Contents

Configuring PBR 1

About PBR 1

Packet forwarding process 1

PBR types 1

Policy 1

PBR and Track 2

Restrictions and guidelines: PBR configuration 2

PBR tasks at a glance 2

Configuring a policy 3

Creating a node 3

Setting match criteria for a node 3

Configuring actions for a node 4

Specifying a policy for PBR 6

Specifying a policy for local PBR 6

Specifying a policy for interface PBR 6

Specifying a policy for outbound PBR on a VXLAN tunnel interface 7

Display and maintenance commands for PBR 7

PBR configuration examples 8

Example: Configuring packet type-based local PBR 8

Example: Configuring packet type-based interface PBR 10

Example: Configuring VXLAN ID-based outbound PBR 11

Example: Configuring EVPN-based service chain PBR 13

Configuring PBR

About PBR

Policy-based routing (PBR) uses user-defined policies to route packets. A policy can specify parameters for packets that match specific criteria such as ACLs or that have specific VXLAN IDs. The parameters include the next hop and default next hop.

Packet forwarding process

The device forwards received packets using the following process:

1. The device uses PBR to forward matching packets.
2. If one of the following events occurs, the device searches for a route (except the default route) in the routing table to forward packets:
   - The packets do not match the PBR policy.
   - The PBR-based forwarding fails.
3. If the forwarding fails, the device uses the default next hop defined in PBR to forward packets.
4. If the forwarding fails, the device uses the default route to forward packets.

PBR types

PBR includes the following types:

- **Local PBR**—Guides the forwarding of locally generated packets, such as ICMP packets generated by using the **`ping`** command.
- **Interface PBR**—Guides the forwarding of packets received on an interface.
- **Outbound PBR on a VXLAN tunnel interface**—Guides the forwarding of outgoing packets when equal-cost routes exist.

Policy

A policy includes match criteria and actions to be taken on the matching packets. A policy can have one or multiple nodes as follows:

- Each node is identified by a node number. A smaller node number has a higher priority.
- A node contains **`if-match`** and **`apply`** clauses. An **`if-match`** clause specifies a match criterion, and an **`apply`** clause specifies an action.
- A node has a match mode of **permit** or **deny**.

A policy compares packets with nodes in priority order. If a packet matches the criteria on a node, it is processed by the action on the node. If the packet does not match any criteria on the node, it goes to the next node for a match. If the packet does not match the criteria on any node, the device performs a routing table lookup.

Relationship between if-match clauses

On a node, you can specify multiple types of **`if-match`** clauses but only one **`if-match`** clause for each type.

To match a node, a packet must match all types of the **`if-match`** clauses for the node.

Relationship between apply clauses

You can specify multiple **apply** clauses for a node, but some of them might not be executed. For more information about relationship between **apply** clauses, see "Configuring actions for a node."

Relationship between the match mode and clauses on the node

Does a packet match all the if-match clauses on the node?	Match mode	
	Permit	**Deny**
Yes.	• If the node contains **apply** clauses, PBR executes the **apply** clauses on the node. ○ If PBR-based forwarding succeeds, PBR does not compare the packet with the next node. ○ If PBR-based forwarding fails, PBR does not compare the packet with the next node. • If the node does not contain **apply** clauses, the device performs a routing table lookup for the packet.	The device performs a routing table lookup for the packet.
No.	PBR compares the packet with the next node.	PBR compares the packet with the next node.

NOTE:

A node that has no **if-match** clauses matches any packet.

PBR and Track

PBR can work with the Track feature to dynamically adapt the availability status of an **apply** clause to the link status of a tracked object. The tracked object can be a next hop or default next hop.

- When the track entry associated with an object changes to **Negative**, the **apply** clause is invalid.
- When the track entry changes to **Positive** or **NotReady**, the **apply** clause is valid.

For more information about Track and PBR collaboration, see *High Availability Configuration Guide*.

Restrictions and guidelines: PBR configuration

If a packet destined for the local device matches a PBR policy, PBR will execute the apply clauses in the policy, including the clause for forwarding. When you configure a PBR policy, be careful to avoid this situation.

PBR tasks at a glance

To configure PBR, perform the following tasks:

1. Configuring a policy
   a. Creating a node
   b. Setting match criteria for a node

c. Configuring actions for a node

2. Specifying a policy for PBR

   Choose the following tasks as needed:

   - Specifying a policy for local PBR
   - Specifying a policy for interface PBR
   - Specifying a policy for outbound PBR on a VXLAN tunnel interface

Configuring a policy

Creating a node

1. Enter system view.

   system-view

2. Create a node for a policy, and enter its view.

   policy-based-route *policy-name* [**deny** | **permit**] **node** *node-number*

3. (Optional.) Configure a description for the policy node.

   description *text*

   By default, no description is configured for a policy node.

Setting match criteria for a node

Restrictions and guidelines

On a transport network device, you can configure PBR on the Layer 3 interface to guide the forwarding of packets based on VXLAN IDs. On a VTEP, you can configure PBR on the tunnel interface to guide the forwarding of packets based on VXLAN IDs.

Procedure

1. Enter system view.

   system-view

2. Enter policy node view.

   policy-based-route *policy-name* [**deny** | **permit**] **node** *node-number*

3. Set match criteria.

   - Set an ACL match criterion.

     if-match acl { *acl-number* | **name** *acl-name* }

     By default, no ACL match criterion is set.

     The ACL match criterion cannot match Layer 2 information.

     When using the ACL to match packets, PBR ignores the action (permit or deny) and time range settings in the ACL.

   - Set a local QoS ID match criterion.

     if-match qos-local-id *local-id-value* [**qppb-manipulation**]

     By default, no local QoS ID match criterion is set.

