Contents

Configuring MCE

About MCE

The Multi-VPN Instance Customer Edge (MCE) feature allows multiple VPNs to share a CE with ensured data security in an MPLS L3VPN network. MCE provides traffic separation between VPNs by distinguishing routes of the VPNs.

MPLS L3VPN problems solved by MCE

MPLS L3VPN is a L3VPN technology used to interconnect geographically dispersed VPN sites. MPLS L3VPN uses BGP to advertise VPN routes and uses MPLS to forward VPN packets over a service provider backbone. MPLS L3VPN provides flexible networking modes, excellent scalability, and convenient support for MPLS TE.

In conventional MPLS L3VPN, a VPN site must exclusively use a CE to connect a PE. With increasing business, a private network must be divided into multiple VPNs and the VPNs must be isolated from one another for data security. Using a CE exclusively for each VPN costs much in buying and maintaining the network devices. Using a common CE for multiple VPNs cannot ensure data security because the VPNs use the same routing table.

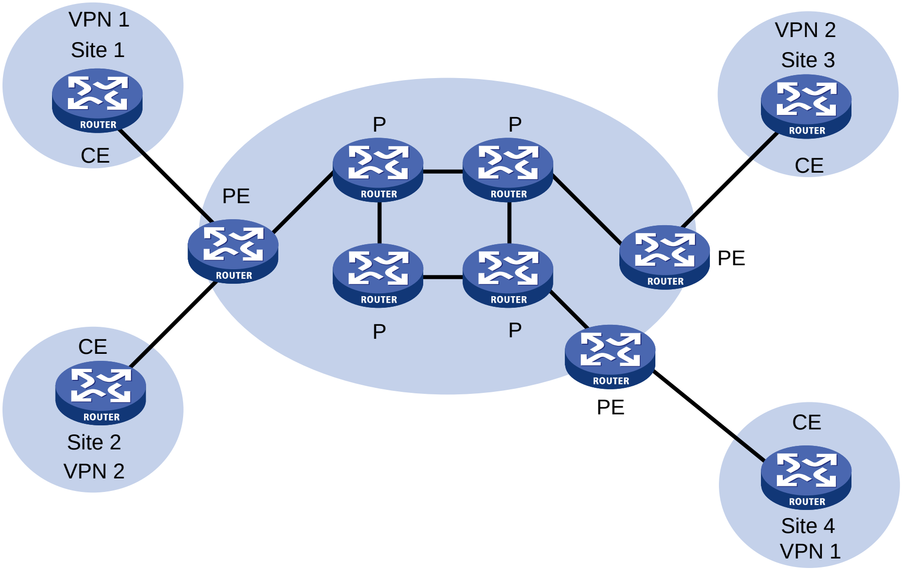
The MCE feature offers balance between cost and security. It creates multiple VPN instances on a CE to provide logically separated routing tables and address spaces for the VPNs, so the VPNs can share the CE. This CE is called the MCE device.

Basic MPLS L3VPN architecture

A basic MPLS L3VPN architecture has the following types of devices:

- **Customer edge device**—A CE device resides on a customer network and has one or more interfaces directly connected to a service provider network. It does not support MPLS.
- **Provider edge device**—A PE device resides at the edge of a service provider network and is connected to one or more CEs. All MPLS VPN services are processed on PEs.
- **Provider device**—A P device is a core device on a service provider network. It is not directly connected to any CEs. A P device has only basic MPLS forwarding capability and does not handle VPN routing information.

Figure 1 Basic MPLS L3VPN architecture



MCE-related MPLS L3VPN concepts

Site

A site has the following features:

- A site is a group of IP systems with IP connectivity that does not rely on any service provider networks.
- The classification of a site depends on the topology relationship of the devices, rather than the geographical positions. However, the devices at a site are, in most cases, adjacent to each other geographically.
- The devices at a site can belong to multiple VPNs, which means that a site can belong to multiple VPNs.
- A site is connected to a provider network through one or more CEs. A site can contain multiple CEs, but a CE can belong to only one site.

Sites connected to the same provider network can be classified into different sets by policies. Only the sites in the same set can access each other through the provider network. Such a set is called a VPN.

VPN instance

VPN instances implement route isolation, data independence, and data security for VPNs.

A VPN instance has the following components:

- A separate Label Forwarding Information Base (LFIB).
- An IP routing table.
- Interfaces bound to the VPN instance.
- VPN instance administration information, including route distinguishers (RDs), route targets (RTs), and route filtering policies.

To associate a site with a VPN instance, bind the VPN instance to the PE's interface connected to the site. A site can be associated with only one VPN instance, and different sites can be associated with the same VPN instance. A VPN instance contains the VPN membership and routing rules of associated sites.

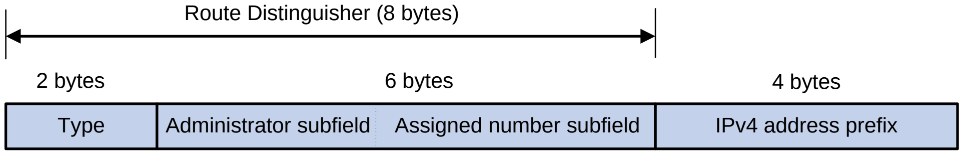
VPN-IPv4 address

Each VPN independently manages its address space. The address spaces of VPNs might overlap. For example, if both VPN 1 and VPN 2 use the addresses on subnet 10.110.10.0/24, address space overlapping occurs.

Multiprotocol BGP (MP-BGP) can solve this problem by advertising VPN-IPv4 addresses (also called VPNv4 addresses).

As shown in Figure 2, a VPN-IPv4 address consists of 12 bytes. The first eight bytes represent the RD, followed by a four-byte IPv4 prefix. The RD and the IPv4 prefix form a unique VPN-IPv4 prefix.

Figure 2 VPN-IPv4 address structure



An RD can be in one of the following formats:

- When the Type field is 0, the Administrator subfield occupies two bytes, the Assigned number subfield occupies four bytes, and the RD format is *16-bit AS number*:*32-bit user-defined number*. For example, 100:1.
- When the Type field is 1, the Administrator subfield occupies four bytes, the Assigned number subfield occupies two bytes, and the RD format is *32-bit IPv4 address*:*16-bit user-defined number*. For example, 172.1.1.1:1.
- When the Type field is 2, the Administrator subfield occupies four bytes, the Assigned number subfield occupies two bytes, and the RD format is *32-bit AS number*:*16-bit user-defined number*, where the minimum value of the AS number is 65536. For example, 65536:1.

To guarantee global uniqueness for a VPN-IPv4 address, do not set the Administrator subfield to any private AS number or private IP address.

Route target attribute

MPLS L3VPN uses route target community attributes to control the advertisement of VPN routing information. A VPN instance on a PE supports the following types of route target attributes:

- **Export target attribute**—A PE sets the export target attribute for VPN-IPv4 routes learned from directly connected sites before advertising them to other PEs.
- **Import target attribute**—A PE checks the export target attribute of VPN-IPv4 routes received from other PEs. If the export target attribute matches the import target attribute of a VPN instance, the PE adds the routes to the routing table of the VPN instance.

Route target attributes define which sites can receive VPN-IPv4 routes, and from which sites a PE can receive routes.

Like RDs, route target attributes can be one of the following formats:

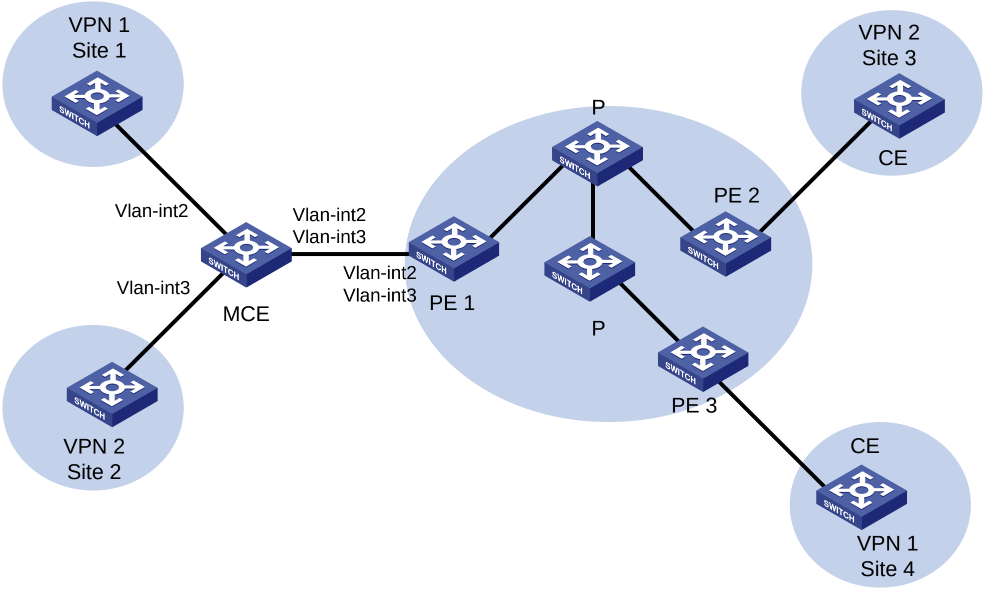
- *16-bit AS number*:*32-bit user-defined number*. For example, 100:1.
- *32-bit IPv4 address:16-bit user-defined number*. For example, 172.1.1.1:1.
- *32-bit AS number*:*16-bit user-defined number*, where the minimum value of the AS number is 65536. For example, 65536:1.

MCE working mechanism

As shown in Figure 3, the MCE exchanges private routes with VPN sites and PE 1, and adds the private routes to the routing tables of corresponding VPN instances.

- **Route exchange between MCE and VPN site**—Create VPN instances VPN 1 and VPN 2 on the MCE. Bind VLAN-interface 2 to VPN 1, and VLAN-interface 3 to VPN 2. The MCE adds a received route to the routing table of the VPN instance that is bound to the receiving interface.
- **Route exchange between MCE and PE**—The MCE connects to PE 1 through trunk links. On PE 1, create VPN instances for VPN 1 and VPN 2. Bind VLAN-interface 2 to VPN 1, and VLAN-interface 3 to VPN 2. The MCE and PE add a received route to the routing table of the VPN instance that is bound to the receiving interface.

Figure 3 Network diagram for the MCE feature



You can configure static routes, RIP, OSPF, IS-IS, EBGP, or IBGP between an MCE and a VPN site and between an MCE and a PE.

NOTE:

To implement dynamic IP assignment for DHCP clients in private networks, you can configure DHCP server or DHCP relay agent on the MCE. When the MCE functions as the DHCP server, the IP addresses assigned to different private networks cannot overlap.

Restrictions and guidelines: MCE configuration

On the PE, disable routing loop detection to avoid route loss during route calculation, and disable route redistribution between routing protocols to save system resources.

