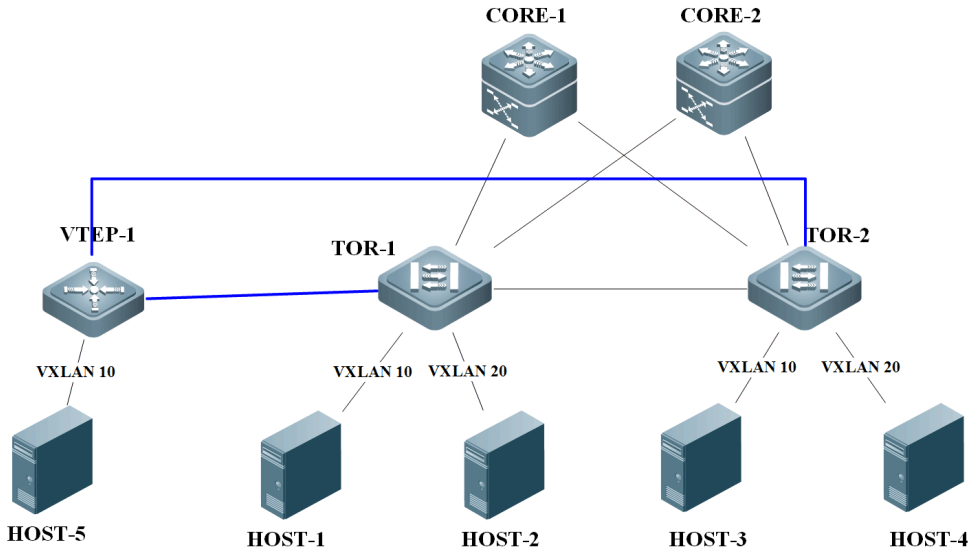


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

	with the same VXLAN instance must be the same and such overlay router interfaces belong to the same tenant (VRF instance). • In addition, all overlay router interfaces must be configured as anycast gateways. • Associate VXLAN instances with overlay router interfaces on TOR-1 and TOR-2 to implement VXLAN routing. • Create VXLAN overlay tunnels on TOR-1, TOR-2, and VTEP-1 and configure the SIP and DIP. • (Optional) Configure ARP suppression on TOR-1 and TOR-2 to reduce ARP packets flowing into the VXLAN.
HOST	The detailed configuration of the servers is omitted here. Configure the IP address and gateway according to the figure above.
CORE	VXLAN does not need to be configured on the core switches. The OSPF and BGP network configurations are omitted here.
TOR1	TOR1# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR1(config)# route-map dc TOR1(config-route-map)# match route-type evpn-type-2 TOR1(config-route-map)# set next-hop 1.1.1.1 TOR1(config-route-map)# exit TOR1(config)# interface Loopback 1 TOR1(config-if- Loopback 1)# ip address 1.1.1.1/32

Scenario
Figure 1-29

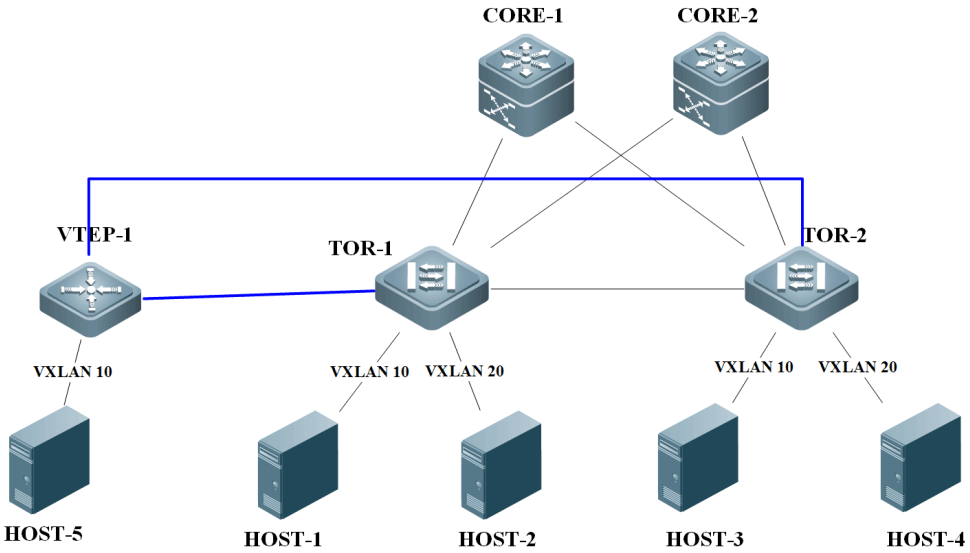


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

```
TOR1(config-if- Loopback 1)# exit
TOR1(config)# vtep
TOR1(config-vtep)# source loopback 1
TOR1(config-vtep)# arp suppress enable
TOR1(config-vtep)# vxlan outside center vtep-ip 3.3.3.3
TOR1(config-vtep)# exit
TOR1(config)# fabric anycast-gateway-mac 0000.1234.5678
TOR1(config)# int overlayrouter 10
TOR1(config-if-OverlayRouter 10)# ip address 10.1.1.1/24
TOR1(config-if-OverlayRouter 10)# anycast-gateway
TOR1(config-if-OverlayRouter 10)# route-in-vni  //Optional. It needs to be used in combination with the arp suppress enable command.
TOR1(config-if-OverlayRouter 10)# exit
TOR1(config)# int overlayrouter 20
TOR1(config-if-OverlayRouter 20)# ip address 10.1.2.1/24
TOR1(config-if-OverlayRouter 20)# anycast-gateway
TOR1(config-if-OverlayRouter 20)# route-in-vni
TOR1(config-if-OverlayRouter 20)# exit
TOR1(config)# int overlaytunnel 1
TOR1(config-if-OverlayTunnel 1)# tunnel source 1.1.1.1
TOR1(config-if-OverlayTunnel 1)# tunnel destination 3.3.3.3
TOR1(config-if-OverlayTunnel 1)# evpn mac advertise enable
```

