VXLAN Configuration

www.qtech.ru

Оглавление

1. DETAILED CONFIGURATION

1.1 Configure VXLAN SDN

Configuration effect

- Create a VXLAN instance and associate the OverlayRouter port and OverlayTunnel port to provide the VXLAN routing (IP gateway) function to achieve cross-VXLAN communication. The VXLAN configuration can be issued by the SDN controller through communication mechanisms such as NETCONF. It can also be configured via CLI.
- For the centralized anycast all-active gateway, you need to configure the anycast gateway and anycast MAC address. The centralized anycast all-active gateway is externally embodied as a logical gateway VTEP, using the same VTEP IP. Only one tunnel is configured for each TOR bridge device and logical gateway VTEP. Through the underlay ECMP, the message load is balanced to the physical gateway to realize the gateway's multi-active function.
- The SDN controller will issue the host routing and VXLAN forwarding flow table to the device, and the device can also generate routes and table entries through device self-learning. The self-learning capability of the device is turned on or off through configuration commands. When the controller fails, the normal operation of the VXLAN network can be ensured through device self-learning.

Precautions

- The VXLAN configuration can be issued by the SDN controller through communication mechanisms such as NETCONF. It can also be configured via CLI. It is recommended that the configuration is only delivered through the SDN controller under normal circumstances.
- VXLAN instances need to use existing unicast routing in the network. Therefore, devices in the network must be configured with IPv4 unicast routing, such as OSPF.
- On a centralized anycast gateway, the anycast IP of the gateway needs to be divided into different routing domains to avoid IP conflicts.

Configuration method

↘ Create VXLAN instance

- Must be configured.

↘ Create OverlayRouter port

- The VXLAN gateway must be configured.

↘ Configure the OverlayRouter port as anycast

- The centralized anycast gateway must be configured.

↘ Configure anycast MAC address

- The centralized anycast gateway must be configured.

↘ ConfigurationOverlayTunnel tunnel

- Must be configured.

↘ ConfigurationOverlayTunnel tunnel source IP and destination IP

- Must be configured.

↘ Configure the VXLAN instance to associate with the OverlayRouter port

- The gateway must be configured.

↘ Associate VLAN with VXLAN instance

- The TOR bridge must be configured.

↘ Associate a VXLAN instance with an OverlayTunnel tunnel

- Must be configured.
- Used to statically specify a VXLAN tunnel.

↘ Configure static VXLAN MAC address table

- Optional. The VXLAN MAC address table issued by the SDN controller is a static VXLAN MAC address table.
- You can also configure a static VXLAN MAC address table through the CLI.

↘ Configure VXLAN UDP destination port

- Optional configuration. Since the VXLAN UDP destination port used by earlier devices may not be 4789, this command can be used for compatibility. In addition, the VXLAN UDP destination port can also be customized through this command.
- The VXLAN UDP destination port 4789 designated by IANA is used by default.

↘ Configure to enable ARP self-learning

- Optional. ARP self-learning is enabled by default.
- After enabling ARP self-learning, the device can learn host ARP entries on its own without relying entirely on the SDN controller to issue.

↘ Configure to enable IPv6 ND self-learning

- Optional. IPv6 ND self-learning is enabled by default.
- After the IPv6 ND self-learning is enabled, the device can learn the host ND table entries on its own without relying entirely on the SDN controller to deliver.

Testing method

After configuring the SDN-VXLAN function, virtual machines can communicate with each other.

- Use show vxlan vni-number to check whether VXLAN devices have learned each other's VTEP adjacency.
- Check whether the VXLAN MAC address has been learned through show vxlan mac.
- Check whether all local/remote entries have been learned through show arp. Use show ip route to check whether the route of the VXLAN IP gateway has been learned.
- Check whether all local/remote IPv6 ND entries have been learned through show ipv6 neighbors. Use show ipv6 route to check whether the route of the VXLAN IPv6 gateway has been learned.
- View the VXLAN UDP destination port through show vxlan udp-port.