MCE tasks at a glance

To configure MCE, perform the following tasks:

1. Configuring VPN instances

   Perform the following VPN instance tasks on PEs and MCEs:

   a. Creating a VPN instance
   b. Associating a VPN instance with an interface
   c. (Optional.) Configuring route related attributes for a VPN instance
2. Configuring routing between an MCE and a VPN site

3. Configuring routing between an MCE and a PE

Configuring VPN instances

Creating a VPN instance

About VPN instances

A VPN instance is a collection of the VPN membership and routing rules of its associated site. A VPN instance might correspond to more than one VPN.

Procedure

1. Enter system view.

   system-view

2. Create a VPN instance and enter VPN instance view.

   ip vpn-instance *vpn-instance-name*

3. Configure an RD for the VPN instance.

   route-distinguisher *route-distinguisher*

   By default, no RD is configured for a VPN instance.

4. (Optional.) Configure a description for the VPN instance.

   description *text*

   By default, no description is configured for a VPN instance.

5. (Optional.) Configure a VPN ID for the VPN instance.

   vpn-id *vpn-id*

   By default, no VPN ID is configured for a VPN instance.

Associating a VPN instance with an interface

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

   This interface connects the device to the CE.

3. Associate a VPN instance with the interface.

   ip binding vpn-instance *vpn-instance-name*

   By default, an interface is not associated with a VPN instance and it belongs to the public network.

   Configuration of this command deletes the IP address of the current interface. You must reconfigure an IP address for the interface after configuring the command.

Configuring route related attributes for a VPN instance

Restrictions and guidelines

If you configure route related attributes in both VPN instance view and VPN instance IPv4 VPN view, IPv4 VPN uses the configuration in VPN instance IPv4 VPN view.

Prerequisites

Before you perform this task, create the routing policies to be used by this task. For information about routing policies, see *Layer 3—IP Routing Configuration Guide*.

Procedure

1. Enter system view.

   system-view

2. Enter VPN instance view or VPN instance IPv4 VPN view.
   - Enter VPN instance view.

     ip vpn-instance *vpn-instance-name*
   - Execute the following commands in sequence to enter VPN instance IPv4 VPN view:

     ip vpn-instance *vpn-instance-name*

     address-family ipv4

3. Configure route targets.

   vpn-target *vpn-target*&<1-8> [**both** | **export-extcommunity** | **import-extcommunity**]

   By default, no route targets are configured.

4. Set the maximum number of active routes.

   routing-table limit *number* { *warn-threshold* | **simply-alert** }

   By default, the number of active routes in a VPN instance is not limited.

   Setting the maximum number of active routes for a VPN instance can prevent the device from learning too many routes.

5. Apply an import routing policy.

   import route-policy *route-policy*

   By default, all routes matching the import target attribute are accepted.

6. Apply an export routing policy.

   export route-policy *route-policy*

   By default, routes to be advertised are not filtered.

Configuring routing between an MCE and a VPN site

Configuring static routing between an MCE and a VPN site

About static routing between an MCE and a VPN site

An MCE can reach a VPN site through a static route. Static routing on a traditional CE is globally effective and does not support address overlapping among VPNs. An MCE supports binding a static route to a VPN instance, so that the static routes of different VPN instances can be isolated from each other.

Perform this configuration on the MCE. On the VPN site, configure a common static route.

Procedure

1. Enter system view.

   system-view

2. Configure a static route for a VPN instance.

```
ip route-static vpn-instance s-vpn-instance-name dest-address
{ mask-length | mask } { interface-type interface-number
[ next-hop-address ] | next-hop-address [ public ] | vpn-instance
d-vpn-instance-name next-hop-address }
```

3. (Optional.) Configure the default preference for static routes.

   ```
   ip route-static default-preference default-preference
   ```

   The default preference is 60.

Configuring RIP between an MCE and a VPN site

About RIP between an MCE and a VPN site

Binding RIP processes to VPN instances can isolate routes of different VPNs. For more information about RIP, see *Layer 3—IP Routing Configuration Guide*.

Perform this configuration on the MCE. On a VPN site, create a common RIP process.

Procedure

1. Enter system view.

   ```
   system-view
   ```

2. Create a RIP process for a VPN instance and enter RIP view.

   ```
   rip [ process-id ] vpn-instance vpn-instance-name
   ```

   A RIP process can belong to only one VPN instance.

3. Enable RIP on the interface attached to the specified network.

   ```
   network network-address [ wildcard-mask ]
   ```

   By default, RIP is disabled on an interface.

4. Redistribute remote site routes advertised by the PE into RIP.

   ```
   import-route protocol [ as-number ] [ process-id | all-processes |
   allow-ibgp ] [ allow-direct | cost cost-value | route-policy
   route-policy-name | tag tag ] *
   ```

   By default, no route is redistributed into RIP.

Configuring OSPF between an MCE and a VPN site

About OSPF between an MCE and a VPN site

Binding OSPF processes to VPN instances can isolate routes of different VPNs. For more information about OSPF, see *Layer 3—IP Routing Configuration Guide*.

Perform this configuration on the MCE. On a VPN site, create a common OSPF process.

Procedure

1. Enter system view.

   ```
   system-view
   ```

2. Create an OSPF process for a VPN instance and enter OSPF view.

   ```
   ospf [ process-id ] router-id router-id vpn-instance vpn-instance-name
   ```

Keywords	Usage guidelines
router-id *router-id*	An OSPF process that is bound to a VPN instance does not use the public network router ID configured in system view. Therefore, you must specify a router ID when creating a process or configure an IP address for a minimum of one interface in the VPN instance.

vpn-instance *vpn-instance-name*	An OSPF process can belong to only one VPN instance. If you delete a VPN instance, all OSPF processes of the VPN instance are also deleted.

3. Redistribute remote site routes advertised by the PE into OSPF.

   import-route *protocol* [*as-number*] [*process-id* | **all-processes** | **allow-ibgp**] [**allow-direct** | **cost** *cost-value* | **nssa-only** | **route-policy** *route-policy-name* | **tag** *tag* | **type** *type*] *

   By default, no routes are redistributed into OSPF.

4. Create an OSPF area and enter OSPF area view.

   area *area-id*

5. Enable OSPF on the interface attached to the specified network in the area.

   network *ip-address wildcard-mask*

   By default, an interface neither belongs to any area nor runs OSPF.

Configuring IS-IS between an MCE and a VPN site

About IS-IS between an MCE and a VPN site

Binding IS-IS processes to VPN instances can isolate routes of different VPNs. For more information about IS-IS, see *Layer 3—IP Routing Configuration Guide*.

Perform this configuration on the MCE. On a VPN site, configure a common IS-IS process.

Procedure

1. Enter system view.

   system-view

2. Create an IS-IS process for a VPN instance and enter IS-IS view.

   isis [*process-id*] **vpn-instance** *vpn-instance-name*

   An IS-IS process can belong to only one VPN instance.

3. Configure a network entity title.

   network-entity *net*

   By default, no NET is configured.

4. Create the IS-IS IPv4 unicast address family and enter its view.

   address-family ipv4 [**unicast**]

5. Redistribute remote site routes advertised by the PE into IS-IS.

   import-route *protocol* [*as-number*] [*process-id* | **all-processes** | **allow-ibgp**] [**allow-direct** | **cost** *cost-value* | **cost-type** { **external** | **internal** } | [**level-1** | **level-1-2** | **level-2**] | **route-policy** *route-policy-name* | **tag** *tag*] *

   By default, IS-IS does not redistribute routes from any other routing protocol.

   If you do not specify the route level in the command, the command redistributes routes to the level-2 routing table by default.

6. Return to system view.

   quit

7. Enter interface view.

   interface *interface-type interface-number*

8. Enable the IS-IS process on the interface.

```
isis enable [ process-id ]
```

By default, no IS-IS process is enabled on the interface.

Configuring EBGP between an MCE and a VPN site

About EBGP between an MCE and a VPN site

To run EBGP between an MCE and a VPN site, you must configure a BGP peer for each VPN instance on the MCE, and redistribute the IGP routes of each VPN instance on the VPN site.

Restrictions and guidelines

You must configure a VPN site to advertise the VPN network addresses it can reach to the connected MCE.

Configuring the MCE

1. Enter system view.

   ```
   system-view
   ```

2. Enter BGP instance view.

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

   By default, BGP is not enabled.

3. Enter BGP-VPN instance view.

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

   Commands in BGP-VPN instance view are the same as those in BGP instance view. For more information, see *Layer 3—IP Routing Configuration Guide*.

4. Configure the VPN site as an EBGP peer.

   ```
   peer { group-name | ipv4-address [ mask-length ] } as-number as-number
   ```

5. Enter BGP-VPN IPv4 unicast address family view.

   ```
   address-family ipv4 [ unicast ]
   ```

6. Enable BGP to exchange IPv4 unicast routes with the specified peer.

   ```
   peer { group-name | ipv4-address [ mask-length ] } enable
   ```

   By default, BGP does not exchange IPv4 unicast routes with peers.

7. (Optional.) Allow the local AS number to appear in the AS_PATH attribute of routes received from the peer, and set the maximum number of repetitions.

   ```
   peer { group-name | ipv4-address [ mask-length ] } allow-as-loop
   [ number ]
   ```

   By default, BGP discards incoming route updates that contain the local AS number.

8. Redistribute remote site routes advertised by the PE into BGP.

   ```
   import-route protocol [ { process-id | all-processes } [ allow-direct |
   med med-value | route-policy route-policy-name ] * ]
   ```

   By default, no routes are redistributed into BGP.

Configuring the VPN site

1. Enter system view.

   ```
   system-view
   ```

2. Enter BGP instance view.

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

   By default, BGP is not enabled.

3. Configure the MCE as an EBGP peer.

```
peer { group-name | ipv4-address [ mask-length ] } as-number as-number
```

4. Enter BGP IPv4 unicast address family view.

   ```
   address-family ipv4 [ unicast ]
   ```

5. Enable BGP to exchange IPv4 unicast routes with the specified peer or peer group.

   ```
   peer { group-name | ipv4-address [ mask-length ] } enable
   ```

   By default, BGP does not exchange IPv4 unicast routes with peers or peer groups.

6. Redistribute the IGP routes of the VPN into BGP.

   ```
   import-route protocol [ { process-id | all-processes } [ allow-direct |
   med med-value | route-policy route-policy-name ] * ]
   ```

   By default, no routes are redistributed into BGP.

Configuring IBGP between an MCE and a VPN site

About IBGP between an MCE and a VPN site

To run IBGP between an MCE and a VPN site, you must configure a BGP peer for each VPN instance on the MCE, and redistribute the IGP routes of each VPN instance on the VPN site.

Restrictions and guidelines

You must configure a VPN site to advertise VPN network addresses to the connected MCE.

Configuring the MCE

1. Enter system view.

   ```
   system-view
   ```

2. Enter BGP instance view.

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

   By default, BGP is not enabled.