Scenario
Figure 1-29

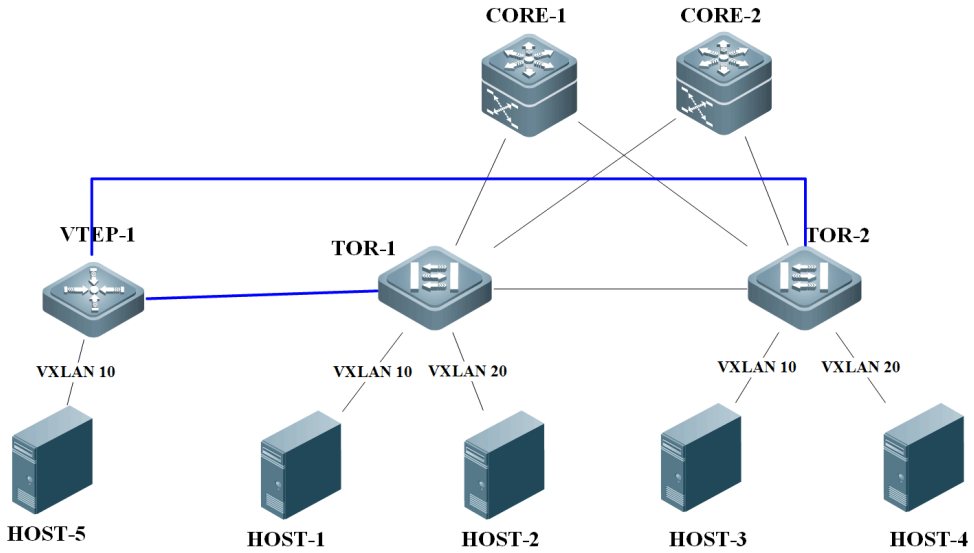


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

```
TOR1(config-if-OverlayTunnel 1)# evpn macip advertise enable
TOR1(config-if-OverlayTunnel 1)# exit
TOR1(config)# vxlan 10
TOR1(config-vxlan)# extend-vlan 10
TOR1(config-vxlan)# router-interface OverlayRouter 10
TOR1(config-vxlan)# tunnel-interface overlaytunnel 1
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# vxlan 20
TOR1(config-vxlan)# extend-vlan 20
TOR1(config-vxlan)# router-interface OverlayRouter 20
TOR1(config-vxlan)# arp suppress enable
TOR1(config-vxlan)# exit
TOR1(config)# router bgp 100
TOR1(config-router)# neighbor 2.2.2.2 remote-as 100
TOR1(config-router)# neighbor 2.2.2.2 update-source loopback 1
TOR1(config-router)# address-family l2vpn evpn
TOR1(config-router-af)# neighbor 2.2.2.2 activate
TOR1(config-router-af)# neighbor 2.2.2.2 route-map dc out
TOR1(config-router-af)# exit
TOR1(config-router)# exit
```

Scenario
Figure 1-29

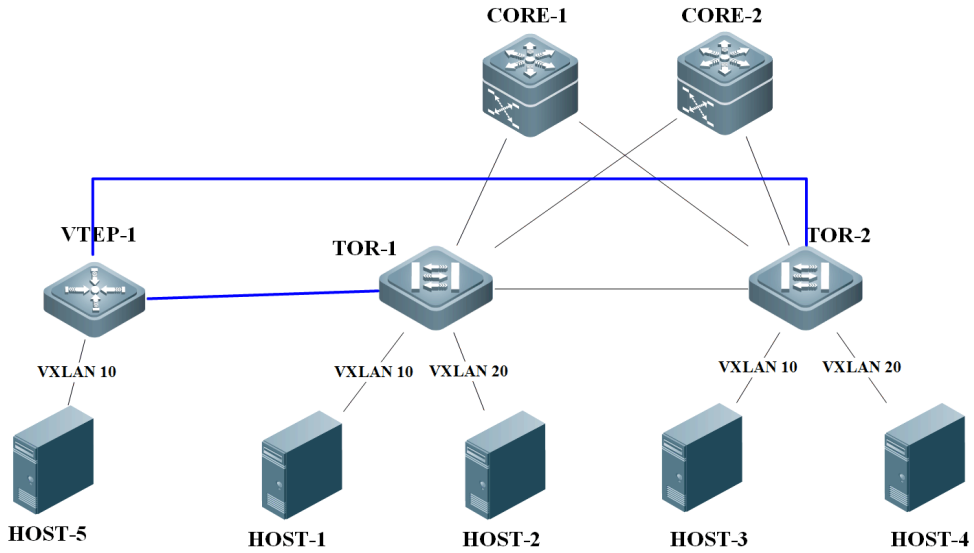


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

	TOR1(config)# evpn TOR1(config-evpn)# vni 10 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit TOR1(config-evpn)# vni 20 TOR1(config-evpn-vni)# rd auto TOR1(config-evpn-vni)# route-target both auto TOR1(config-evpn-vni)# exit
TOR2	TOR2# configure terminal Enter configuration commands, one per line. End with CNTL/Z. TOR2(config)# interface Loopback 1 TOR2(config-if- Loopback 1)# ip address 2.2.2.2/32 TOR2(config-if- Loopback 1)# exit TOR2(config)# vtep TOR2(config-vtep)# source loopback 1 TOR2(config-vtep)# arp suppress enable TOR2(config-vtep)# exit TOR2(config)# fabric anycast-gateway-mac 0000.1234.5678 TOR2(config)# int overlayrouter 10 TOR2(config-if-OverlayRouter 10)# ip address 10.1.1.1/24