     If you specify the **qppb-manipulation** keyword, the command applies only to QPPB in which the device acts as a BGP receiver. The device determines that a received packet matches the policy node if the local QoS ID obtained from its matching route entry is identical to the specified local QoS ID criterion. For more information about QPPB, see *ACL and QoS Configuration Guide*.

- Set a VXLAN match criterion.

  if-match vxlan-id *vxlan-id*

  By default, no VXLAN match criterion is set.
- Set a service chain match criterion.

  if-match service-chain { **path-id** *service-path-id* [**path-index** *service-path-index*] }

  By default, no service chain match criterion is set.

Configuring actions for a node

About apply clauses

The **apply** clauses allow you to specify the actions to be taken on matching packets on a node.

The following **apply** clauses determine the packet forwarding paths in a descending order:

- **apply next-hop**
- **apply output-interface NULL0**
- **apply default-next-hop**

PBR supports the **apply** clauses in Table 1.

Table 1 Apply clauses supported in PBR

Clause	Meaning	Remarks
apply precedence	Sets an IP precedence.	This clause is always executed.
apply next-hop	Sets next hops.	N/A
apply output-interface NULL0	Sets the output interface to Null 0.	If both the **apply next-hop** and **apply output-interface NULL0** clauses are configured, only the **apply next-hop** clause is executed.
apply service-chain	Sets service chain information.	For this clause to take effect, you must also use the **apply next-hop** clause to specify a reachable next hop.
apply default-next-hop	Sets default next hops.	The clauses take effect only in the following cases: • No next hops are set or the next hops are invalid. • The packet does not match any route in the routing table.
apply fail-action-drop next-hop	Sets the action that drops matching packets when all next hops become invalid.	If no next hops are specified on the policy node, this clause drops all packets that match the policy node. This clause is typically used in scenarios that require strict routing paths.

Restrictions and guidelines

If you specify a next hop or default next hop, PBR periodically performs a lookup in the FIB table to determine its availability. Temporary service interruption might occur if PBR does not update the route immediately after its availability status changes.

Configuring actions to modify packet fields

1. Enter system view.

   system-view

2. Enter policy node view.

   policy-based-route *policy-name* [**deny** | **permit**] **node** *node-number*

3. Set an IP precedence.

   apply precedence { *type* | *value* }

   By default, no IP precedence is specified.

Configuring actions to direct packet forwarding

1. Enter system view.

   system-view

2. Enter policy node view.

   policy-based-route *policy-name* [**deny** | **permit**] **node** *node-number*

3. Configure actions.
   - Set next hops.

     apply next-hop [**vpn-instance** *vpn-instance-name*] { *ip-address* [**direct**] [**track** *track-entry-number*] }&<1-2>

     By default, no next hops are specified.

     On a node, you can specify a maximum of two next hops for backup in one command line or by executing this command multiple times.
   - Set default next hops.

     apply default-next-hop [**vpn-instance** *vpn-instance-name*] { *ip-address* [**direct**] [**track** *track-entry-number*] }&<1-2>

     By default, no default next hops are specified.

     On a node, you can specify a maximum of two default next hops for backup in one command line or by executing this command multiple times.
   - Set the output interface to Null 0.

     apply output-interface NULL 0

     By default, no action is configured to set the output interface to Null 0.
   - Set service chain information.

     apply service-chain path-id *service-path-id* [**path-index** *service-path-index*]

     By default, no service chain information is set.

     This clause does not apply to software forwarded packets.
   - Set the action that drops matching packets when all next hops specified on the policy node are invalid.

     apply fail-action-drop next-hop

     By default, the drop action is not configured. The matching packets are forwarded based on the typical packet forwarding process as described in "Packet forwarding process."

     This command does not apply to software-forwarded packets.

Specifying a policy for PBR

Specifying a policy for local PBR

About local PBR

Perform this task to specify a policy for local PBR to guide the forwarding of locally generated packets.

Restrictions and guidelines

You can specify only one policy for local PBR and must make sure the specified policy already exists. Before you apply a new policy, you must first remove the current policy.

Local PBR might affect local services such as ping and Telnet. When you use local PBR, make sure you fully understand its impact on local services of the device.

Procedure

1. Enter system view.

   `system-view`

2. Specify a policy for local PBR.

   `ip local policy-based-route` *`policy-name`*

   By default, local PBR is not enabled.

Specifying a policy for interface PBR

About interface PBR

Perform this task to apply a policy to an interface to guide the forwarding of packets received on the interface.

Restrictions and guidelines

You can apply only one policy to an interface and must make sure the specified policy already exists. Before you can apply a new interface PBR policy to an interface, you must first remove the current policy from the interface.

You can apply a policy to multiple interfaces.

In a VXLAN IP gateway or EVPN gateway deployment, when you apply a policy to an interface on a border gateway, follow these restrictions and guidelines:

- If the interface is a Layer 3 Ethernet interface, the policy is also applied to its subinterfaces. For more information about VXLAN IP gateway deployment, see *VXLAN Configuration Guide*.
- If the interface is a Layer 3 aggregate interface, the policy is also applied to its subinterfaces. For more information about EVPN gateway deployment, see *EVPN Configuration Guide*.

Procedure

1. Enter system view.

   `system-view`

2. Enter interface view.

   `interface` *`interface-type interface-number`*

3. Specify a policy for interface PBR.

   `ip policy-based-route` *`policy-name`*

   By default, no interface policy is applied to an interface.

Specifying a policy for outbound PBR on a VXLAN tunnel interface

About outbound PBR on a VXLAN tunnel interface

In a VXLAN network, equal-cost routes might exist between two endpoints of a VXLAN tunnel. The device cannot route VXLAN packets to the exact next hop. To choose the desired next hop for outgoing VXLAN packets, use outbound PBR on the VXLAN tunnel interface.