3. Enter BGP-VPN instance view.

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

4. Configure the VPN site as an IBGP peer.

   ```
   peer { group-name | ipv4-address [ mask-length ] } as-number as-number
   ```

5. Enter BGP-VPN IPv4 unicast address family view.

   ```
   address-family ipv4 [ unicast ]
   ```

6. Enable BGP to exchange IPv4 unicast routes with the specified peer or peer group.

   ```
   peer { group-name | ipv4-address [ mask-length ] } enable
   ```

   By default, BGP does not exchange IPv4 unicast routes with peers or peer groups.

7. (Optional.) Configure the local device as the RR, and specify the peer as the client of the RR.

   ```
   peer { group-name | ipv4-address [ mask-length ] } reflect-client
   ```

   By default, no RR or RR client is configured.

   After you configure a VPN site as an IBGP peer, the MCE does not advertise the BGP routes learned from the VPN site to other IBGP peers, including VPNv4 peers. The MCE advertises routes learned from a VPN site only when you configure the VPN site as a client of the RR (the MCE).

8. Redistribute remote site routes advertised by the PE into BGP.

   ```
   import-route protocol [ { process-id | all-processes } [ allow-direct |
   med med-value | route-policy route-policy-name ] * ]
   ```

   By default, no routes are redistributed into BGP.

Configuring the VPN site

1. Enter system view.

   system-view

2. Enter BGP instance view.

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

   By default, BGP is not enabled.

3. Configure the MCE as an IBGP peer.

   peer { *group-name* | *ipv4-address* [*mask-length*] } **as-number** *as-number*

4. Enter BGP IPv4 unicast address family view.

   address-family ipv4 [**unicast**]

5. Enable BGP to exchange IPv4 unicast routes with the specified peer.

   peer { *group-name* | *ipv4-address* [*mask-length*] } **enable**

   By default, BGP does not exchange IPv4 unicast routes with peers.

6. Redistribute the IGP routes of the VPN into BGP.

   import-route *protocol* [{ *process-id* | **all-processes** } [**allow-direct** | **med** *med-value* | **route-policy** *route-policy-name*] *]

   By default, no routes are redistributed into BGP.

Configuring routing between an MCE and a PE

About routing between an MCE and a PE

MCE-PE routing configuration includes these tasks:

- Binding the MCE-PE interfaces to VPN instances.
- Performing route configurations.
- Redistributing VPN routes into the routing protocol running between the MCE and the PE.

Perform the following configurations on the MCE. For information about configuring the PE, see the documentation for the PE.

Configuring static routing between an MCE and a PE

1. Enter system view.

   system-view

2. Configure a static route for a VPN instance.

   ip route-static vpn-instance *s-vpn-instance-name dest-address* { *mask-length* | *mask* } { *interface-type interface-number* [*next-hop-address*] | *next-hop-address* [**public**] | **vpn-instance** *d-vpn-instance-name next-hop-address* }

3. (Optional.) Set the default preference for static routes.

   ip route-static default-preference *default-preference*

   The default preference is 60.

Configuring RIP between an MCE and a PE

1. Enter system view.

system-view

2. Create a RIP process for a VPN instance and enter RIP view.

   rip [*process-id*] **vpn-instance** *vpn-instance-name*

3. Enable RIP on the interface attached to the specified network.

   network *network-address* [*wildcard-mask*]

   By default, RIP is disabled on an interface.

4. Redistribute the VPN routes.

   import-route *protocol* [*as-number*] [*process-id* | **all-processes** | **allow-ibgp**] [**allow-direct** | **cost** *cost-value* | **route-policy** *route-policy-name* | **tag** *tag*] *

   By default, no routes are redistributed into RIP.

Configuring OSPF between an MCE and a PE

1. Enter system view.

   system-view

2. Create an OSPF process for a VPN instance and enter OSPF view.

   ospf [*process-id* | **router-id** *router-id* | **vpn-instance** *vpn-instance-name*] *

3. Disable routing loop detection.

   vpn-instance-capability simple

   By default, routing loop detection is enabled and the MCE does not receive OSPF routes from the PE.

4. Redistribute the VPN routes.

   import-route *protocol* [*as-number*] [*process-id* | **all-processes** | **allow-ibgp**] [**allow-direct** | **cost** *cost-value* | **nssa-only** | **route-policy** *route-policy-name* | **tag** *tag* | **type** *type*] *

   By default, no routes are redistributed into OSPF.

5. Create an OSPF area and enter OSPF area view.

   area *area-id*

6. Enable OSPF on the interface attached to the specified network in the area.

   network *ip-address wildcard-mask*

   By default, an interface neither belongs to an area nor runs OSPF.

Configuring IS-IS between an MCE and a PE

1. Enter system view.

   system-view

2. Create an IS-IS process for a VPN instance and enter IS-IS view.

   isis [*process-id*] **vpn-instance** *vpn-instance-name*

3. Configure a network entity title.

   network-entity *net*

   By default, no NET is configured.

4. Create the IS-IS IPv4 unicast address family and enter its view.

   address-family ipv4 [**unicast**]

5. Redistribute VPN routes.

```
import-route protocol [ as-number ] [ process-id | all-processes |
allow-ibgp ] [ allow-direct | cost cost-value | cost-type { external |
internal } | [ level-1 | level-1-2 | level-2 ] | route-policy
route-policy-name | tag tag ] *
```

By default, IS-IS does not redistribute routes from any other routing protocol.

If you do not specify the route level in the command, the command redistributes routes to the level-2 routing table by default.

6. Return to system view.

```
quit
```

7. Enter interface view.

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

8. Enable the IS-IS process on the interface.

```
isis enable [ process-id ]
```

By default, no IS-IS process is enabled on the interface.

Configuring EBGP between an MCE and a PE

1. Enter system view.

```
system-view
```

2. Enter BGP instance view.

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

By default, BGP is not enabled.

3. Enter BGP-VPN instance view.

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

4. Configure the PE as an EBGP peer.

```
peer { group-name | ipv4-address [ mask-length ] } as-number as-number
```

5. Enter BGP-VPN IPv4 unicast address family view.

```
address-family ipv4 [ unicast ]
```

6. Enable BGP to exchange IPv4 unicast routes with the specified peer.

```
peer { group-name | ipv4-address [ mask-length ] } enable
```

By default, BGP does not exchange IPv4 unicast routes with peers.

7. Redistribute the VPN routes of the VPN site.

```
import-route protocol [ { process-id | all-processes } [ allow-direct |
med med-value | route-policy route-policy-name ] * ]
```

By default, no routes are redistributed into BGP.

Configuring IBGP between an MCE and a PE

1. Enter system view.

```
system-view
```

2. Enter BGP instance view.

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

By default, BGP is not enabled.

3. Enter BGP-VPN instance view.

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

4. Configure the PE as an IBGP peer.

   peer { *group-name* | *ipv4-address* [*mask-length*] } **as-number** *as-number*

5. Enter BGP-VPN IPv4 unicast address family view.

   address-family ipv4 [**unicast**]

6. Enable BGP to exchange IPv4 unicast routes with the specified peer.

   peer { *group-name* | *ipv4-address* [*mask-length*] } **enable**

   By default, BGP does not exchange IPv4 unicast routes with peers.

7. Redistribute the VPN routes of the VPN site.

   import-route *protocol* [{ *process-id* | **all-processes** } [**allow-direct** | **med** *med-value* | **route-policy** *route-policy-name*] *]

   By default, no routes are redistributed into BGP.

Display and maintenance commands for MCE

Execute **display** commands in any view.

Task	Command
Display VPN instance information.	**display ip vpn-instance** [**instance-name** *vpn-instance-name*]

For commands that display routing tables for VPN instances, see *Layer 3—IP Routing Command Reference*.

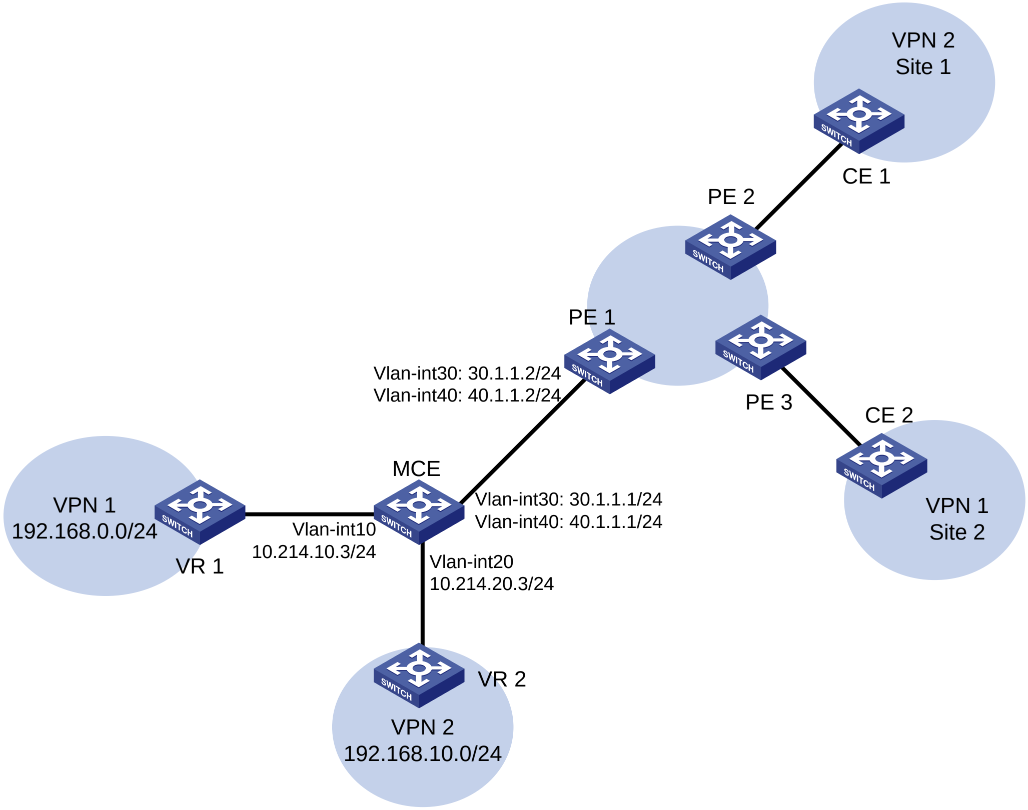
MCE configuration examples

Example: Configuring the MCE that uses OSPF to advertise VPN routes to the PE

Network configuration

As shown in Figure 4, VPN 2 runs OSPF. The edge devices of VPN 1 and VPN 2 are VR 1 and VR 2, respectively. Configure the MCE device to separate routes from different VPNs and to advertise the VPN routes to PE 1 through OSPF.