Scenario
Figure 1-29

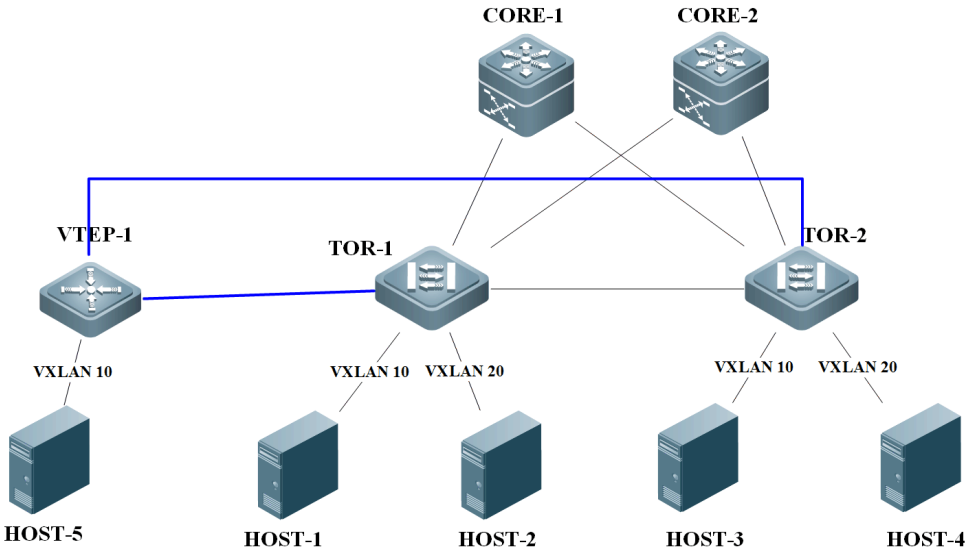


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

```
TOR2(config-if-OverlayRouter 10)# anycast-gateway
TOR2(config-if-OverlayRouter 10)# route-in-vni  //Optional. It needs to be used in combination with the arp suppress enable command.
TOR2(config-if-OverlayRouter 10)# exit
TOR2(config)# int overlayrouter 20
TOR2(config-if-OverlayRouter 20)# ip address 10.1.2.1/24
TOR2(config-if-OverlayRouter 20)# anycast-gateway
TOR2(config-if-OverlayRouter 20)# route-in-vni  //Optional. It needs to be used in combination with the arp suppress enable command.
TOR2(config-if-OverlayRouter 20)# exit
TOR2(config)# vxlan 10
TOR2(config-vxlan)# extend-vlan 10
TOR2(config-vxlan)# router-interface OverlayRouter 10
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# vxlan 20
TOR2(config-vxlan)# extend-vlan 20
TOR2(config-vxlan)# router-interface OverlayRouter 20
TOR2(config-vxlan)# arp suppress enable
TOR2(config-vxlan)# exit
TOR2(config)# router bgp 100
TOR2(config-router)# neighbor 1.1.1.1 remote-as 100
```

Scenario
Figure 1-29

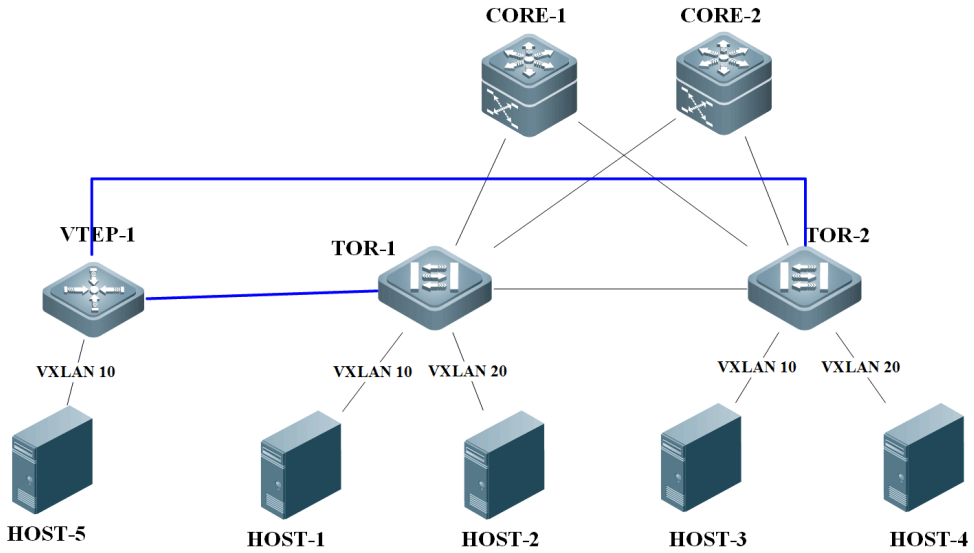


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

	TOR2(config-router)# neighbor 1.1.1.1 update-source loopback 1 TOR2(config-router)# address-family l2vpn evpn TOR2(config-router-af)# neighbor 1.1.1.1 activate TOR2(config-router-af)# exit TOR2(config-router)# exit TOR2(config)# evpn TOR2(config-evpn)# vni 10 TOR2(config-evpn-vni)# rd auto TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# exit TOR2(config-evpn)# vni 20 TOR2(config-evpn-vni)# rd auto TOR2(config-evpn-vni)# route-target both auto TOR2(config-evpn-vni)# exit
VTEP1	VTEP1# configure terminal Enter configuration commands, one per line. End with CNTL/Z. VTEP1(config)# interface Loopback 1 VTEP1(config-if- Loopback 1)# ip address 3.3.3.3/32 VTEP1(config-if- Loopback 1)# exit VTEP1(config)# int overlaytunnel 1 VTEP1(config-if-OverlayTunnel 1)# tunnel source 3.3.3.3