Restrictions and guidelines

You can specify only one policy to a VXLAN tunnel interface and must make sure the specified policy already exists. Before you can apply a new policy to an interface, you must first remove the current policy from the interface.

Procedure

1. Enter system view.

   `system-view`

2. Create a VXLAN tunnel interface and enter tunnel interface view.

   `interface tunnel` *`tunnel-number`* **`mode vxlan`**

   The endpoints of a tunnel must use the same tunnel mode to correctly transmit packets.

3. Specify a policy for outbound PBR.

   `ip policy-based-route` *`policy-name`* **`egress`**

   By default, no policy is specified for outbound PBR on a VXLAN tunnel interface.

Display and maintenance commands for PBR

Execute **`display`** commands in any view and **`reset`** commands in user view.

Task	Command
Display PBR policy information.	**`display ip policy-based-route`** [**`policy`** *`policy-name`*]
Display outbound PBR configuration and statistics for an interface.	**`display ip policy-based-route egress interface`** *`interface-type interface-number`* [**`slot`** *`slot-number`*]

Task	Command
Display interface PBR configuration and statistics.	**display ip policy-based-route interface** *interface-type interface-number* [**slot** *slot-number*]
Display local PBR configuration and statistics.	**display ip policy-based-route local** [**slot** *slot-number*]
Display PBR configuration.	**display ip policy-based-route setup**
Clear PBR statistics.	**reset ip policy-based-route statistics** [**policy** *policy-name*]

PBR configuration examples

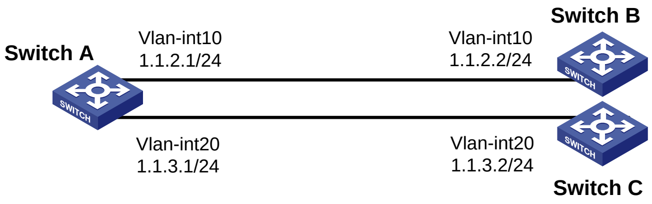
Example: Configuring packet type-based local PBR

Network configuration

As shown in Figure 1, Switch B and Switch C do not have a route to reach each other.

Configure PBR on Switch A to forward all TCP packets to the next hop 1.1.2.2 (Switch B).

Figure 1 Network diagram



Procedure

1. Configure Switch A:

   # Create VLAN 10 and VLAN 20.

   ```
   <SwitchA> system-view
   [SwitchA] vlan 10
   [SwitchA-vlan10] quit
   ```

```
[SwitchA] vlan 20
[SwitchA-vlan20] quit
```

# Configure the IP addresses of VLAN-interface 10 and VLAN-interface 20.

```
[SwitchA] interface vlan-interface 10
[SwitchA-Vlan-interface10] ip address 1.1.2.1 24
[SwitchA-Vlan-interface10] quit
[SwitchA] interface vlan-interface 20
[SwitchA-Vlan-interface20] ip address 1.1.3.1 24
[SwitchA-Vlan-interface20] quit
```

# Configure ACL 3101 to match TCP packets.

```
[SwitchA] acl advanced 3101
[SwitchA-acl-ipv4-adv-3101] rule permit tcp
[SwitchA-acl-ipv4-adv-3101] quit
```

# Configure Node 5 for the policy **aaa** to forward TCP packets to next hop 1.1.2.2.

```
[SwitchA] policy-based-route aaa permit node 5
[SwitchA-pbr-aaa-5] if-match acl 3101
[SwitchA-pbr-aaa-5] apply next-hop 1.1.2.2
[SwitchA-pbr-aaa-5] quit
```

# Configure local PBR by applying the policy **aaa** to Switch A.

```
[SwitchA] ip local policy-based-route aaa
```

2. Configure Switch B:

# Create VLAN 10.

```
<SwitchB> system-view
[SwitchB] vlan 10
[SwitchB-vlan10] quit
```

# Configure the IP address of VLAN-interface 10.

```
[SwitchB] interface vlan-interface 10
[SwitchB-Vlan-interface10] ip address 1.1.2.2 24
```

3. Configure Switch C:

# Create VLAN 20.

```
<SwitchC> system-view
[SwitchC] vlan 20
[SwitchC-vlan20] quit
```

# Configure the IP address of VLAN-interface 20.

```
[SwitchC] interface vlan-interface 20
[SwitchC-Vlan-interface20] ip address 1.1.3.2 24
```

Verifying the configuration

1. Perform telnet operations to verify that local PBR on Switch A operates as configured to forward the matching TCP packets to the next hop 1.1.2.2 (Switch B), as follows:

# Verify that you can telnet to Switch B from Switch A successfully. (Details not shown.)

# Verify that you cannot telnet to Switch C from Switch A. (Details not shown.)

2. Verify that Switch A forwards packets other than TCP packets through VLAN-interface 20. For example, verify that you can ping Switch C from Switch A. (Details not shown.)

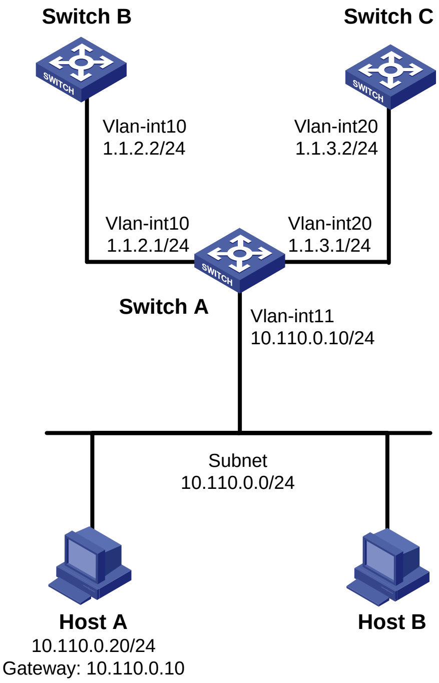
Example: Configuring packet type-based interface PBR

Network configuration

As shown in Figure 2, Switch B and Switch C do not have a route to reach each other.