Figure 4 Network diagram



Procedure

1. Configure the VPN instances on the MCE and PE 1:

   # On the MCE, configure VPN instances **vpn1** and **vpn2**, and specify an RD and route targets for each VPN instance.

   ```
   <MCE> system-view
   [MCE] ip vpn-instance vpn1
   [MCE-vpn-instance-vpn1] route-distinguisher 10:1
   [MCE-vpn-instance-vpn1] vpn-target 10:1
   [MCE-vpn-instance-vpn1] quit
   [MCE] ip vpn-instance vpn2
   [MCE-vpn-instance-vpn2] route-distinguisher 20:1
   [MCE-vpn-instance-vpn2] vpn-target 20:1
   [MCE-vpn-instance-vpn2] quit
   ```

   # Bind VLAN-interface 10 to VPN instance **vpn1**, and configure an IP address for the VLAN interface.

   ```
   [MCE] interface vlan-interface 10
   [MCE-Vlan-interface10] ip binding vpn-instance vpn1
   [MCE-Vlan-interface10] ip address 10.214.10.3 24
   [MCE-Vlan-interface10] quit
   ```

   # Bind VLAN-interface 20 to VPN instance **vpn2**, and configure an IP address for the VLAN interface.

   ```
   [MCE] interface vlan-interface 20
   [MCE-Vlan-interface20] ip binding vpn-instance vpn2
   [MCE-Vlan-interface20] ip address 10.214.20.3 24
   ```

```
[MCE-Vlan-interface20] quit
```

# On PE 1, configure VPN instances **vpn1** and **vpn2**, and specify an RD and route targets for each VPN instance.

```
<PE1> system-view
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] route-distinguisher 10:1
[PE1-vpn-instance-vpn1] vpn-target 10:1
[PE1-vpn-instance-vpn1] quit
[PE1] ip vpn-instance vpn2
[PE1-vpn-instance-vpn2] route-distinguisher 20:1
[PE1-vpn-instance-vpn2] vpn-target 20:1
[PE1-vpn-instance-vpn2] quit
```

2. Configure routing between the MCE and VPN sites:

The MCE is connected to VPN 1 directly, and no routing protocol is enabled in VPN 1. Therefore, you can configure static routes.

# On VR 1, assign IP address 10.214.10.2/24 to the interface connected to MCE and 192.168.0.1/24 to the interface connected to VPN 1. Add ports to VLANs correctly. (Details not shown.)

# On VR 1, configure a default route with the next hop being 10.214.10.3.

```
<VR1> system-view
[VR1] ip route-static 0.0.0.0 0.0.0.0 10.214.10.3
```

# On the MCE, configure a static route to 192.168.0.0/24 with the next hop 10.214.10.2. Bind the static route to VPN instance **vpn1**.

```
[MCE] ip route-static vpn-instance vpn1 192.168.0.0 24 10.214.10.2
```

# On the MCE, display the routing information maintained for VPN instance **vpn1**.

```
[MCE] display ip routing-table vpn-instance vpn1

Destinations : 13        Routes : 13

Destination/Mask   Proto   Pre Cost        NextHop         Interface
0.0.0.0/32         Direct  0   0           127.0.0.1       InLoop0
10.214.10.0/24     Direct  0   0           10.214.10.3     Vlan10
10.214.10.0/32     Direct  0   0           10.214.10.3     Vlan10
10.214.10.3/32     Direct  0   0           127.0.0.1       InLoop0
10.214.10.255/32   Direct  0   0           10.214.10.3     Vlan10
127.0.0.0/8        Direct  0   0           127.0.0.1       InLoop0
127.0.0.0/32       Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32       Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
192.168.0.0/24     Static  60  0           10.214.10.2     Vlan10
224.0.0.0/4        Direct  0   0           0.0.0.0         NULL0
224.0.0.0/24       Direct  0   0           0.0.0.0         NULL0
255.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
```

The output shows that the MCE has a static route for VPN instance **vpn1**.

# Run OSPF in VPN 2. Create OSPF process 20 and bind it to VPN instance **vpn2** on the MCE, so that the MCE can learn the routes of VPN 2 and add them to the routing table of VPN instance **vpn2**.

```
[MCE] ospf 2 vpn-instance vpn2
```

# Advertise subnet 10.214.20.0.

```
[MCE-ospf-2] area 0
[MCE-ospf-2-area-0.0.0.0] network 10.214.20.0 0.0.0.255
[MCE-ospf-2-area-0.0.0.0] quit
[MCE-ospf-2] quit
```

# On VR 2, assign IP address 10.214.20.2/24 to the interface connected to MCE and 192.168.10.1/24 to the interface connected to VPN 2. (Details not shown.)

# Configure OSPF process 2, and advertise subnets 192.168.10.0 and 10.214.20.0.

```
<VR2> system-view
[VR2] ospf 2
[VR2-ospf-2] area 0
[VR2-ospf-2-area-0.0.0.0] network 192.168.10.0 0.0.0.255
[VR2-ospf-2-area-0.0.0.0] network 10.214.20.0 0.0.0.255
[VR2-ospf-2-area-0.0.0.0] quit
[VR2-ospf-2] quit
```

# On the MCE, display the routing information maintained for VPN instance **vpn2**.

```
[MCE] display ip routing-table vpn-instance vpn2

Destinations : 13        Routes : 13

Destination/Mask   Proto   Pre Cost        NextHop         Interface
0.0.0.0/32         Direct  0   0           127.0.0.1       InLoop0
10.214.20.0/24     Direct  0   0           10.214.20.3     Vlan20
10.214.20.0/32     Direct  0   0           10.214.20.3     Vlan20
10.214.20.3/32     Direct  0   0           127.0.0.1       InLoop0
10.214.20.255/32   Direct  0   0           10.214.20.3     Vlan20
127.0.0.0/8        Direct  0   0           127.0.0.1       InLoop0
127.0.0.0/32       Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32       Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
192.168.10.0/24    O_INTRA 10  2           10.214.20.2     Vlan20
224.0.0.0/4        Direct  0   0           0.0.0.0         NULL0
224.0.0.0/24       Direct  0   0           0.0.0.0         NULL0
255.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
```

The output shows that the MCE has learned the private routes of VPN 2. The MCE maintains the routes of VPN 1 and those of VPN2 in two different routing tables. In this way, routes from different VPNs are separated.

3. Configure routing between the MCE and PE 1:

# On the MCE, bind VLAN-interface 30 to VPN instance **vpn1**, and configure an IP address for the VLAN interface.

```
[MCE] interface vlan-interface 30
[MCE-Vlan-interface30] ip binding vpn-instance vpn1
[MCE-Vlan-interface30] ip address 30.1.1.1 24
[MCE-Vlan-interface30] quit
```

# Bind VLAN-interface 40 to VPN instance **vpn2**, and configure an IP address for the VLAN interface.

```
[MCE] interface vlan-interface 40
[MCE-Vlan-interface40] ip binding vpn-instance vpn2
[MCE-Vlan-interface40] ip address 40.1.1.1 24
[MCE-Vlan-interface40] quit
```

# On PE 1, bind VLAN-interface 30 to VPN instance **vpn1**, and configure an IP address for the VLAN interface.

```
[PE1] interface vlan-interface 30
[PE1-Vlan-interface30] ip binding vpn-instance vpn1
[PE1-Vlan-interface30] ip address 30.1.1.2 24
[PE1-Vlan-interface30] quit
```

# Bind VLAN-interface 40 to VPN instance **vpn2**, and configure an IP address for the VLAN interface.

```
[PE1] interface vlan-interface 40
[PE1-Vlan-interface40] ip binding vpn-instance vpn2
[PE1-Vlan-interface40] ip address 40.1.1.2 24
[PE1-Vlan-interface40] quit
```

# Configure the IP address of the interface Loopback 0 as 101.101.10.1 for the MCE and as 100.100.10.1 for PE 1. Specify the loopback interface address as the router ID for the MCE and PE 1. (Details not shown.)

# Enable OSPF process 10 on the MCE, and bind the process to VPN instance **vpn1**.

```
[MCE] ospf 10 router-id 101.101.10.1 vpn-instance vpn1
```

# Disable OSPF routing loop detection for the VPN instance.

```
[MCE-ospf-10] vpn-instance-capability simple
```

# Set the domain ID to 10.

```
[MCE-ospf-10] domain-id 10
```

# On the MCE, advertise subnet 30.1.1.0 in area 0, and redistribute the static route of VPN 1.

```
[MCE-ospf-10] area 0
[MCE-ospf-10-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[MCE-ospf-10-area-0.0.0.0] quit
[MCE-ospf-10] import-route static
```

# On PE 1, enable OSPF process 10, and bind the process to VPN instance **vpn1**.

```
[PE1] ospf 10 router-id 100.100.10.1 vpn-instance vpn1
```

# Set the domain ID to 10.

```
[PE1-ospf-10] domain-id 10
```

# Advertise subnet 30.1.1.0 in area 0.

```
[PE1-ospf-10] area 0
[PE1-ospf-10-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[PE1-ospf-10-area-0.0.0.0] quit
[PE1-ospf-10] quit
```

# Use similar procedures to configure OSPF process 20 between MCE and PE 1 and redistribute VPN 2's routing information. (Details not shown.)

Verifying the configuration

# Verify that PE 1 has learned the static route of VPN 1 through OSPF.

```
[PE1] display ip routing-table vpn-instance vpn1

Destinations : 13        Routes : 13

Destination/Mask   Proto   Pre Cost        NextHop         Interface
0.0.0.0/32         Direct  0   0           127.0.0.1       InLoop0
30.1.1.0/24        Direct  0   0           30.1.1.2        Vlan30
30.1.1.0/32        Direct  0   0           30.1.1.2        Vlan30
30.1.1.2/32        Direct  0   0           127.0.0.1       InLoop0
```

```
30.1.1.255/32       Direct  0   0           30.1.1.2        Vlan30
127.0.0.0/8         Direct  0   0           127.0.0.1       InLoop0
127.0.0.0/32        Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32        Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
192.168.0.0/24      O_ASE2  150 1           30.1.1.1        Vlan30
224.0.0.0/4         Direct  0   0           0.0.0.0         NULL0
224.0.0.0/24        Direct  0   0           0.0.0.0         NULL0
255.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
```

# Verify that PE 1 has learned the routes of OSPF process 20 in VPN 2 through OSPF.

```
[PE1] display ip routing-table vpn-instance vpn2

Destinations : 13        Routes : 13

Destination/Mask    Proto   Pre Cost        NextHop         Interface
0.0.0.0/32          Direct  0   0           127.0.0.1       InLoop0
40.1.1.0/24         Direct  0   0           40.1.1.2        Vlan40
40.1.1.0/32         Direct  0   0           40.1.1.2        Vlan40
40.1.1.2/32         Direct  0   0           127.0.0.1       InLoop0
40.1.1.255/32       Direct  0   0           40.1.1.2        Vlan40
127.0.0.0/8         Direct  0   0           127.0.0.1       InLoop0
127.0.0.0/32        Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32        Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
192.168.10.0/24     O_ASE2  150 1           40.1.1.1        Vlan40
224.0.0.0/4         Direct  0   0           0.0.0.0         NULL0
224.0.0.0/24        Direct  0   0           0.0.0.0         NULL0
255.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
```

The routing information for the two VPNs has been redistributed into the routing tables on PE 1.