Related commands

↘ Create or enter VXLAN instance

【Command format】	**vxlan** *vni-number*
【Parameter Description】	*vni-number*: VNI (VXLAN Network Identifier), the value range is 1~16777215.
【Command Mode】	Global configuration mode
【User guides】	-

↘ Associate VLAN with VXLAN instance

【Command format】	**extend-vlan** *vlan-id-list*
【Parameter Description】	*vlan-id-list*: VLAN-ID queue, the range of VLAN-ID is 1~4094.
【Command Mode】	VXLAN configuration mode
【User guides】	Configure the VLAN associated with the VXLAN instance, the VLAN packets received by the device will be associated with the VXLAN instance

↘ Create OverlayRouter port

【Command format】	**interface OverlayRouter** port-id
【Parameter	*port-id*： OverlayRouter port ID, the value range is 1~8000

Description】	
【Command Mode】	Global configuration mode
【User guides】	This interface is used as a VXLAN IP gateway in the VXLAN routing environment, similar to the SVI port in the VLAN.

↘ Configure OverlayRouter port VRF

【Command format】	**vrf forwarding** *vrf-name*
【Parameter Description】	*vrf-name*：VRF to which the OverlayRouter port belongs
【Command Mode】	OverlayRouter interface configuration mode
【User guides】	This VRF is used for VXLAN tenant division. Traffic isolation between VXLAN instances of different VRFs.

↘ Configure the IP address of the OverlayRouter port

【Command format】	**ip address** *ip-address mask*
【Parameter Description】	*ip-address*: IP address of OverlayRouter port *mask*: Subnet mask
【Command Mode】	OverlayRouter interface configuration mode
【User guides】	This IP address is used as the VXLAN IP gateway address in the VXLAN routing environment, similar to the IP address of the SVI in the VLAN.

↘ Configure the IPv6 address of the OverlayRouter port

【Command format】	**ipv6 address** *ip-address mask*
【Parameter Description】	*ip-address*: IPv6 address of the OverlayRouter port *mask*: Subnet mask
【Command Mode】	OverlayRouter interface configuration mode
【User guides】	This IPv6 address is used as the VXLAN IPv6 gateway address in the VXLAN routing environment, similar to the IP address of the SVI in the VLAN.

↘ Configure the OverlayRouter port as anycast

【Command format】	**anycast-gateway**
【Parameter Description】	-
【Command Mode】	OverlayRouter interface configuration mode
【User guides】	The gateway is configured as anycast gateway.

↘ Associate the VXLAN instance with the OverlayRouer port

【Command format】	**router-interface** *interface-name*
【Parameter Description】	*interface-name*: OverlayRouter interface name
【Command Mode】	VXLAN configuration mode
【User guides】	Router-interfaces between VXLANs cannot conflict, and different VXLANs cannot be associated with the same OverlayRouter port.

↘ Configure ANYCAST virtual mac

【Command format】	**fabric anycast-gateway-mac***mac-addr*
【Parameter Description】	*mac-addr: mac address, format xxxx.xxxx.xxxx*
【Command Mode】	Global configuration mode
【User guides】	All gateways that enable anycast use this mac as the gateway mac. It is not allowed to configure a native mac, nor is it allowed to be configured the same as the mac of any device in the overlay network

↘ Create OverlayTunnel port

【Command format】	**interface OverlayTunnel** *port-id*
【Parameter Description】	*port-id*: OverlayTunnel port ID

【Command Mode】	Global configuration mode
【User guides】	This interface is used to statically create an Overlay tunnel, which can be associated with VXLAN through the tunnel-interface command.

Configure OverlayTunnel port tunnel source IP

【Command format】	**tunnel source** *ip-address*
【Parameter Description】	*ip-address*: Tunnel source IP
【Command Mode】	OverlayTunnel interface configuration mode
【User guides】	Specify the source IP of the overlay tunnel. The outer source IP of the packet is specified as the address when the packet is encapsulated and forwarded.