Scenario
Figure 1-29



Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

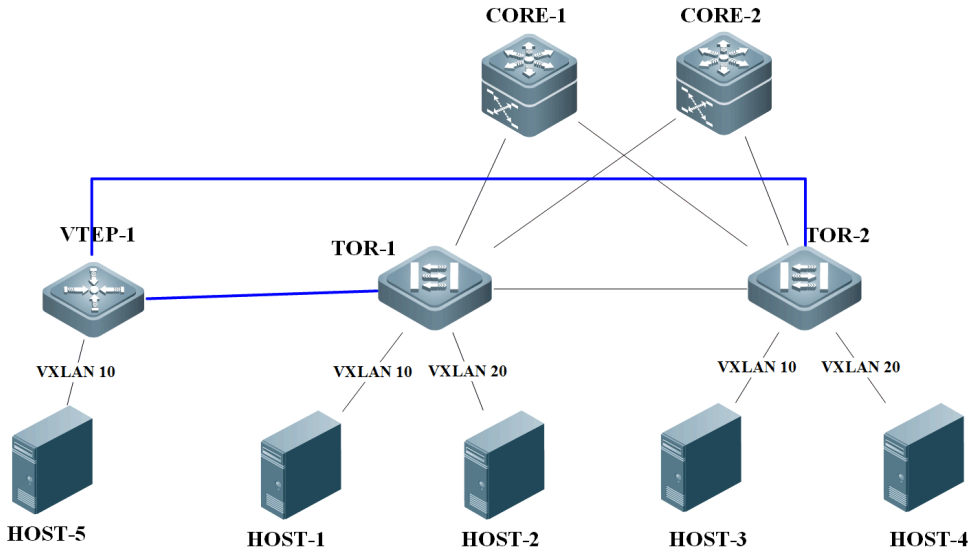
```
VTEP1(config-if-OverlayTunnel 1)# tunnel destination 1.1.1.1
VTEP1(config-if-OverlayTunnel 1)# exit
VTEP1(config)# vxlan 10
VTEP1(config-vxlan)# extend-vlan 10
VTEP1(config-vxlan)# tunnel-interface overlaytunnel 1
```

Verification

- Verify that HOST-1, HOST-2, HOST-3, HOST-4, and HOST-5 can ping each other.
- Verify that virtual machines can be migrated between hosts in the same VXLAN and can access the network normally after migration, with no need to modify the configuration.

```
TOR1# sho vxlan
VXLAN Total Count: 2
VXLAN Capacity   : 8000


VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : overlayrouter 10 (anycast)
   Extend VLAN         : 10
   VTEP Adjacency Count: 2
VTEP Adjacency List :
Interface          Source IP      Destination IP  Type
```

Scenario
Figure 1-29

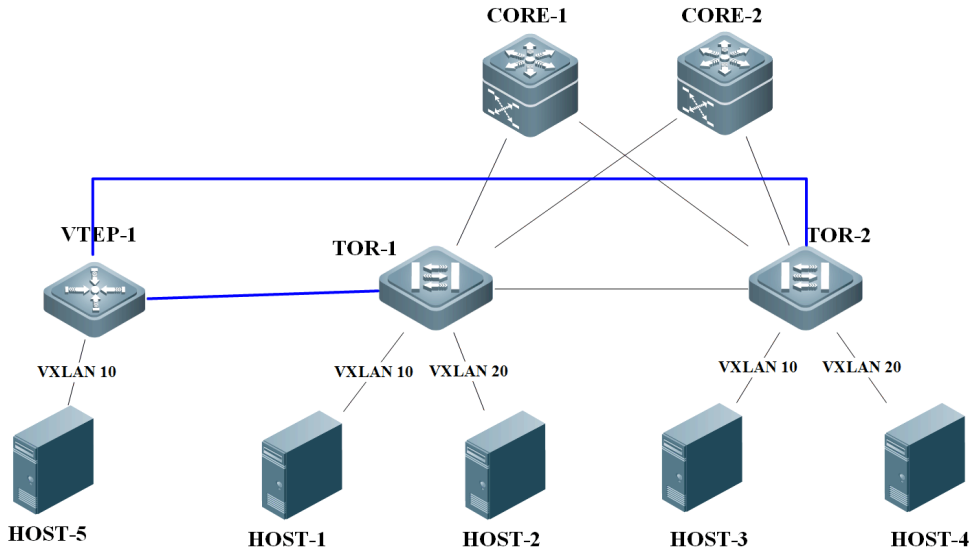


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

```
---------------------- --------------- --------------- -------
OverlayTunnel 1        1.1.1.1         3.3.3.3         static
OverlayTunnel 6145     1.1.1.1         2.2.2.2         dynamic


VXLAN 20
   Symmetric property  : FALSE
   Router Interface    : overlayrouter 20 (anycast)
   Extend VLAN         : 20
   VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface            Source IP       Destination IP  Type
   ---------------------- --------------- --------------- -------
   OverlayTunnel 6145     1.1.1.1         2.2.2.2         dynamic



TOR2# sho vxlan
VXLAN Total Count: 2
VXLAN Capacity   : 8000


VXLAN 10
```