Configure PBR on Switch A to forward all TCP packets received on VLAN-interface 11 to the next hop 1.1.2.2 (Switch B).

Figure 2 Network diagram



Procedure

1. Make sure Switch B and Switch C can reach Host A. (Details not shown.)
2. Configure Switch A:

   # Create VLAN 10 and VLAN 20.

```
<SwitchA> system-view
[SwitchA] vlan 10
[SwitchA-vlan10] quit
[SwitchA] vlan 20
[SwitchA-vlan20] quit
```

   # Configure the IP addresses of VLAN-interface 10 and VLAN-interface 20.

```
[SwitchA] interface vlan-interface 10
[SwitchA-Vlan-interface10] ip address 1.1.2.1 24
[SwitchA-Vlan-interface10] quit
[SwitchA] interface vlan-interface 20
[SwitchA-Vlan-interface20] ip address 1.1.3.1 24
[SwitchA-Vlan-interface20] quit
```

   # Configure ACL 3101 to match TCP packets.

```
[SwitchA] acl advanced 3101
[SwitchA-acl-ipv4-adv-3101] rule permit tcp
[SwitchA-acl-ipv4-adv-3101] quit
```

# Configure Node 5 for the policy **aaa** to forward TCP packets to next hop 1.1.2.2.

```
[SwitchA] policy-based-route aaa permit node 5
[SwitchA-pbr-aaa-5] if-match acl 3101
[SwitchA-pbr-aaa-5] apply next-hop 1.1.2.2
[SwitchA-pbr-aaa-5] quit
```

# Configure interface PBR by applying the policy **aaa** to VLAN-interface 11.

```
[SwitchA] interface vlan-interface 11
[SwitchA-Vlan-interface11] ip address 10.110.0.10 24
[SwitchA-Vlan-interface11] ip policy-based-route aaa
[SwitchA-Vlan-interface11] quit
```

Verifying the configuration

1. Perform telnet operations to verify that interface PBR on Switch A operates as configured to forward the matching TCP packets to the next hop 1.1.2.2 (Switch B), as follows:

   # Verify that you can telnet to Switch B from Host A successfully. (Details not shown.)

   # Verify that you cannot telnet to Switch C from Host A. (Details not shown.)

2. Verify that Switch A forwards packets other than TCP packets through VLAN-interface 20. For example, verify that you can ping Switch C from Host A. (Details not shown.)

Example: Configuring VXLAN ID-based outbound PBR

Network configuration

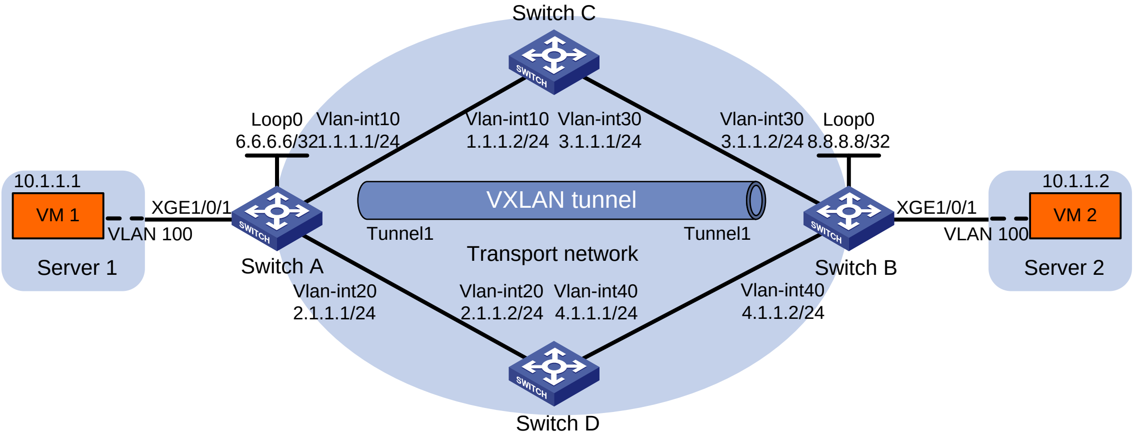
As shown in Figure 3, Switch C and Switch D are Layer 3 switches in the transport network. VXLAN 10 is configured on Switch A and Switch B (as VTEPs) to provide Layer 2 connectivity for VM 1 and VM 2 across the network sites.

- A VXLAN tunnel between the VTEPs is manually established.
- The tunnel is assigned to the VXLAN.
- Remote MAC address learning is enabled.
- The VTEPs flood VXLAN traffic in unicast mode (head-end replication).

Traffic from VM 1 to VM 2 enters the VXLAN tunnel through one of the equal-cost routes (with the next hop 2.1.1.2) between the VXLAN endpoints.

Configure PBR for Tunnel 1 on Switch A to forward the outgoing traffic through the route with the next hop 1.1.2.