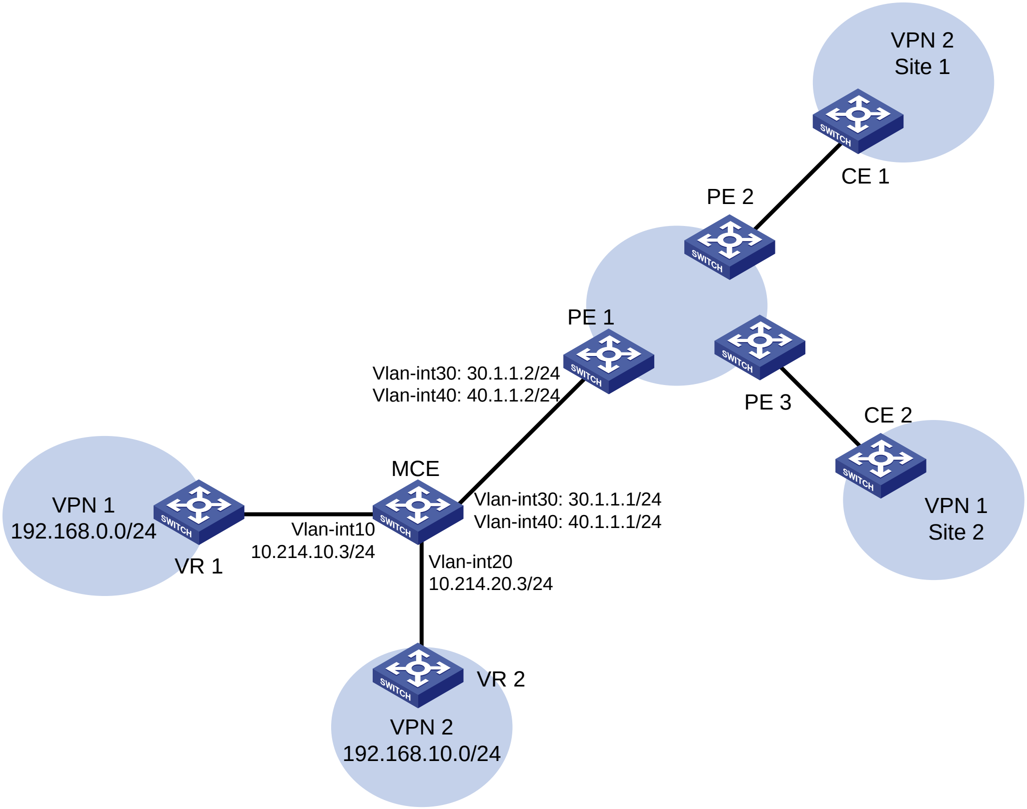
Example: Configuring the MCE that uses EBGP to advertise VPN routes to the PE

Network configuration

As shown in Figure 5, configure the MCE to advertise the routes of VPNs 1 and 2 to PE 1, so that the sites of each VPN can communicate with each other over the MPLS backbone.

Run OSPF in both VPN 1 and VPN 2. Run EBGP between the MCE and PE 1.

Figure 5 Network diagram



Procedure

1. Create VPN instances on the MCE and PE 1, and bind the VPN instances to VLAN interfaces. For the configuration procedure, see "Example: Configuring the MCE that uses OSPF to advertise VPN routes to the PE."
2. Configure routing between the MCE and VPN sites:

   # Enable an OSPF process on the devices in the two VPNs, and advertise the subnets. (Details not shown.)

   # Configure OSPF on the MCE, and bind OSPF process 10 to VPN instance **vpn1** to learn the routes of VPN 1.

   ```
   <MCE> system-view
   [MCE] ospf 10 router-id 10.10.10.1 vpn-instance vpn1
   [MCE-ospf-10] area 0
   [MCE-ospf-10-area-0.0.0.0] network 10.214.10.0 0.0.0.255
   [MCE-ospf-10-area-0.0.0.0] quit
   [MCE-ospf-10] quit
   ```

   # Display the routing table of VPN 1 on the MCE.

   ```
   [MCE] display ip routing-table vpn-instance vpn1

   Destinations : 13        Routes : 13

   Destination/Mask    Proto  Pre Cost        NextHop         Interface
   0.0.0.0/32          Direct 0   0           127.0.0.1       InLoop0
   10.214.10.0/24      Direct 0   0           10.214.10.3     Vlan10
   10.214.10.0/32      Direct 0   0           10.214.10.3     Vlan10
   ```

```
10.214.10.3/32      Direct  0   0           127.0.0.1       InLoop0
10.214.10.255/32    Direct  0   0           10.214.10.3     Vlan10
127.0.0.0/8         Direct  0   0           127.0.0.1       InLoop0
127.0.0.0/32        Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32        Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32  Direct  0   0           127.0.0.1       InLoop0
192.168.0.0/24      O_INTRA 10  2           10.214.10.2     Vlan10
224.0.0.0/4         Direct  0   0           0.0.0.0         NULL0
224.0.0.0/24        Direct  0   0           0.0.0.0         NULL0
255.255.255.255/32  Direct  0   0           127.0.0.1       InLoop0
```

The output shows that the MCE has learned the private route of VPN 1 through OSPF process 10.

# On the MCE, bind OSPF process 20 to VPN instance **vpn2** to learn the routes of VPN 2. The configuration procedure is similar to that for OSPF process 10.

The output shows that the MCE has learned the private route of VPN 2 through OSPF.

```
[MCE] display ip routing-table vpn-instance vpn2

Destinations : 13        Routes : 13

Destination/Mask    Proto   Pre Cost        NextHop         Interface
0.0.0.0/32          Direct  0   0           127.0.0.1       InLoop0
10.214.20.0/24      Direct  0   0           10.214.20.3     Vlan20
10.214.20.0/32      Direct  0   0           10.214.20.3     Vlan20
10.214.20.3/32      Direct  0   0           127.0.0.1       InLoop0
10.214.20.255/32    Direct  0   0           10.214.20.3     Vlan20
127.0.0.0/8         Direct  0   0           127.0.0.1       InLoop0
127.0.0.0/32        Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32        Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32  Direct  0   0           127.0.0.1       InLoop0
192.168.10.0/24     O_INTRA 10  2           10.214.20.2     Vlan20
224.0.0.0/4         Direct  0   0           0.0.0.0         NULL0
224.0.0.0/24        Direct  0   0           0.0.0.0         NULL0
255.255.255.255/32  Direct  0   0           127.0.0.1       InLoop0
```

3. Configure routing between the MCE and PE 1:

   # Configure the ports between the MCE and PE 1 as trunk ports. The configuration procedure is similar to that described in "Example: Configuring the MCE that uses OSPF to advertise VPN routes to the PE." (Details not shown.)

   # Enable BGP in AS 100 on the MCE.

   ```
   [MCE] bgp 100
   ```

   # Enter the BGP-VPN instance view of VPN instance **vpn1**.

   ```
   [MCE-bgp-default] ip vpn-instance vpn1
   ```

   # Specify the EBGP peer PE 1 in AS 200.

   ```
   [MCE-bgp-default-vpn1] peer 30.1.1.2 as-number 200
   ```

   # Activate the EBGP VPNv4 peer PE 1, and redistribute routing information from OSPF process 10 to BGP.

   ```
   [MCE-bgp-default-vpn1] address-family ipv4
   [MCE-bgp-default-ipv4-vpn1] peer 30.1.1.2 enable
   [MCE-bgp-default-ipv4-vpn1] import-route ospf 10
   ```

# On PE 1, enable BGP in AS 200, and specify the MCE as its EBGP peer.

```
[PE1] bgp 200
[PE1-bgp-default] ip vpn-instance vpn1
[PE1-bgp-default-vpn1] peer 30.1.1.1 as-number 100
[PE1-bgp-default-vpn1] address-family ipv4
[PE1-bgp-default-ipv4-vpn1] peer 30.1.1.1 enable
[PE1-bgp-default-ipv4-vpn1] quit
[PE1-bgp-default-vpn1] quit
[PE1-bgp-default] quit
```

# Use similar procedures to configure VPN 2 settings on MCE and PE 1. (Details not shown.)

Verifying the configuration

# Verify that PE 1 has learned the OSPF route of VPN 1 through BGP.

```
[PE1] display ip routing-table vpn-instance vpn1

Destinations : 13        Routes : 13

Destination/Mask   Proto   Pre Cost        NextHop         Interface
0.0.0.0/32         Direct  0   0           127.0.0.1       InLoop0
30.1.1.0/24        Direct  0   0           30.1.1.2        Vlan30
30.1.1.0/32        Direct  0   0           30.1.1.2        Vlan30
30.1.1.2/32        Direct  0   0           127.0.0.1       InLoop0
30.1.1.255/32      Direct  0   0           30.1.1.2        Vlan30
127.0.0.0/8        Direct  0   0           127.0.0.1       InLoop0
127.0.0.0/32       Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32       Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
192.168.0.0/24     BGP     255 3           30.1.1.1        Vlan30
224.0.0.0/4        Direct  0   0           0.0.0.0         NULL0
224.0.0.0/24       Direct  0   0           0.0.0.0         NULL0
255.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
```

# Verify that PE 1 has learned the OSPF route of VPN 2 through BGP.

```
[PE1] display ip routing-table vpn-instance vpn2

Destinations : 13        Routes : 13

Destination/Mask   Proto   Pre Cost        NextHop         Interface
0.0.0.0/32         Direct  0   0           127.0.0.1       InLoop0
40.1.1.0/24        Direct  0   0           40.1.1.2        Vlan40
40.1.1.0/32        Direct  0   0           40.1.1.2        Vlan40
40.1.1.2/32        Direct  0   0           127.0.0.1       InLoop0
40.1.1.255/32      Direct  0   0           40.1.1.2        Vlan40
127.0.0.0/8        Direct  0   0           127.0.0.1       InLoop0
127.0.0.0/32       Direct  0   0           127.0.0.1       InLoop0
127.0.0.1/32       Direct  0   0           127.0.0.1       InLoop0
127.255.255.255/32 Direct  0   0           127.0.0.1       InLoop0
192.168.10.0/24    BGP     255 3           40.1.1.1        Vlan40
224.0.0.0/4        Direct  0   0           0.0.0.0         NULL0
```

```
224.0.0.0/24       Direct  0   0            0.0.0.0         NULL0
255.255.255.255/32 Direct  0   0            127.0.0.1       InLoop0
```

The MCE has redistributed the OSPF routes of the two VPN instances into the EBGP routing tables of PE 1.

Configuring IPv6 MCE

About IPv6 MCE

IPv6 MCE provides traffic separation between VPNs by distinguishing routes of the VPNs. It allows multiple VPNs to share the same CE with ensured data security in an IPv6 MPLS L3VPN network.

IPv6 MPLS L3VPN uses BGP to advertise IPv6 VPN routes and uses MPLS to forward IPv6 VPN packets on the service provider backbone.

IPv6 MCE uses the same working mechanism as MCE. It advertises IPv6 routes and exchanges IPv6 packets between the internal network and the PE.