Scenario
Figure 1-29

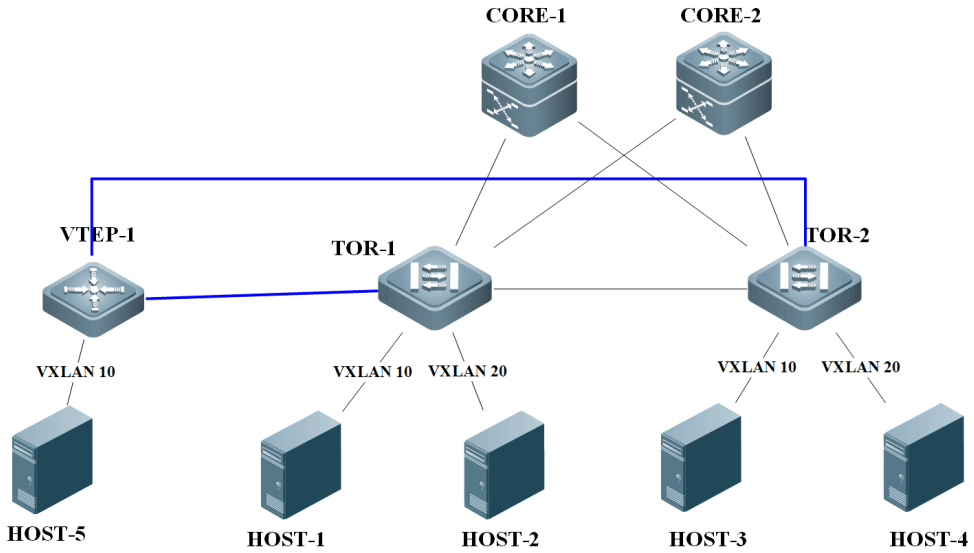


Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

```
    Symmetric property  : FALSE
    Router Interface    : overlayrouter 10 (anycast)
    Extend VLAN         : 10
    VTEP Adjacency Count: 2
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
    ---------------------- -------------- -------------- -------
OverlayTunnel 1       2.2.2.2        3.3.3.3        static
    OverlayTunnel 6145    2.2.2.2        1.1.1.1        dynamic


VXLAN 20
    Symmetric property  : FALSE
    Router Interface    : overlayrouter 20 (anycast)
    Extend VLAN         : 20
    VTEP Adjacency Count: 1
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
    ---------------------- -------------- -------------- -------
    OverlayTunnel 6145    2.2.2.2        1.1.1.1        dynamic
```

Scenario
Figure 1-29



Note:

Blue lines in the figure indicate the VXLAN tunnels that the manually configured VTEP-1 establishes with other VTEPs.

```
VTEP1# sho vxlan
VXLAN Total Count: 1
VXLAN Capacity   : 8000

VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : -
   Extend VLAN         : 10
   VTEP Adjacency Count: 2
VTEP Adjacency List :
Interface           Source IP      Destination IP  Type
   -------------------- --------------- --------------- -------
   OverlayTunnel 1      3.3.3.3        1.1.1.1         static
OverlayTunnel 2       3.3.3.3        2.2.2.2         static
```

1.4.4. Configuring L2 Subinterfaces to Access a VXLAN

Configuration Effect

- Configure hosts to access a VXLAN through L2 subinterfaces.
- L2 subinterfaces can access a VXLAN in VLAN encapsulation or untagged encapsulation mode.

Notes

- If the main interface is a trunk interface, the subinterfaces are not recommended to access a VXLAN in untagged encapsulation mode due to chip limitations. If the untagged encapsulation mode is configured for the subinterfaces, tagged packets are transferred to the logic of subinterfaces using the untagged packaging mode.
- The extend-VLAN configured for a VXLAN instance is like the VLAN or untagged encapsulation rule configured for a subinterface. When a subinterface is available, the encapsulation rule of the subinterface has a higher priority than the extend-VLAN configured for a VXLAN instance.

Configuration Steps

Configuring the VXLAN Encapsulation Rule for a Subinterface

- Mandatory.
- Configure the VXLAN encapsulation rule for a specified subinterface.

Configuring the VLAN and Untagged Encapsulation Rules for a Subinterface

- Mandatory.
- L2 subinterfaces can access a VXLAN in VLAN encapsulation or untagged encapsulation mode.

Creating a VXLAN Instance

- Mandatory.

Associating the VXLAN Instance with an Overlay Router Interface

- Mandatory for VXLAN gateways.
- The device supports the VXLAN routing function and can serve as a VXLAN IP gateway only after the VXLAN is associated with an overlay router interface.

Verification

L2 subinterfaces can access the VXLAN. Run the following commands for verification.

- Run the **show vxlan** *vni-number* command to check the local configuration of the VXLAN.
- Run the **show vxlan mac** to check whether the VXLAN MAC addresses are learned.
- Run the **show arp** command to check whether the ARP entry of the VXLAN IP gateway is learned.
- Run the **show ipv6 neighbors** command to check whether all local/remote IPv6 ND entries are learned.
- Run the **show running** command to display the subinterface configuration.

Related Commands

Configuring the VXLAN Encapsulation Rule for a Subinterface

Command	**encapsulation vxlan** *vni-number*
Parameter Description	*vni-number*: Indicates the VNI. The value ranges from 1 to 16,777,215.
Command Mode	L2 subinterface configuration mode
Usage Guide	1. Only one VXLAN encapsulation rule can be configured for one subinterface. 2. You can preconfigure VXLAN encapsulation rules when no VXLAN instance exists.

Configuring the VLAN and Untagged Encapsulation Rules for a Subinterface	
Command	**encapsulation dot1q {untag | s-vid** *vlan-id***}**
Parameter Description	**{untag | s-vid** *vlan-id***}**: Specifies VLAN encapsulation or untagged encapsulation for a port. The VLAN ID ranges from 1 to 4094 when VLAN encapsulation is adopted.
Command Mode	L2 subinterface configuration mode
Usage Guide	If the main interface is a trunk interface, the subinterfaces are not recommended to access a VXLAN in untagged encapsulation mode due to chip limitations. If the untagged encapsulation mode is configured for the subinterfaces, tagged packets are transferred to the logic of subinterfaces using the untagged packaging mode.

Creating an Overlay Router Interface	
Command	**interface OverlayRouter** *port-id*
Parameter Description	*port-id*: Indicates the ID of an overlay router interface.
Command Mode	Global configuration mode
Usage Guide	This interface serves as the VXLAN IP gateway in the VXLAN routing environment. It is similar to an SVI interface in a VLAN.

Configuring an IP Address for the Overlay Router Interface	
Command	**ip address** *ip-address mask*
Parameter Description	*ip-address*: Indicates the IP address of the overlay router interface. *mask*: Indicates the subnet mask.
Command Mode	Interface configuration mode
Usage Guide	This IP address serves as the VXLAN IP gateway address in the VXLAN routing environment. It is similar to the IP address of an SVI in a VLAN.