Figure 3 Network diagram



Procedure

1. Configure IP addresses and unicast routing settings:

   # Assign IP address to interfaces, as shown in Figure 3. (Details not shown.)

   # Configure OSPF on all switches in the transport network. (Details not shown.)

2. Configure Switch A:

   # Enable L2VPN.

```
<SwitchA> system-view
[SwitchA] l2vpn enable
```

   # Enable Layer 2 forwarding for VXLANs.

```
[SwitchA] undo vxlan ip-forwarding
```

   # Create the VSI **vpna** and VXLAN 10.

```
[SwitchA] vsi vpna
[SwitchA-vsi-vpna] vxlan 10
[SwitchA-vsi-vpna-vxlan-10] quit
[SwitchA-vsi-vpna] quit
```

   # Assign an IP address to Loopback 0. The IP address will be used as the source IP address of the VXLAN tunnel.

```
[SwitchA] interface loopback 0
[SwitchA-Loopback0] ip address 6.6.6.6 255.255.255.255
[SwitchA-Loopback0] quit
```

   # Create a VXLAN tunnel to Switch B. The tunnel interface is Tunnel 1. The tunnel destination is Loopback 0 at 8.8.8.8 on Switch B.

```
[SwitchA] interface tunnel 1 mode vxlan
[SwitchA-Tunnel1] source 6.6.6.6
[SwitchA-Tunnel1] destination 8.8.8.8
[SwitchA-Tunnel1] quit
```

   # Assign Tunnel 1 to VXLAN 10.

```
[SwitchA] vsi vpna
[SwitchA-vsi-vpna] vxlan 10
[SwitchA-vsi-vpna-vxlan-10] tunnel 1
[SwitchA-vsi-vpna-vxlan-10] quit
[SwitchA-vsi-vpna] quit
```

   # Create Ethernet service instance 1000 on Ten-GigabitEthernet 1/0/1 to match frames from VLAN 10.

```
[SwitchA] interface ten-gigabitethernet 1/0/1
[SwitchA-Ten-GigabitEthernet1/0/1] service-instance 1000
[SwitchA-Ten-GigabitEthernet1/0/1-srv1000] encapsulation s-vid 10
```

   # Map Ethernet service instance 1000 to the VSI **vpna**.

```
[SwitchA-Ten-GigabitEthernet1/0/1-srv1000] xconnect vsi vpna
[SwitchA-Ten-GigabitEthernet1/0/1-srv1000] quit
[SwitchA-Ten-GigabitEthernet1/0/1] quit
```

   # Configure node 5 for the policy **aaa**, and configure a VXLAN match criterion for the policy to forward packets with the VXLAN ID 10 to 1.1.1.2.

```
[SwitchA] policy-based-route aaa permit node 5
[SwitchA-pbr-aaa-5] if-match vxlan 10
[SwitchA-pbr-aaa-5] apply next-hop 1.1.1.2
[SwitchA-pbr-aaa-5] quit
```

# Apply the policy to Tunnel 1 to guide the forwarding of outgoing packets.

```
[SwitchA] interface tunnel 1
[SwitchA-Tunnel1] ip policy-based-route aaa egress
[SwitchA-Tunnel1] quit
```

3. Configure Switch B:

# Enable L2VPN.

```
<SwitchB> system-view
[SwitchB] l2vpn enable
```

# Enable Layer 2 forwarding for VXLANs.

```
[SwitchB] undo vxlan ip-forwarding
```

# Create the VSI **vpna** and VXLAN 10.

```
[SwitchB] vsi vpna
[SwitchB-vsi-vpna] vxlan 10
[SwitchB-vsi-vpna-vxlan-10] quit
[SwitchB-vsi-vpna] quit
```

# Assign an IP address to Loopback 0. The IP address will be used as the source IP address of the VXLAN tunnel.

```
[SwitchB] interface loopback 0
[SwitchB-Loopback0] ip address 8.8.8.8 255.255.255.255
[SwitchB-Loopback0] quit
```

# Create a VXLAN tunnel to Switch A. The tunnel interface is Tunnel 1. The tunnel destination is Loopback 0 at 6.6.6.6 on Switch A.

```
[SwitchB] interface tunnel 1 mode vxlan
[SwitchB-Tunnel1] source 8.8.8.8
[SwitchB-Tunnel1] destination 6.6.6.6
[SwitchB-Tunnel1] quit
```

# Assign Tunnel 1 to VXLAN 10.

```
[SwitchB] vsi vpna
[SwitchB-vsi-vpna] vxlan 10
[SwitchB-vsi-vpna-vxlan-10] tunnel 1
[SwitchB-vsi-vpna-vxlan-10] quit
[SwitchB-vsi-vpna] quit
```

# Create Ethernet service instance 1000 on Ten-GigabitEthernet 1/0/1 to match frames from VLAN 10.

```
[SwitchB] interface ten-gigabitethernet 1/0/1
[SwitchB-Ten-GigabitEthernet1/0/1] service-instance 1000
[SwitchB-Ten-GigabitEthernet1/0/1-srv1000] encapsulation s-vid 10
```

# Map Ethernet service instance 1000 to the VSI **vpna**.

```
[SwitchB-Ten-GigabitEthernet1/0/1-srv1000] xconnect vsi vpna
[SwitchB-Ten-GigabitEthernet1/0/1-srv1000] quit
[SwitchB-Ten-GigabitEthernet1/0/1] quit
```