Restrictions and guidelines: IPv6 MCE configuration

On the PE, disable routing loop detection to avoid route loss during route calculation, and disable route redistribution between routing protocols to save system resources.

IPv6 MCE tasks at a glance

To configure IPv6 MCE, perform the following tasks:

1. Configuring VPN instances

   Perform the following VPN instance tasks on PEs and MCEs:

   a. Creating a VPN instance

   b. Associating a VPN instance with an interface

   c. (Optional.) Configuring route related attributes for a VPN instance
2. Configuring routing between an MCE and a VPN site
3. Configuring routing between an MCE and a PE

Configuring VPN instances

Creating a VPN instance

About VPN instances

A VPN instance is a collection of the VPN membership and routing rules of its associated site. A VPN instance might correspond to more than one VPN.

Procedure

1. Enter system view.

   system-view
2. Create a VPN instance and enter VPN instance view.

   ip vpn-instance *vpn-instance-name*
3. Configure an RD for the VPN instance.

   route-distinguisher *route-distinguisher*

By default, no RD is configured for a VPN instance.

4. (Optional.) Configure a description for the VPN instance.

   description *text*

   By default, no description is configured for a VPN instance.

   The description should contain the VPN instance's related information, such as its relationship with a certain VPN.

5. (Optional.) Set an ID for the VPN instance.

   vpn-id *vpn-id*

   By default, no ID is configured for a VPN instance.

Associating a VPN instance with an interface

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

   This interface connects the device to the CE.

3. Associate a VPN instance with the interface.

   ip binding vpn-instance *vpn-instance-name*

   By default, an interface is not associated with a VPN instance and it belongs to the public network.

   Configuration of this command deletes the IPv6 address of the current interface. You must reconfigure an IPv6 address for the interface after configuring the command.

Configuring route related attributes for a VPN instance

Restrictions and guidelines

If you configure route related attributes in both VPN instance view and VPN instance IPv6 VPN view, IPv6 VPN uses the configurations in VPN instance IPv6 VPN view.

Prerequisites

Before you perform this task, create the routing policies to be used by this task. For information about routing policies, see *Layer 3—IP Routing Configuration Guide*.

Procedure

1. Enter system view.

   system-view

2. Enter VPN instance view or VPN instance IPv6 VPN view.

   - Enter VPN instance view.

     ip vpn-instance *vpn-instance-name*

   - Enter VPN instance IPv6 VPN view.

     ip vpn-instance *vpn-instance-name*

     address-family ipv6

3. Configure route targets.

   vpn-target *vpn-target*&<1-8> [**both** | **export-extcommunity** | **import-extcommunity**]

   By default, no route targets are configured.

4. Set the maximum number of active routes.

   routing-table limit *number* { *warn-threshold* | **simply-alert** }

   By default, the number of active routes in a VPN instance is not limited.

   Setting the maximum number of active routes for a VPN instance can prevent the device from storing too many routes.

5. Apply an import routing policy.

   import route-policy *route-policy*

   By default, all routes matching the import target attribute are accepted.

6. Apply an export routing policy.

   export route-policy *route-policy*

   By default, routes to be advertised are not filtered.

Configuring routing between an MCE and a VPN site

Configuring IPv6 static routing between an MCE and a VPN site

About IPv6 static routing between an MCE and a VPN site

An MCE can reach a VPN site through an IPv6 static route. IPv6 static routing on a traditional CE is globally effective and does not support address overlapping among VPNs. An MCE supports binding an IPv6 static route with an IPv6 VPN instance, so that the IPv6 static routes of different IPv6 VPN instances can be isolated from each other.

Perform this configuration on the MCE. On a VPN site, configure normal IPv6 static routes.

Procedure

1. Enter system view.

   system-view

2. Configure an IPv6 static route for an IPv6 VPN instance.

   ipv6 route-static vpn-instance *s-vpn-instance-name ipv6-address prefix-length* { *interface-type interface-number* [*next-hop-address*] | *nexthop-address* [**public**] | **vpn-instance** *d-vpn-instance-name nexthop-address* }

3. (Optional.) Configure the default preference for IPv6 static routes.

   ipv6 route-static default-preference *default-preference*

   The default preference for IPv6 static routes is 60.

Configuring RIPng between an MCE and a VPN site

About RIPng between an MCE and a VPN site

By configuring RIPng process-to-IPv6 VPN instance bindings on a MCE, you allow routes of different VPNs to be exchanged between the MCE and the sites through different RIPng processes, ensuring the separation and security of IPv6 VPN routes.

For more information about RIPng, see *Layer 3—IP Routing Configuration Guide*.

Perform this configuration on the MCE. On a VPN site, configure normal RIPng.

Procedure

1. Enter system view.

   system-view

2. Create a RIPng process for a VPN instance and enter RIPng view.

   ripng [*process-id*] **vpn-instance** *vpn-instance-name*

   A RIPng process can belong to only one IPv6 VPN instance.

3. Redistribute remote site routes advertised by the PE.

   import-route *protocol* [*as-number* | *process-id*] [**allow-ibgp**] [**allow-direct** | **cost** *cost-value* | **route-policy** *route-policy-name*] *

   By default, no routes are redistributed into RIPng.

4. Return to system view.

   quit

5. Enter interface view.

   interface *interface-type interface-number*

6. Enable RIPng on the interface.

   ripng *process-id* **enable**

   By default, RIPng is disabled.

Configuring OSPFv3 between an MCE and a VPN site

About OSPFv3 between an MCE and a VPN site

By configuring OSPFv3 process-to-IPv6 VPN instance bindings on a MCE, you allow routes of different IPv6 VPNs to be exchanged between the MCE and the sites through different OSPFv3 processes, ensuring the separation and security of IPv6 VPN routes.

For more information about OSPFv3, see *Layer 3—IP Routing Configuration Guide*.

Perform this configuration on the MCE. On a VPN site, configure common OSPFv3.

Procedure

1. Enter system view.

   system-view

2. Create an OSPFv3 process for a VPN instance and enter OSPFv3 view.

   ospfv3 [process-id | **vpn-instance** vpn-instance-name] *

   An OSPFv3 process can belong to only one IPv6 VPN instance.

   Deleting a VPN instance also deletes all related OSPFv3 processes.

3. Set the router ID.

   router-id *router-id*

4. Redistribute remote site routes advertised by the PE.

   import-route *protocol* [*as-number*] [*process-id* | **all-processes** | **allow-ibgp**] [**allow-direct** | **cost** *cost-value* | **nssa-only** | **route-policy** *route-policy-name* | **tag** *tag* | **type** *type*] *

   By default, no routes are redistributed into OSPFv3.

5. Return to system view.

   quit

6. Enter interface view.

   interface *interface-type interface-number*

7. Enable OSPFv3 on the interface.

   ospfv3 *process-id* **area** *area-id* [**instance** *instance-id*]

   By default, OSPFv3 is disabled on an interface.

Configuring IPv6 IS-IS between an MCE and a VPN site

About IPv6 IS-IS between an MCE and a VPN site

By configuring IPv6 IS-IS process-to-IPv6 VPN instance bindings on a MCE, you allow routes of different IPv6 VPNs to be exchanged between the MCE and the sites through different IPv6 IS-IS processes. This ensures the separation and security of IPv6 VPN routes. For more information about IPv6 IS-IS, see *Layer 3—IP Routing Configuration Guide*.

Perform this configuration on the MCE. On a VPN site, configure common IPv6 IS-IS.

Procedure

1. Enter system view.

   system-view

2. Create an IPv6 IS-IS process for a VPN instance and enter IS-IS view.

   isis [*process-id*] **vpn-instance** *vpn-instance-name*

   An IPv6 IS-IS process can belong to only one IPv6 VPN instance.

3. Configure a network entity title for the IS-IS process.

   network-entity *net*

   By default, no NET is configured.

4. Create the IS-IS IPv6 unicast address family and enter its view.

   address-family ipv6 [**unicast**]

5. (Optional.) Redistribute remote site routes advertised by the PE.

   import-route *protocol* [*as-number* | *process-id*] [**allow-ibgp**] [**allow-direct** | **cost** *cost-value* | [**level-1** | **level-1-2** | **level-2**] | **route-policy** *route-policy-name* | **tag** *tag*] *

   By default, no routes are redistributed to IPv6 IS-IS.

   If you do not specify the route level in the command, redistributed routes are added to the level-2 routing table.

6. Return to system view.

   quit

7. Enter interface view.

   interface *interface-type interface-number*

8. Enable the IPv6 IS-IS process on the interface.

   isis ipv6 enable [*process-id*]

   By default, no IPv6 IS-IS process is enabled on the interface.

Configuring EBGP between an MCE and a VPN site

About EBGP between an MCE and a VPN site

To use EBGP between an MCE and IPv6 VPN sites, you must configure a BGP peer for each IPv6 VPN instance on the MCE, and redistribute the IGP routes of each VPN instance on the IPv6 VPN sites.

For more information about IPv6 BGP, see *Layer 3—IP Routing Configuration Guide*.

Restrictions and guidelines

A VPN site must advertise IPv6 VPN network addresses it can reach to the connected MCE.

Configuring the MCE

1. Enter system view.

   ```
   system-view
   ```

2. Enter BGP instance view.

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

3. Enter BGP-VPN instance view.

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

4. Configure the VPN site as an EBGP peer.

   ```
   peer { group-name | ipv6-address [ prefix-length ] } as-number
   as-number
   ```

5. Enter BGP-VPN IPv6 unicast address family view.

   ```
   address-family ipv6 [ unicast ]
   ```

6. Enable BGP to exchange IPv6 unicast routes with the specified peer.

   ```
   peer { group-name | ipv6-address [ prefix-length ] } enable
   ```

   By default, BGP does not exchange IPv6 unicast routes with any peer.

7. Redistribute remote site routes advertised by the PE.

   ```
   import-route protocol [ { process-id | all-processes } [ allow-direct |
   med med-value | route-policy route-policy-name ] * ]
   ```

   By default, no route redistribution is configured.

Configuring the VPN site

1. Enter system view.

   ```
   system-view
   ```

2. Enter BGP instance view.

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

3. Configure the MCE as an EBGP peer.

   ```
   peer { group-name | ipv6-address [ prefix-length ] } as-number
   as-number
   ```

4. Enter BGP IPv6 unicast address family view.

   ```
   address-family ipv6 [ unicast ]
   ```

5. Enable BGP to exchange IPv6 unicast routes with the specified peer.

   ```
   peer { group-name | ipv6-address [ prefix-length ] } enable
   ```

   By default, BGP does not exchange IPv6 unicast routes with any peer.

6. Redistribute the IGP routes of the VPN.

   ```
   import-route protocol [ { process-id | all-processes } [ allow-direct |
   med med-value | route-policy route-policy-name ] * ]
   ```

   By default, no routes are redistributed into BGP.