Configuring an IPv6 Address for the Overlay Router Interface	
Command	**ipv6 address** *ip-address mask*
Parameter Description	*ip-address*: Indicates the IPv6 address of the overlay router interface. *mask*: Indicates the subnet mask.
Command Mode	Overlay router interface configuration mode
Usage Guide	This IPv6 address serves as the VXLAN IPv6 gateway address in the VXLAN routing environment. It is similar to the IP address of an SVI in a VLAN.

Associating the Overlay Router Interface with a VRF Instance

Command	**vrf forwarding** *table name*
Parameter Description	*table name*: Indicates the VRF instance, with which the overlay router interface is associated.
Command Mode	Interface configuration mode
Usage Guide	This command is used to associate with a VRF instance in the VXLAN routing environment and is used for VXLAN L3 routing isolation.

Creating a VXLAN Instance or Entering the VXLAN Configuration Mode

Command	**vxlan** *vni-number*
Parameter Description	*vni-number*: Indicates the VNI. The value ranges from 1 to 16,777,215.
Command Mode	Global configuration mode
Usage Guide	N/A

Associating the VXLAN Instance with the Overlay Router Interface

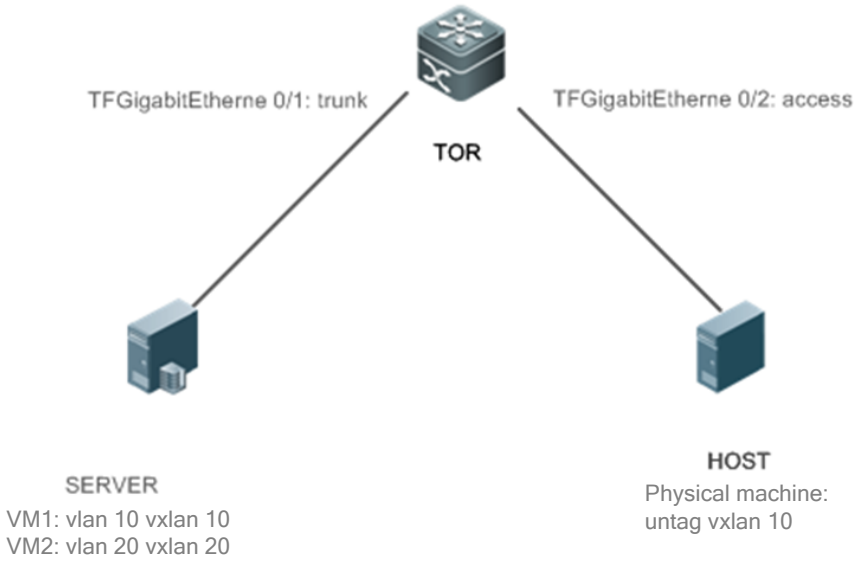
Command	**router-interface** *interface-name*
Parameter Description	*interface-name*: Indicates the name of the overlay router interface.
Command Mode	VXLAN configuration mode
Usage Guide	The **overlay router interfaces** between VXLANs cannot conflict with each other and different VXLANs cannot associate with the same overlay router interface.

Configuration Example

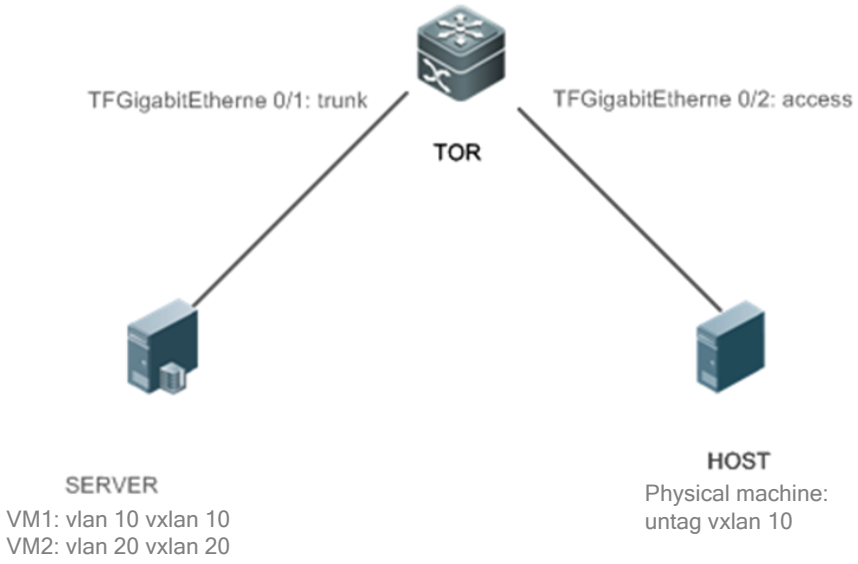
Note:

- Only configuration related to the VXLAN is described below.
- Only IPv4 configuration is used as an example below and the IPv6 scenario configuration is largely the same as the IPv4 scenario configuration.

The recommended configuration is as follows:

Scenario **Figure 1-30**	
Configuration Steps	• Configure a virtual server, virtual machine, and physical server (omitted). • Create an L2 subinterface on the TOR switch, configure VXLAN encapsulation and VLAN or untagged encapsulation rule for the subinterface. • Create an overlay router interface on the TOR switch and configure the VXLAN gateway IP address. • Configure the VXLAN instance to associate with the overlay router interface on the TOR switch to implement VXLAN routing.
TOR	TOR# configure terminal TOR(config)#vlan 10 TOR(config-vlan)#exit TOR(config)#vlan 20 TOR(config-vlan)#exit TOR(config)#interface TFGigabitEthernet 0/1 TOR(config-if-TFGigabitEthernet 0/1)#switchport mode trunk TOR(config-if-TFGigabitEthernet 0/1)#exit TOR(config)#interface TFGigabitEthernet 0/1.1 TOR(config-subif-TFGigabitEthernet 0/1.1)#encapsulation dot1q s-vid 10 TOR(config-subif-TFGigabitEthernet 0/1.1)#encapsulation vxlan 10 TOR(config-subif-TFGigabitEthernet 0/1.1)#exit TOR(config)#interface TFGigabitEthernet 0/1.2 TOR(config-subif-TFGigabitEthernet 0/1.2)#encapsulation dot1q s-vid 20 TOR(config-subif-TFGigabitEthernet 0/1.2)#encapsulation vxlan 20 TOR(config-subif-TFGigabitEthernet 0/1.2)#exit