Configure the destination IP of the OverlayTunnel port

【Command format】	**tunnel destination** *ip-address*
【Parameter Description】	*ip-address*: Tunnel destination IP
【Command Mode】	OverlayTunnel interface configuration mode
【User guides】	Specify the destination IP of the overlay tunnel. The outer destination IP of the packet is specified as the address when the packet is encapsulated and forwarded. The destination IP of the tunnel is globally unique. Different OverlayTunnels cannot be configured with the same destination IP, otherwise the configuration conflicts.

Configure the VXLAN instance to associate the OverlayTunnel port

【Command format】	**tunnel-interface** *interface-name*
【Parameter Description】	*interface-name*: OverlayTunnel interface name
【Command Mode】	VXLAN configuration mode
【User guides】	Used to statically specify VXLAN VTEP.

Configure static VXLAN MAC address table

【Command format】	**vxlan mac static***mac-addr* **vni** *vxlan-id* **interface** *interface-name*[**vlan***vid*]
【Parameter Description】	*mac-addr*: mac entry address
	vxlan-id: vni to which mac entry belongs *interface-name*: The next hop exit of the mac table, which may be the overlay tunnel port, or Ethernet port, or AP aggregation port *vid*: The vlan id of the mac entry
【Command Mode】	Global configuration mode
【User guides】	1. Send static VXLAN MAC address entries through SDN, or statically configure through CLI, which is mainly used for host forwarding table. 2. When the next hop interface is a non-overlaytunnel tunnel port, vid must be configured, and the overlaytunnel tunnel port does not need to be configured.

↘ Configure VXLAN UDP destination port

【Command format】	**vxlan udp-port** *port-number*
【Parameter Description】	*port-number*: UDP destination port number, the value range is 0~65535, and the default value is 4789.
【Command Mode】	Global configuration mode
【User guides】	Be careful not to conflict with common UDP ports

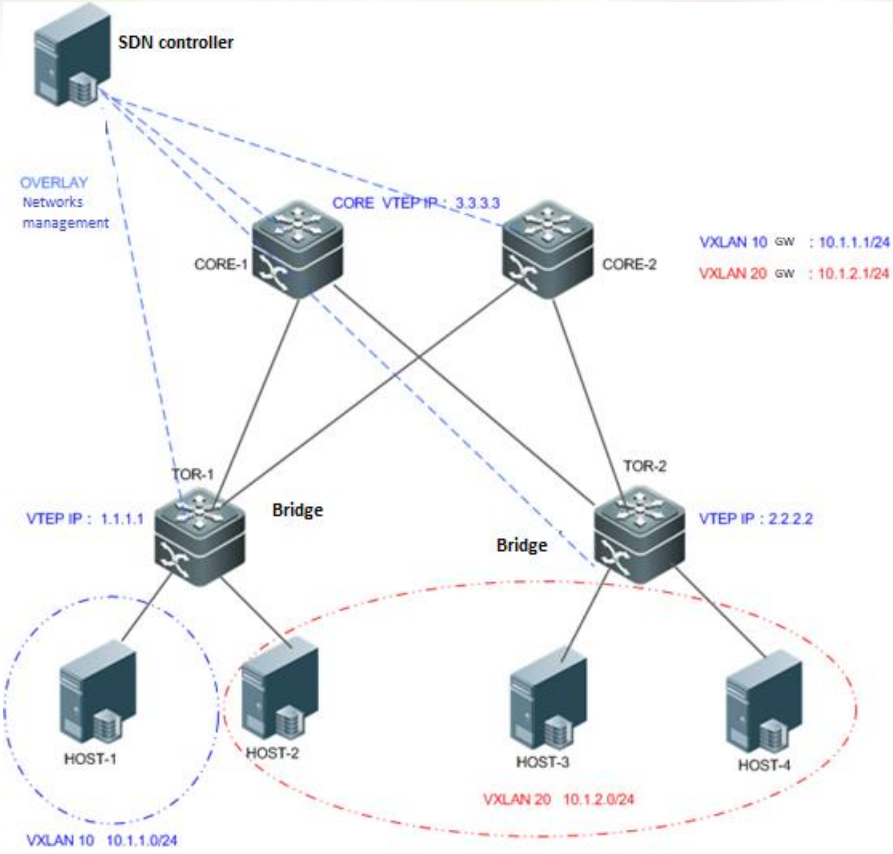
Configuration example

- The following configuration examples only introduce the configuration related to VXLAN.
- The following configuration examples only take IPv4 configuration as an example configuration. The IPv6 scenario configuration is basically the same as that of IPv4.