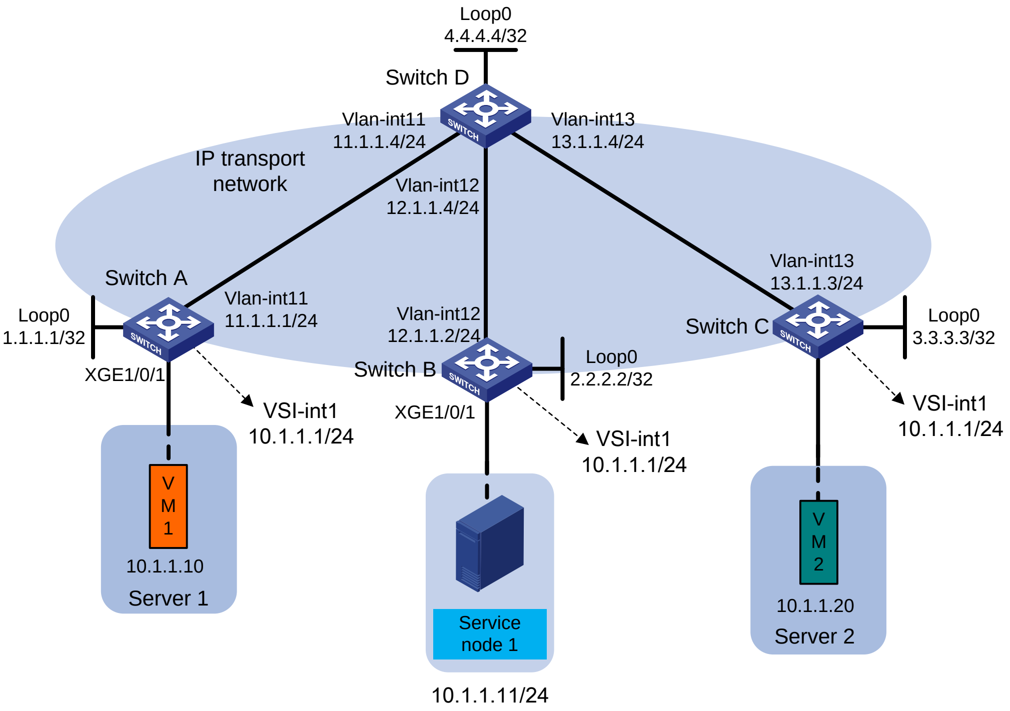
Example: Configuring EVPN-based service chain PBR

Network configuration

As shown in Figure 4, Switch A, Switch B, and Switch C are distributed EVPN gateway devices. Switch D acts as a route reflector to reflect BGP routes for the other switches.

Configure PBR to direct packets sent by Server 1 to Service node 1. After being processed, the packets are forwarded to Server 2.

Figure 4 Network diagram



Procedure

1. Configure IP addresses and subnet masks for interfaces, as shown in Figure 4. (Details not shown.)
2. Configure Switch A:

# Enable L2VPN.

```
<SwitchA> system-view
[SwitchA] l2vpn enable
```

# Disable remote-MAC address learning and remote ARP learning.

```
[SwitchA] vxlan tunnel mac-learning disable
[SwitchA] vxlan tunnel arp-learning disable
```

# Create an EVPN instance in VSI instance view, and configure the system to automatically generate an RT and RD.

```
[SwitchA] vsi vpna
[SwitchA-vsi-vpna] evpn encapsulation vxlan
[SwitchA-vsi-vpna-evpn-vxlan] route-distinguisher auto
[SwitchA-vsi-vpna-evpn-vxlan] vpn-target auto
[SwitchA-vsi-vpna-evpn-vxlan] quit
```

# Create VXLAN 10.

```
[SwitchA-vsi-vpna] vxlan 10
[SwitchA-vsi-vpna-vxlan-10] quit
[SwitchA-vsi-vpna] quit
```

# Configure BGP to advertise EVPN routes.

```
[SwitchA] bgp 200
[SwitchA-bgp-default] peer 4.4.4.4 as-number 200
```

```
[SwitchA-bgp-default] peer 4.4.4.4 connect-interface loopback 0
[SwitchA-bgp-default] address-family l2vpn evpn
[SwitchA-bgp-default-evpn] peer 4.4.4.4 enable
[SwitchA-bgp-default-evpn] quit
[SwitchA-bgp-default] quit
```

# Create VPN instance **vpna**.

```
[SwitchA] ip vpn-instance vpna
[SwitchA-vpn-instance-vpna] route-distinguisher 1:1
[SwitchA-vpn-instance-vpna] address-family ipv4
[SwitchA-vpn-ipv4-vpna] vpn-target 2:2
[SwitchA-vpn-ipv4-vpna] quit
[SwitchA-vpn-instance-vpna] address-family evpn
[SwitchA-vpn-evpn-vpna] vpn-target 1:1
[SwitchA-vpn-evpn-vpna] quit
[SwitchA-vpn-instance-vpna] quit
```

# Configure VSI-interface 1.

```
[SwitchA] interface vsi-interface 1
[SwitchA-Vsi-interface1] ip binding vpn-instance vpna
[SwitchA-Vsi-interface1] ip address 10.1.1.1 255.255.255.0
[SwitchA-Vsi-interface1] mac-address 0001-0001-0001
[SwitchA-Vsi-interface1] local-proxy-arp enable
[SwitchA-Vsi-interface1] distributed-gateway local
[SwitchA-Vsi-interface1] quit
```

# Configure VSI-interface 3, associate the interface with VPN instance **vpna**, and set the L3 VXLAN ID to 1000.

```
[SwitchA] interface vsi-interface 3
[SwitchA-Vsi-interface3] ip binding vpn-instance vpna
[SwitchA-Vsi-interface3] l3-vni 1000
[SwitchA-Vsi-interface3] quit
```

# Associate VSI instance **vpna** with VSI-interface 1.

```
[SwitchA] vsi vpna
[SwitchA-vsi-vpna] gateway vsi-interface 1
[SwitchA-vsi-vpna] quit
```

# Configure VLAN-interface 11.

```
[SwitchA] interface vlan-interface 11
[SwitchA-Vlan-interface11] ip address 11.1.1.1 255.255.255.0
[SwitchA-Vlan-interface11] ospf 1 area 0.0.0.0
[SwitchA-Vlan-interface11] quit
```

# Associate Ethernet service instance 1000 with VSI instance **vpna**.

```
[SwitchA] interface ten-gigabitethernet 1/0/1
[SwitchA-Ten-GigabitEthernet1/0/1] port link-mode bridge
[SwitchA-Ten-GigabitEthernet1/0/1] service-instance 1000
[SwitchA-Ten-GigabitEthernet1/0/1-srv1000] encapsulation s-vid 2
[SwitchA-Ten-GigabitEthernet1/0/1-srv1000] xconnect vsi vpna
```