Scenario
Figure 1-30



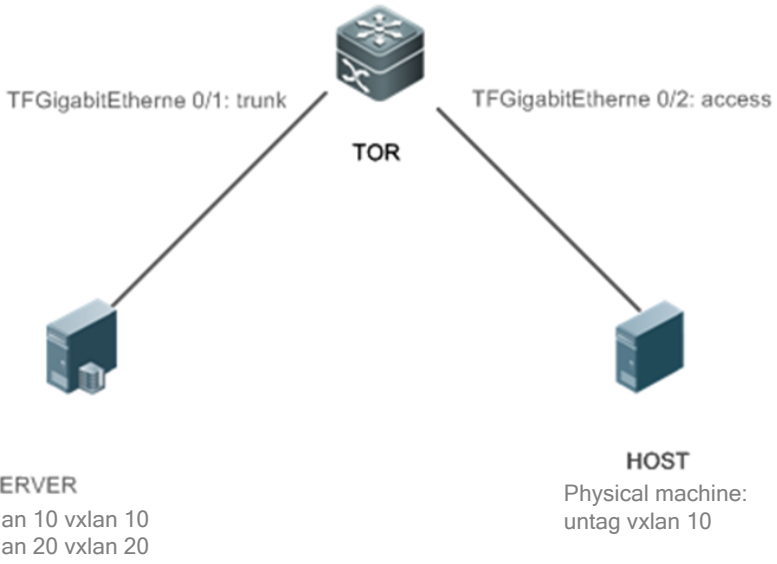

```
TOR(config)#interface TFGigabitEthernet 0/2
TOR(config-if-TFGigabitEthernet 0/2)#switchport mode access vlan 10
TOR(config-if-TFGigabitEthernet 0/2)#exit
TOR(config)#interface TFGigabitEthernet 0/2.1
TOR(config-subif-TFGigabitEthernet 0/2.1)#encapsulation dot1q untag
TOR(config-subif-TFGigabitEthernet 0/2.1)#encapsulation vxlan 10
TOR(config-subif-TFGigabitEthernet 0/2.1)#exit
TOR(config)# interface overlayrouter 10
TOR(config-if-OverlayRouter 10)# ip address 10.1.1.1/24
TOR(config-if-OverlayRouter 10)# exit
TOR(config)# vxlan 10
TOR(config-vxlan)# router-interface OverlayRouter 10
TOR(config-vxlan)# exit
TOR(config)# interface overlayrouter 20
TOR(config-if-OverlayRouter 20)# ip address 20.1.1.1/24
TOR(config-if-OverlayRouter 20)# exit
TOR(config)# vxlan 20
TOR(config-vxlan)# extend-vlan 20
TOR(config-vxlan)# router-interface OverlayRouter 20
TOR1(config-vxlan)# exit
```

Verification

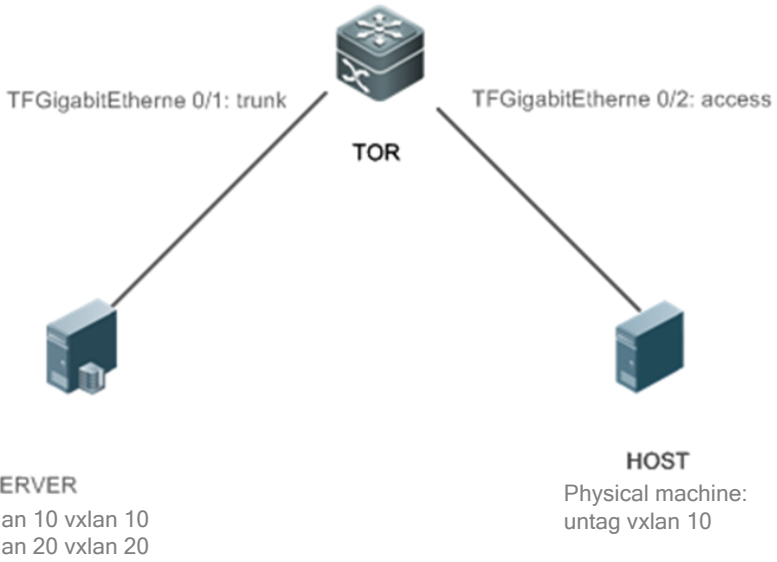
- Verify that the virtual machine and physical machine can ping each other.

```
TOR# sho vxlan
VXLAN Total Count: 2
```

Scenario
Figure 1-30




```
VXLAN Capacity   : 8000

VXLAN 10
   Symmetric property  : FALSE
   Router Interface    : overlayrouter 10 (non-anycast)
   Extend VLAN         : -
   VTEP Adjacency Count: 0

VXLAN 20
   Symmetric property  : FALSE
   Router Interface    : overlayrouter 20 (non-anycast)
   Extend VLAN         : -
   VTEP Adjacency Count: 0

TOR#sh running-config
!
vlan range 1,10,20
!
interface TFGigabitEthernet 0/1
 switchport mode trunk
!
interface TFGigabitEthernet 0/1.1
 encapsulation dot1q s-vid 10
```

Scenario **Figure 1-30**	

```
 encapsulation vxlan 10
!
interface TFGigabitEthernet 0/1.2
 encapsulation dot1q s-vid 20
 encapsulation vxlan 20
!
interface TFGigabitEthernet 0/2
 switchport trunk native vlan 10
!
interface TFGigabitEthernet 0/2.1
 encapsulation dot1q untag
 encapsulation vxlan 10
```

- Banning synchronization of the local MAC address to the remote VTEP through EVPN messages on an L2-VNI VXLAN instance
- Banning delivery of the MAC addresses remotely synchronized through EVPN messages to the local MAC address table on an L2-VNI VXLAN instance
- Stopping an L2-VNI VXLAN instance from generating EVPN type-2 routes