Configuring IBGP between an MCE and a VPN site

About IBGP between an MCE and a VPN site

To use IBGP between an MCE and a VPN site, you must configure a BGP peer for each IPv6 VPN instance on the MCE, and redistribute the IGP routes of each VPN instance on the VPN site.

Restrictions and guidelines

A VPN site must advertise VPN network addresses to the connected MCE.

Configuring the MCE

1. Enter system view.

   system-view

2. Enter BGP instance view.

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

   By default, BGP is not enabled.

3. Enter BGP-VPN instance view.

   ip vpn-instance *vpn-instance-name*

4. Configure the VPN site as an IBGP peer.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **as-number** *as-number*

5. Enter BGP-VPN IPv6 unicast address family view.

   address-family ipv6 [**unicast**]

6. Enable BGP to exchange IPv6 unicast routes with the peer.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **enable**

   By default, BGP does not exchange IPv6 unicast routes with any peer.

7. (Optional.) Configure the local device as the RR, and specify the peer as the client of the RR.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **reflect-client**

   By default, no RR or RR client is configured.

   After you configure a VPN site as an IBGP peer, the MCE does not advertise the BGP routes learned from the VPN site to other IBGP peers, including VPNv6 peers. The MCE advertises routes learned from a VPN site only when you configure the VPN site as a client of the RR (the MCE).

8. Redistribute remote site routes advertised by the PE into BGP.

   import-route *protocol* [{ *process-id* | **all-processes** } [**allow-direct** | **med** *med-value* | **route-policy** *route-policy-name*] *]

   By default, no routes are redistributed into BGP.

Configuring the VPN site

1. Enter system view.

   system-view

2. Enter BGP instance view.

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

   By default, BGP is not enabled.

3. Configure the MCE as an IBGP peer.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **as-number** *as-number*

4. Enter BGP IPv6 unicast address family view.

   address-family ipv6 [**unicast**]

5. Enable BGP to exchange IPv6 unicast routes with the peer.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **enable**

   By default, BGP does not exchange IPv6 unicast routes with any peer.

6. Redistribute the IGP routes of the VPN into BGP.

import-route *protocol* [{ *process-id* | **all-processes** } [**allow-direct** | **med** *med-value* | **route-policy** *route-policy-name*] *]

By default, no routes are redistributed into BGP.

Configuring routing between an MCE and a PE

About routing between an MCE and a PE

MCE-PE routing configuration includes the following tasks:

- Binding the MCE-PE interfaces to IPv6 VPN instances.
- Performing routing configurations.
- Redistributing IPv6 VPN routes into the routing protocol running between the MCE and the PE.

Perform the following configurations on the MCE. For information about configuring the PE, see the documentation for the PE.

Configuring IPv6 static routing between an MCE and a PE

1. Enter system view.

   system-view

2. Configure an IPv6 static route for an IPv6 VPN instance.

   ipv6 route-static vpn-instance *s-vpn-instance-name ipv6-address prefix-length* { *interface-type interface-number* [*next-hop-address*] | *nexthop-address* [**public**] | **vpn-instance** *d-vpn-instance-name nexthop-address* }

3. (Optional.) Set the default preference for IPv6 static routes.

   ipv6 route-static default-preference *default-preference*

   The default value is 60.

Configuring RIPng between an MCE and a PE

1. Enter system view.

   system-view

2. Create a RIPng process for an IPv6 VPN instance and enter RIPng view.

   ripng [*process-id*] **vpn-instance** *vpn-instance-name*

3. Redistribute VPN routes.

   import-route *protocol* [*as-number* | *process-id*] [**allow-ibgp**] [**allow-direct** | **cost** *cost-value* | **route-policy** *route-policy-name*] *

   By default, no routes are redistributed into RIPng.

4. Return to system view.

   quit

5. Enter interface view.

   interface *interface-type interface-number*

6. Enable the RIPng process on the interface.

   ripng *process-id* **enable**

   By default, RIPng is disabled on an interface.

Configuring OSPFv3 between an MCE and a PE

1. Enter system view.

   ```
   system-view
   ```

2. Create an OSPFv3 process for an IPv6 VPN instance and enter OSPFv3 view.

   ```
   ospfv3 [ process-id | vpn-instance vpn-instance-name ] *
   ```

3. Set the router ID.

   ```
   router-id router-id
   ```

4. Disable routing loop detection.

   ```
   vpn-instance-capability simple
   ```

   By default, routing loop detection is enabled and the MCE does not receive OSPFv3 routes from the PE.

5. Redistribute VPN routes.

   ```
   import-route protocol [ as-number ] [ process-id | all-processes |
   allow-ibgp ] [ allow-direct | cost cost-value | nssa-only | route-policy
   route-policy-name | tag tag | type type ] *
   ```

   By default, no routes are redistributed into OSPFv3.

6. Return to system view.

   ```
   quit
   ```

7. Enter interface view.

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

8. Enable the OSPFv3 process on the interface.

   ```
   ospfv3 process-id area area-id [ instance instance-id ]
   ```

   By default, OSPFv3 is disabled on an interface.

Configuring IPv6 IS-IS between an MCE and a PE

1. Enter system view.

   ```
   system-view
   ```

2. Create an IS-IS process for an IPv6 VPN instance and enter IS-IS view.

   ```
   isis [ process-id ] vpn-instance vpn-instance-name
   ```

3. Configure a network entity title.

   ```
   network-entity net
   ```

   By default, no NET is configured.

4. Create the IS-IS IPv6 unicast address family and enter its view.

   ```
   address-family ipv6 [ unicast ]
   ```

5. Redistribute VPN routes.

   ```
   import-route protocol [ as-number | process-id ] [ allow-ibgp ]
   [ allow-direct | cost cost-value | [ level-1 | level-1-2 | level-2 ] |
   route-policy route-policy-name | tag tag ] *
   ```

   By default, IPv6 IS-IS does not redistribute routes from any other routing protocol.

   If you do not specify the route level in the command, the command redistributes routes to the level-2 routing table.

6. Return to system view.

   ```
   quit
   ```

7. Enter interface view.

   interface *interface-type interface-number*

8. Enable the IPv6 IS-IS process on the interface.

   isis ipv6 enable [*process-id*]

   By default, no IPv6 IS-IS process is enabled on the interface.

Configuring EBGP between an MCE and a PE

1. Enter system view.

   system-view

2. Enter BGP instance view.

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

3. Enter BGP-VPN instance view.

   ip vpn-instance *vpn-instance-name*

4. Configure the PE as an EBGP peer.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **as-number** *as-number*

5. Enter BGP-VPN IPv6 unicast address family view.

   address-family ipv6 [**unicast**]

6. Enable BGP to exchange IPv6 unicast routes with the specified peer.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **enable**

   By default, BGP does not exchange IPv6 unicast routes with peers.

7. Redistribute VPN routes.

   import-route *protocol* [{ *process-id* | **all-processes** } [**allow-direct** | **med** *med-value* | **route-policy** *route-policy-name*] *]

   By default, no routes are redistributed into BGP.

Configuring IBGP between an MCE and a PE

1. Enter system view.

   system-view

2. Enter BGP instance view.

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

   By default, BGP is not enabled.

3. Enter BGP-VPN instance view.

   ip vpn-instance *vpn-instance-name*

4. Configure the PE as an IBGP peer.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **as-number** *as-number*

5. Enter BGP-VPN IPv6 unicast address family view.

   address-family ipv6 [**unicast**]

6. Enable BGP to exchange IPv6 unicast routes with the specified peer.

   peer { *group-name* | *ipv6-address* [*prefix-length*] } **enable**

   By default, BGP does not exchange IPv6 unicast routes with peers.

7. Redistribute the VPN routes of the VPN site.

   import-route *protocol* [{ *process-id* | **all-processes** } [**allow-direct** | **med** *med-value* | **route-policy** *route-policy-name*] *]

By default, no routes are redistributed into BGP.

Display and maintenance commands for IPv6 MCE

Execute **display** commands in any view.

Task	Command
Display VPN instance information.	**display ip vpn-instance** [**instance-name** *vpn-instance-name*]

For commands that display routing tables for VPN instances, see *Layer 3—IP Routing Command Reference*.

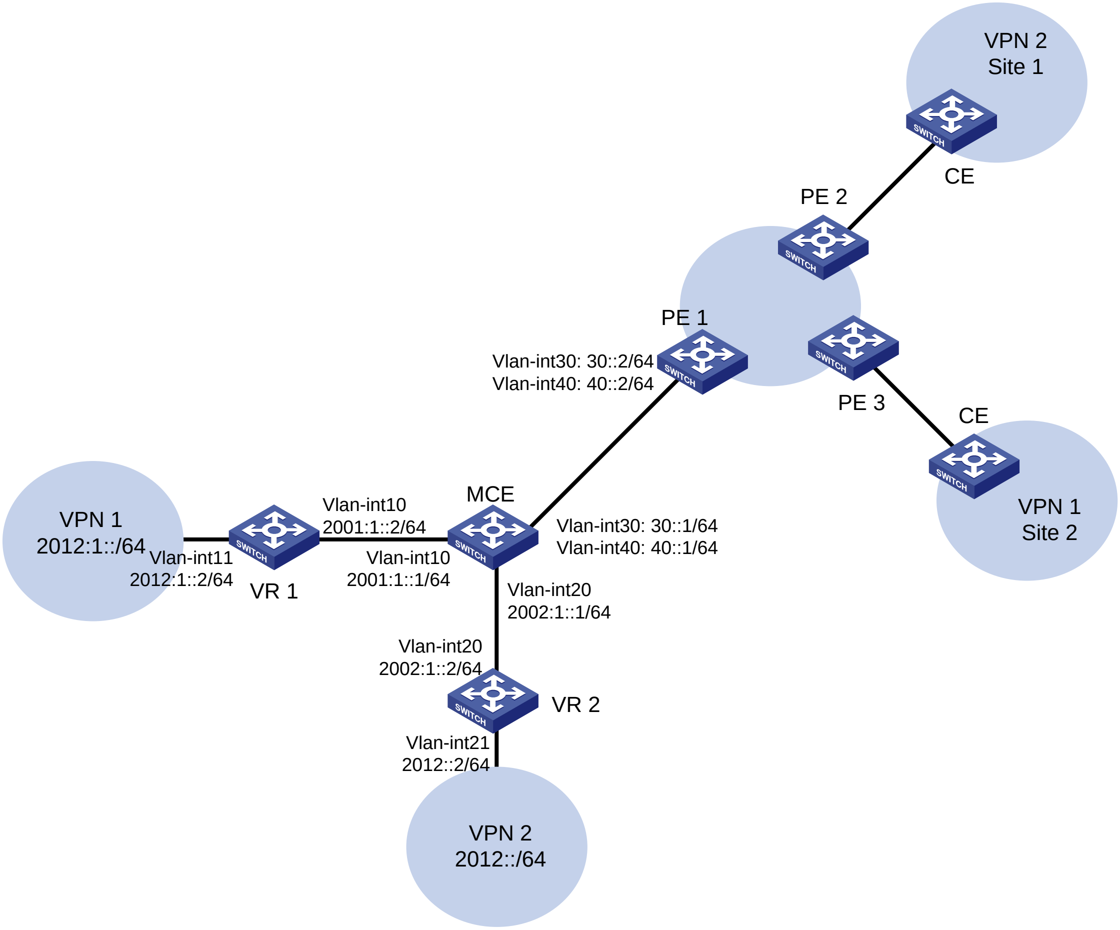
IPv6 MCE configuration examples

Example: Configuring IPv6 MCE

Network configuration

As shown in Figure 6, RIPng runs in VPN 2. The edge routers of VPN 1 and VPN 2 are VR1 and VR2, respectively. Configure the MCE device to separate routes from different VPNs and advertise VPN routes to PE 1 through OSPFv3.