# Create ACL 3000 to permit packets with source IP address 10.1.1.10 and destination IP address 10.1.1.20.

```
<SwitchA> system-view
[SwitchA] acl advanced 3000
```

```
[SwitchA-acl-ipv4-adv-3000] rule 0 permit ip source 10.1.1.10 0 destination 10.1.1.20
```

# Create node 0 that uses ACL 3000 to filter packets. Apply next hop 10.1.1.11 and service chain path ID 1 to packets with source IP address 10.1.1.1 and destination IP address 10.1.1.20.

```
[SwitchA] policy-based-route aa permit node 0
[SwitchA-pbr-aa-0] if-match acl 3000
[SwitchA-pbr-aa-0] apply service-chain path-id 1
[SwitchA-pbr-aa-0] apply next-hop vpn-instance vpna 10.1.1.11
```

# Apply policy **aa** to VSI-interface 3.

```
[SwitchA] interface vsi-interface 3
[SwitchA-Vsi-interface3] ip policy-based-route aa
[SwitchA-Vsi-interface3] quit
```

3. Configure Switch B:

# Enable L2VPN.

```
<SwitchB> system-view
[SwitchB] l2vpn enable
```

# Disable remote-MAC address learning and remote ARP learning.

```
[SwitchB] vxlan tunnel mac-learning disable
[SwitchB] vxlan tunnel arp-learning disable
```

# Create an EVPN instance in VSI instance view, and configure the system to automatically generate an RT and RD.

```
[SwitchB] vsi vpna
[SwitchB-vsi-vpna] evpn encapsulation vxlan
[SwitchB-vsi-vpna-evpn-vxlan] route-distinguisher auto
[SwitchB-vsi-vpna-evpn-vxlan] vpn-target auto
[SwitchB-vsi-vpna-evpn-vxlan] quit
```

# Configure VXLAN 10.

```
[SwitchB-vsi-vpna] vxlan 10
[SwitchB-vsi-vpna-vxlan-10] quit
[SwitchB-vsi-vpna] quit
```

# Configure BGP to advertise EVPN routes.

```
[SwitchB] bgp 200
[SwitchB-bgp-default] peer 4.4.4.4 as-number 200
[SwitchB-bgp-default] peer 4.4.4.4 connect-interface loopback0
[SwitchB-bgp-default] address-family l2vpn evpn
[SwitchB-bgp-default-evpn] peer 4.4.4.4 enable
```

# Create VPN instance **vpna**.

```
[SwitchB] ip vpn-instance vpna
[SwitchB-vpn-instance-vpna] route-distinguisher 1:1
[SwitchB-vpn-instance-vpna] address-family ipv4
[SwitchB-vpn-ipv4-vpna] vpn-target 2:2
[SwitchB-vpn-ipv4-vpna] quit
[SwitchB-vpn-instance-vpna] address-family evpn
[SwitchB-vpn-evpn-vpna] vpn-target 1:1
[SwitchB-vpn-evpn-vpna] quit
[SwitchB-vpn-instance-vpna] quit
```

# Configure VSI-interface 1.

```
[SwitchB] interface vsi-interface 1
```

```
[SwitchB-Vsi-interface1] ip binding vpn-instance vpna
[SwitchB-Vsi-interface1] ip address 10.1.1.1 255.255.255.0
[SwitchB-Vsi-interface1] mac-address 0001-0001-0001
[SwitchB-Vsi-interface1] local-proxy-arp enable
[SwitchB-Vsi-interface1] distributed-gateway local
[SwitchB-Vsi-interface1] quit
```

# Associate VSI instance **vpna** with VSI-interface 1.

```
[SwitchB] vsi vpna
[SwitchB-vsi-vpna] gateway vsi-interface 1
[SwitchB-vsi-vpna] quit
```

# Configure VSI-interface 3.

```
[SwitchB] interface vsi-interface 3
[SwitchB-Vsi-interface3] ip binding vpn-instance vpna
[SwitchB-Vsi-interface3] l3-vni 1000
[SwitchB-Vsi-interface3] quit
```

# Configure Ten-GigabitEthernet 1/0/1 as an AC interface.

```
[SwitchB] interface ten-gigabitethernet 1/0/1
[SwitchB-Ten-GigabitEthernet1/0/1] port link-mode bridge
[SwitchB-Ten-GigabitEthernet1/0/1] service-instance 1000
[SwitchB-Ten-GigabitEthernet1/0/1-srv1000] encapsulation s-vid 2
[SwitchB-Ten-GigabitEthernet1/0/1-srv1000] xconnect vsi vpna
[SwitchB-Ten-GigabitEthernet1/0/1-srv1000] quit
[SwitchB-Ten-GigabitEthernet1/0/1] quit
```

# Create node 0 and apply next hop 10.1.1.11 to packets with service chain path ID 1.

```
[SwitchB] policy-based-route aa permit node 0
[SwitchB-pbr-aa-0] if-match service-chain path-id 1
[SwitchB-pbr-aa-0] apply next-hop vpn-instance vpna 10.1.1.11
[SwitchB-pbr-aa-0] quit
```

# Apply policy **aa** to VSI-interface 3.

```
[SwitchB] interface vsi-interface 3
[SwitchB-Vsi-interface3] ip policy-based-route aa
[SwitchB-Vsi-interface3] quit
```

4. Configure Switch C:

# Enable L2VPN.

```
<SwitchC> system-view
[SwitchC] l2vpn enable
```

# Disable remote-MAC address learning and remote ARP learning.

```
[SwitchC] vxlan tunnel mac-learning disable
[SwitchC] vxlan tunnel arp-learning disable
```