Configuring an L2-VNI VXLAN Instance Not to Synchronize the Local MAC Address to the Remote VTEP Through EVPN Messages

Command	**evpn mac advertise disable**
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	This command is not configured on a device by default. The device

Command	evpn mac advertise disable
	generates one MAC-only type-2 route through the VXLAN-EVPN protocol based on a locally learned MAC entry, and synchronizes the type-2 route to the EVPN neighbor (that is, remote VTEP). Then, the remote VTEP can learn the MAC entry from the MAC-only type-2 route. After this command is configured, the device does not generate VXLAN-EVPN MAC-only type-2 routes based on MAC entries, and therefore, it will not advertise MAC-only type-2 routes to the EVPN neighbor. This command is configured on a VXLAN instance and affects only whether the VXLAN instance generates MAC-only type-2 routes. Other VXLAN instances, for which this command is not configured, can still generate MAC-only type-2 routes. This command can be used in combination with the **evpn arp mac-learning enable** and **evpn nd mac-learning enable** commands. After they are executed, the network-wide VXLAN-EVPN neighbors synchronize only MAC-IP type-2 routes but no MAC-only type-2 routes. All devices parse and extract MAC entries from MAC-IP or MAC-IPv6 type-2 routes. Note: This command can be configured only on L2-VNI VXLAN instances (that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

Configuring an L2-VNI VXLAN Instance Not to Deliver MAC Addresses Remotely Synchronized Through EVPN Messages to the Local MAC Address Table

Command	evpn mac inactive
Parameter Description	*N/A*
Command Mode	VXLAN configuration mode
Usage Guide	After this command is configured, the device does not learn MAC entries from VXLAN-EVPN type-2 routes (MAC-IP or MAC-only type-2 routes) synchronized from neighbors. This command is not configured on a device by default. The device learns MAC entries from VXLAN-EVPN type-2 routes synchronized from neighbors. This command is configured on a VXLAN instance and affects only whether the VXLAN instance learns MAC entries from VXLAN-EVPN type-2 routes. Other VXLAN instances, for which this command is not configured, can still learn MAC entries. Note: This command can be configured only on L2-VNI VXLAN instances (that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

Configuring an L2-VNI VXLAN Instance Not to Generate EVPN Type-2 Routes

Command	evpn rt-2 advertise disable
Parameter Description	*N/A*

Command	evpn rt-2 advertise disable
Command Mode	VXLAN configuration mode
Usage Guide	This command is not configured on a device by default. The device generates one MAC-only type-2 route through the VXLAN-EVPN protocol based on a locally learned MAC entry, and synchronizes the type-2 route to the EVPN neighbor (that is, remote VTEP). Then, the remote VTEP learns the MAC entry from the MAC-only type-2 route. In addition, the device generates one MAC-IP type-2 route through the VXLAN-EVPN protocol based on a locally learned ARP entry and synchronizes the type-2 route to the EVPN neighbor. Then, the remote VTEP learns the ARP entry and host route from the MAC-IP type-2 route. The device generates one MAC-IPv6 type-2 route through the VXLAN-EVPN protocol based on a locally learned IPv6 ND entry, and synchronizes the type-2 route to the EVPN neighbor. Then, the remote VTEP learns the IPv6 ND entry and host route from the MAC-IPv6 type-2 route. After this command is configured, the MAC entries, ARP entries, and IPv6 ND entries of the device are not used to generate VXLAN-EVPN type-2 routes and therefore, no type-2 route is advertised to the EVPN neighbor. This command is configured on a VXLAN instance and affects only whether the VXLAN instance generates type-2 routes. Other VXLAN instances, for which this command is not configured, can still generate type-2 routes. Note: This command can be configured only on L2-VNI VXLAN instances (that is, VXLAN instances with the **symmetric** command not configured). It is unavailable on L3-VNI VXLAN instances.

1.5. Monitoring

Displaying

Description	Command
Displays the VXLAN configuration and status of the device.	**show vxlan** *vni-number*
Displays the MAC addresses learned by the device.	**show vxlan mac [vni** *vni-number*] **[address** *mac-address*]
Displays the VXLAN ARP entries learned by the device.	**show arp**
Displays the VXLAN IPv6 ND entries learned by the device.	**show ipv6 neighbors**
Displays the global configurations of the device, such as the VTEP IP address and anycast MAC address.	**show vxlan global**
Displays the ARP suppression status of the device.	**show vxlan arp suppress**
Displays the VXLAN UDP destination	**show vxlan udp-port**

Description	Command
port of the device.	

2. ОБЩАЯ ИНФОРМАЦИЯ

2.1. Замечания и предложения

Мы всегда стремимся улучшить нашу документацию и помочь вам работать лучше, поэтому мы хотим услышать вас. Мы всегда рады обратной связи, в особенности:

- ошибки в содержании, непонятные или противоречащие места в тексте;
- идеи по улучшению документации, чтобы находить информацию быстрее;
- неработающие ссылки и замечания к навигации по документу.

Если вы хотите написать нам по поводу данного документа, то используйте, пожалуйста, форму обратной связи на qtech.ru.

2.2. Гарантия и сервис

Процедура и необходимые действия по вопросам гарантии описаны на сайте QTECH в разделе «Поддержка» −> «Гарантийное обслуживание».

Ознакомиться с информацией по вопросам тестирования оборудования можно на сайте QTECH в разделе «Поддержка» −> «Взять оборудование на тест».

Вы можете написать напрямую в службу сервиса по электронной почте sc@qtech.ru.

2.3. Техническая поддержка

Если вам необходимо содействие в вопросах, касающихся нашего оборудования, то можете воспользоваться нашей автоматизированной системой запросов технического сервис-центра helpdesk.qtech.ru.

Телефон Технической поддержки +7 (495) 477-81-18 доб. 0