↘ VXLAN configuration example

【Web environment】

Figure0-1



[Configuration method]

- Configure IPv4 unicast routing protocols (such as OSPF) on TOR1, TOR2, CORE1, and CORE2 to ensure that unicast routes are reachable. (slightly)
- Configure LOOPBACK IP in TOR1, TOR2, CORE1, CORE2 and distribute them through unicast routing protocol. Among them, the VTEP IP configured by CORE1 and CORE2 is the same and needs to be divided into different routing domains.
- Configure VXLAN on the virtualization server and specify the gateway address of the VM. (slightly)
- Create a BGP neighbor between CORE1 and CORE2, and configure BGP-EVPN.

The following configurations can be delivered through the SDN controller:

- Create VXLAN instances VXLAN 10 and VXLAN 20 in TOR1. Associate VLAN10 and VLAN20 respectively. Configure the address learning mode to be notified by the SDN controller. Configure two overlaytunnel tunnels to connect TOR2 and CORE. Configure VXLAN10 and VXLAN20 to associate two tunnels respectively.
- Create a VXLAN instance VXLAN20 in TOR2 and associate it with VLAN20. Configure the address learning mode to be notified by the SDN controller. Configure two overlaytunnel tunnels to connect TOR1 and CORE. Configure VXLAN20 to associate two tunnels.
- Create VXLAN instances VXLAN10 and VXLAN20 on CORE1. Configure the address learning mode to be notified by the SDN controller. Configure anycast-MAC address. Configure two overlayrouter gateway interfaces and configure IP addresses as

10.1.1.1/24 and 10.1.2.1/24 respectively. Configure VXLAN10 to associate with the overlayrouter gateway interface of 10.1.1.1/24. Configure VXLAN20 to associate with the overlayrouter gateway interface of 10.1.2.1/24. Configure two overlaytunnel tunnels to connect TOR1 and TOR2. Configure VXLAN10 and VXLAN20 to associate two tunnels respectively.

- The configuration of CORE2 is basically the same as that of CORE1.

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
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.
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
TOR2(config-vxlan)# extend-vlan 20
TOR2(config-vxlan)# end
```

CORE1

Create VXLAN instances VXLAN10 and VXLAN20 on CORE1. Configure the address learning mode to be notified by the SDN controller. Configure anycast-MAC address. Configure two overlayrouter gateway interfaces and configure IP addresses as 10.1.1.1/24 and 10.1.2.1/24 respectively. Configure VXLAN10 to associate with the overlayrouter gateway interface of 10.1.1.1/24. Configure VXLAN20 to associate with the overlayrouter gateway interface of 10.1.2.1/24. Configure two overlaytunnel tunnels to connect TOR1 and TOR2. Configure VXLAN10 and VXLAN20 to associate two tunnels respectively. Configure loopback 1 on CORE1 and CORE2 respectively. The IP addresses are 3.3.3.4 and 3.3.3.5 respectively. The two COREs establish a bgp neighbor relationship through loopback 1 port and are activated under the l2vpn evpn address family.

```
TOR1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
CORE1(config)# fabric anycast-gateway-mac 0000.1234.5678
CORE1(config)# fabric anycast-gateway-mac 0011.2233.4455
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
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
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
```

CORE2	Same as CORE1
【Testing method】	● Ping each other from the virtual machine of HOST1/HOST2/HOST3/HOST4 and it can be pinged normally ● You can see the host static MAC table entries issued by the SDN controller on the TOR/CORE