# Create an EVPN instance in VSI instance view, and configure the system to automatically generate an RT and RD.

```
[SwitchC] vsi vpna
[SwitchC-vsi-vpna] evpn encapsulation vxlan
[SwitchC-vsi-vpna-evpn-vxlan] route-distinguisher auto
[SwitchC-vsi-vpna-evpn-vxlan] vpn-target auto
[SwitchC-vsi-vpna-evpn-vxlan] quit
```

# Configure VXLAN 10.

```
[SwitchC-vsi-vpna] vxlan 10
```

```
[SwitchC-vsi-vpna-vxlan-10] quit
[SwitchC-vsi-vpna] quit
```

# Configure BGP to advertise EVPN routes.

```
[SwitchC] bgp 200
[SwitchC-bgp-default] peer 4.4.4.4 as-number 200
[SwitchC-bgp-default] peer 4.4.4.4 connect-interface loopback 0
[SwitchC-bgp-default] address-family l2vpn evpn
[SwitchC-bgp-default-evpn] peer 4.4.4.4 enable
[SwitchC-bgp-default-evpn] quit
[SwitchC-bgp-default] quit
```

# Create VPN instance **vpna**.

```
[SwitchC] ip vpn-instance vpna
[SwitchC-vpn-instance-vpna] route-distinguisher 1:1
[SwitchC-vpn-instance-vpna] address-family ipv4
[SwitchC-vpn-ipv4-vpna] vpn-target 2:2
[SwitchC-vpn-ipv4-vpna] quit
[SwitchC-vpn-instance-vpna] address-family evpn
[SwitchC-vpn-evpn-vpna] vpn-target 1:1
[SwitchC-vpn-evpn-vpna] quit
[SwitchC-vpn-instance-vpna] quit
```

# Create VSI-interface 1, assign an IP address to it, and specify the interface as a distributed gateway in VXLAN 10.

```
[SwitchC] interface vsi-interface 1
[SwitchC-Vsi-interface1] ip binding vpn-instance vpna
[SwitchC-Vsi-interface1] ip address 10.1.1.1 255.255.255.0
[SwitchC-Vsi-interface1] mac-address 0001-0001-0001
[SwitchC-Vsi-interface1] local-proxy-arp enable
[SwitchC-Vsi-interface1] distributed-gateway local
[SwitchC-Vsi-interface1] quit
```

# Configure VSI-interface 3, associate the interface with VPN instance **vpna**, and set the L3 VXLAN ID to 1000.

```
[SwitchC] interface vsi-interface 3
[SwitchC-Vsi-interface3] ip binding vpn-instance vpna
[SwitchC-Vsi-interface3] l3-vni 1000
[SwitchC-Vsi-interface3] quit
```

# Associate VSI instance **vpna** with VSI-interface 1.

```
[SwitchC] vsi vpna
[SwitchC-vsi-vpna] gateway vsi-interface 1
[SwitchC-vsi-vpna] quit
```

# Bind VSI instance **vpna** to Ten-GigabitEthernet 1/0/1.

```
[SwitchC] interface ten-gigabitethernet 1/0/1
[SwitchC-Ten-GigabitEthernet1/0/1] port link-mode bridge
[SwitchC-Ten-GigabitEthernet1/0/1] service-instance 2000
[SwitchC-Ten-GigabitEthernet1/0/1-srv2000] encapsulation s-vid 2
[SwitchC-Ten-GigabitEthernet1/0/1] xconnect vsi vpna
[SwitchC-Ten-GigabitEthernet1/0/1] quit
```

5. Configure Switch D:

# Configure Switch D to establish BGP connections with the other switches.

```
<SwitchD> system-view
[SwitchD] bgp 200
[SwitchD-bgp-default] group evpn
[SwitchD-bgp-default] peer 1.1.1.1 group evpn
[SwitchD-bgp-default] peer 2.2.2.2 group evpn
[SwitchD-bgp-default] peer 3.3.3.3 group evpn
[SwitchD-bgp-default] peer evpn as-number 200
[SwitchD-bgp-default] peer evpn connect-interface loopback 0
```

# Configure BGP to advertise EVPN routes, and disable route target filtering for BGP EVPN routes.

```
[SwitchD-bgp-default] address-family l2vpn evpn
[SwitchD-bgp-default-evpn] peer evpn enable
[SwitchD-bgp-default-evpn] undo policy vpn-target
```

# Configure Switch D as a route reflector.

```
[SwitchD-bgp-default-evpn] peer evpn reflect-client
[SwitchD-bgp-default-evpn] quit
[SwitchD-bgp-default] quit
```

# Configure VLAN-interface 11.

```
[SwitchD] interface vlan-interface 11
[SwitchD-Vlan-interface11] ip address 11.1.1.4 255.255.255.0
[SwitchD-Vlan-interface11] ospf 1 area 0.0.0.0
[SwitchD-Vlan-interface11] quit
```

# Configure VLAN-interface 12.

```
[SwitchD] interface vlan-interface 12
[SwitchD-Vlan-interface12] ip address 12.1.1.4 255.255.255.0
[SwitchD-Vlan-interface12] ospf 1 area 0.0.0.0
[SwitchD-Vlan-interface12] quit
```

# Configure VLAN-interface 13.

```
[SwitchD] interface Vlan-interface 13
[SwitchD-Vlan-interface13] ip address 13.1.1.4 255.255.255.0
[SwitchD-Vlan-interface13] ospf 1 area 0.0.0.0
[SwitchD-Vlan-interface13] quit
```

Verifying the configuration

# Capture packets sent from Server 1 to Server 2 in Ethernet service instance 1000. (Details not shown.)

The packets are processed by Service node 1 before they are delivered to Server 2.