Figure 6 Network diagram



Procedure

1. Configure the VPN instances on the MCE and PE 1:

   # On the MCE, configure VPN instances **vpn1** and **vpn2**, and specify an RD and route targets for each VPN instance.

   ```
   <MCE> system-view
   [MCE] ip vpn-instance vpn1
   [MCE-vpn-instance-vpn1] route-distinguisher 10:1
   [MCE-vpn-instance-vpn1] vpn-target 10:1
   [MCE-vpn-instance-vpn1] quit
   [MCE] ip vpn-instance vpn2
   [MCE-vpn-instance-vpn2] route-distinguisher 20:1
   [MCE-vpn-instance-vpn2] vpn-target 20:1
   [MCE-vpn-instance-vpn2] quit
   ```

   # Bind VLAN-interface 10 to VPN instance **vpn1**, and configure an IPv6 address for the VLAN interface.

   ```
   [MCE] interface vlan-interface 10
   [MCE-Vlan-interface10] ip binding vpn-instance vpn1
   [MCE-Vlan-interface10] ipv6 address 2001:1::1 64
   [MCE-Vlan-interface10] quit
   ```

   # Bind VLAN-interface 20 to VPN instance **vpn2**, and configure an IPv6 address for the VLAN interface.

```
[MCE] interface vlan-interface 20
[MCE-Vlan-interface20] ip binding vpn-instance vpn2
[MCE-Vlan-interface20] ipv6 address 2002:1::1 64
[MCE-Vlan-interface20] quit
```

# On PE 1, configure VPN instances **vpn1** and **vpn2**, and specify an RD and route targets for each VPN instance.

```
<PE1> system-view
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] route-distinguisher 10:1
[PE1-vpn-instance-vpn1] vpn-target 10:1
[PE1-vpn-instance-vpn1] quit
[PE1] ip vpn-instance vpn2
[PE1-vpn-instance-vpn2] route-distinguisher 20:1
[PE1-vpn-instance-vpn2] vpn-target 20:1
[PE1-vpn-instance-vpn2] quit
```

2. Configure routing between the MCE and VPN sites:

The MCE is connected to VPN 1 directly, and no routing protocol is enabled in VPN 1. Therefore, you can configure IPv6 static routes.

# On VR 1, assign IPv6 address 2001:1::2/64 to the interface connected to the MCE and 2012:1::2/64 to the interface connected to VPN 1. Add ports to VLANs. (Details not shown.)

# On VR 1, configure a default route with the next hop being 2001:1::1.

```
<VR1> system-view
[VR1] ipv6 route-static :: 0 2001:1::1
```

# On the MCE, configure an IPv6 static route to 2012:1::/64 with the next hop 2001:1::2. Bind the static route to VPN instance **vpn1**.

```
[MCE] ipv6 route-static vpn-instance vpn1 2012:1:: 64 2001:1::2
```

# Run RIPng in VPN 2. Configure RIPng process 20 for VPN instance **vpn2** on the MCE, so that the MCE can learn the routes of VPN 2 and add them to the routing table of VPN instance **vpn2**.

```
[MCE] ripng 20 vpn-instance vpn2
```

# Advertise subnet 2002:1::/64 through RIPng.

```
[MCE] interface vlan-interface 20
[MCE-Vlan-interface20] ripng 20 enable
[MCE-Vlan-interface20] quit
```

# On VR 2, assign IPv6 address 2002:1::2/64 to the interface connected to the MCE and 2012::2/64 to the interface connected to VPN 2. (Details not shown.)

# Configure RIPng, and advertise subnets 2012::/64 and 2002:1::/64.

```
<VR2> system-view
[VR2] ripng 20
[VR2-ripng-20] quit
[VR2] interface vlan-interface 20
[VR2-Vlan-interface20] ripng 20 enable
[VR2-Vlan-interface20] quit
[VR2] interface vlan-interface 21
[VR2-Vlan-interface21] ripng 20 enable
[VR2-Vlan-interface21] quit
```

# On the MCE, display the routing tables of VPN instances **vpn1** and **vpn2**.

```
[MCE] display ipv6 routing-table vpn-instance vpn1
```

```
Destinations : 6 Routes : 6

Destination: ::1/128                                     Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 2001:1::/64                                 Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : Vlan10                                      Cost      : 0

Destination: 2001:1::1/128                               Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 2012:1::/64                                 Protocol  : Static
NextHop    : 2001:1::2                                   Preference: 60
Interface  : Vlan10                                      Cost      : 0

Destination: FE80::/10                                   Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : NULL0                                       Cost      : 0

Destination: FF00::/8                                    Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : NULL0                                       Cost      : 0
[MCE] display ipv6 routing-table vpn-instance vpn2

Destinations : 6 Routes : 6

Destination: ::1/128                                     Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 2002:1::/64                                 Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : Vlan20                                      Cost      : 0

Destination: 2002:1::1/128                               Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 2012::/64                                   Protocol  : RIPng
NextHop    : FE80::20C:29FF:FE40:701                     Preference: 100
Interface  : Vlan20                                      Cost      : 1

Destination: FE80::/10                                   Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : NULL0                                       Cost      : 0
```

```
Destination: FF00::/8                                      Protocol  : Direct
NextHop    : ::                                            Preference: 0
Interface  : NULL0                                         Cost      : 0
```

The output shows that the MCE has learned the private route of VPN 2. The MCE maintains the routes of VPN 1 and VPN 2 in two different routing tables. In this way, routes from different VPNs are separated.

3. Configure routing between the MCE and PE 1:

   # On the MCE, bind VLAN-interface 30 to VPN instance **vpn1**, and configure an IPv6 address for the VLAN interface.

   ```
   [MCE] interface vlan-interface 30
   [MCE-Vlan-interface30] ip binding vpn-instance vpn1
   [MCE-Vlan-interface30] ipv6 address 30::1 64
   [MCE-Vlan-interface30] quit
   ```

   # Bind VLAN-interface 40 to VPN instance **vpn2**, and configure an IPv6 address for the VLAN interface.

   ```
   [MCE] interface vlan-interface 40
   [MCE-Vlan-interface40] ip binding vpn-instance vpn2
   [MCE-Vlan-interface40] ipv6 address 40::1 64
   [MCE-Vlan-interface40] quit
   ```

   # On PE 1, bind VLAN-interface 30 to VPN instance **vpn1**, and configure an IPv6 address for the VLAN interface.

   ```
   [PE1] interface vlan-interface 30
   [PE1-Vlan-interface30] ip binding vpn-instance vpn1
   [PE1-Vlan-interface30] ipv6 address 30::2 64
   [PE1-Vlan-interface30] quit
   ```

   # Bind VLAN-interface 40 to VPN instance **vpn2**, and configure an IPv6 address for the VLAN interface.

   ```
   [PE1] interface vlan-interface 40
   [PE1-Vlan-interface40] ip binding vpn-instance vpn2
   [PE1-Vlan-interface40] ipv6 address 40::2 64
   [PE1-Vlan-interface40] quit
   ```

   # Configure the IP address of the interface Loopback 0 as 101.101.10.1 for the MCE and as 100.100.10.1 for PE 1. Specify the loopback interface address as the router ID for the MCE and PE 1. (Details not shown.)

   # Enable OSPFv3 process 10 on the MCE, and bind the process to VPN instance **vpn1**.

   ```
   [MCE] ospfv3 10 vpn-instance vpn1
   ```

   # Redistribute the IPv6 static route of VPN 1.

   ```
   [MCE-ospf-10] router-id 101.101.10.1
   [MCE-ospf-10] import-route static
   [MCE-ospf-10] quit
   ```

   # Enable OSPFv3 on VLAN-interface 30.

   ```
   [MCE] interface vlan-interface 30
   [MCE-Vlan-interface30] ospfv3 10 area 0.0.0.0
   [MCE-Vlan-interface30] quit
   ```

   # On PE 1, enable OSPFv3 process 10 and bind the process to VPN instance **vpn1**.

   ```
   [PE1] ospfv3 10 vpn-instance vpn1
   [PE1-ospf-10] router-id 100.100.10.1
   [PE1-ospf-10] quit
   ```

# Enable OSPFv3 on VLAN-interface 30.

```
[PE1] interface vlan-interface 30
[PE1-Vlan-interface30] ospfv3 10 area 0.0.0.0
[PE1-Vlan-interface30] quit
```

# Configure OSPFv3 process 20 between the MCE and PE 1. (Details not shown.)

# Redistribute VPN 2's routes from RIPng process 20 into the OSPFv3 routing table of the MCE. (Details not shown.)

Verifying the configuration

# Verify that PE 1 has learned the private route of VPN 1 through OSPFv3.

```
[PE1] display ipv6 routing-table vpn-instance vpn1

Destinations : 6 Routes : 6

Destination: ::1/128                                     Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 30::/64                                     Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : Vlan30                                      Cost      : 0

Destination: 30::2/128                                   Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 2012:1::/64                                 Protocol  : O_ASE2
NextHop    : FE80::202:FF:FE02:2                         Preference: 150
Interface  : Vlan30                                      Cost      : 1

Destination: FE80::/10                                   Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : NULL0                                       Cost      : 0

Destination: FF00::/8                                    Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : NULL0                                       Cost      : 0
```

# Verify that PE 1 has learned the private route of VPN 2 through OSPFv3.

```
[PE1] display ipv6 routing-table vpn-instance vpn2

Destinations : 6 Routes : 6

Destination: ::1/128                                     Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 40::/64                                     Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : Vlan40                                      Cost      : 0
```

```
Destination: 40::2/128                                 Protocol  : Direct
NextHop    : ::1                                       Preference: 0
Interface  : InLoop0                                   Cost      : 0

Destination: 2012::/64                                 Protocol  : O_ASE2
NextHop    : FE80::200:FF:FE0F:5                       Preference: 150
Interface  : Vlan40                                    Cost      : 1

Destination: FE80::/10                                 Protocol  : Direct
NextHop    : ::                                        Preference: 0
Interface  : NULL0                                     Cost      : 0

Destination: FF00::/8                                  Protocol  : Direct
NextHop    : ::                                        Preference: 0
Interface  : NULL0                                     Cost      : 0
```

The routing information for the two VPNs has been added into the routing tables on PE 1.