VPN Configuration

Contents

1 Configuring MPLS L3VPN

1.1 Introduction

1.1.1 Overview

Multiprotocol Label Switching (MPLS) layer 3 virtual private network (L3VPN) interconnects geographically dispersed VPN sites by using Border Gateway Protocol (BGP) to exchange VPN routes and labels and using MPLS to forward VPN packets through public network tunnels between edge devices on networks of service providers (SPs). In this way, the networks are unified. An MPLS L3VPN is also called a BGP/MPLS IP VPN.

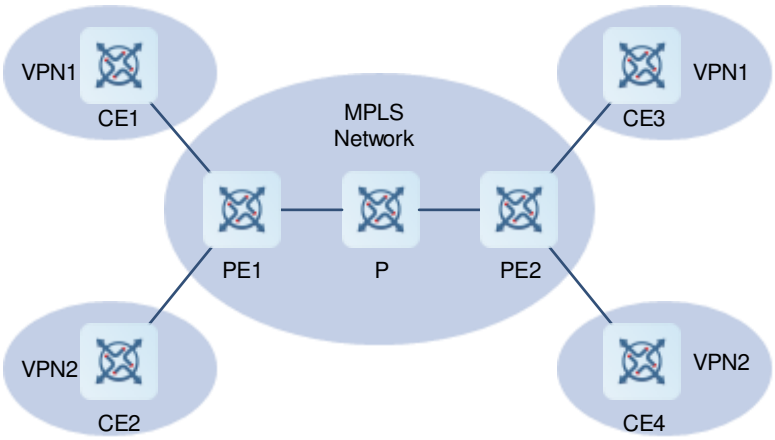
ISPs run MPLS on their IP backbone networks to provide VPN services for users' private networks. The implementation of VPN through MPLS has natural edges. VPN users no longer need dedicated VPN devices and can directly use traditional routers to build VPNs. MPLS L3VPN supports address overlapping between different VPNs, provides flexible networking modes, and has excellent scalability. MPLS L3VPNs managed by Internet service providers (ISPs) have more flexible and advanced dynamic tunnel mechanisms, more reasonable network structures, and more scalable route control and access control, helping enterprise customers dramatically reduce the VPN setup and maintenance costs. MPLS L3VPN has become an important method for ISPs to provide value-added services.

1.1.2 Basic MPLS L3VPN Architecture

1. Network Structure

Figure 1-1 shows the network structure of an MPLS L3VPN.

Figure 1-1 MPLS L3VPN Structure



An MPLS L3VPN comprises the following three important roles:

- Customer edge (CE): A CE is located at customer network edge, and logically belongs to a user VPN. A CE directly connects to a service provider edge (PE). A CE can be a host, router, or switch, as CE1, CE2, CE3, and CE4 in Figure 1-1. A CE may not support MPLS.
- PE: A PE is an edge device on an ISP's backbone network and logically belongs to an SP. A PE is mainly

responsible for receiving the VPN information from CEs and transmitting the information to other PEs, or receiving the VPN information from other PEs and sending it to the CEs. A PE is directly connected to one or more CEs. A PE can be a router, Asynchronous Transfer Mode (ATM) switch, or frame relay (FR) switch, as PE1 and PE2 in Figure 1-1. The PEs must support MPLS.

- Provider (P): P is a core device on the SP backbone network. A P is responsible for routing and fast forwarding and is not directly connected to CEs. A P knows the routes to any destination on the backbone network but does not know the routes to a VPN. As a device on the core MPLS backbone network, a P must support MPLS.

2. Application Features

MPLS L3VPN has the following features:

- VPN tunnels are set up on the PEs of an ISP, and VPN routes are transmitted between PEs. In this manner, users do not need to maintain VPN information.
- MPLS L3VPN utilizes existing routing protocols to dynamically set up VPN tunnels and advertise routes. This facilitates VPN expansion.
- Address overlapping is supported. Different VPN users can use the same address space.
- On SP networks, VPN services are exchanged according to labels rather than traditional routes.
- MPLS L3VPN is as secure as user dedicated lines.

3. Implementation Mechanism

The basic MPLS L3VPN implementation mechanism includes the following features:

- Utilize the Label Distribution Protocol (LDP) to set up label switched paths (LSPs) on the backbone network. This process is generally completed when the SP network is established and the topology becomes stable.
- Forward data packets based on the pushed labels and the local mapping table.
- Use Multiprotocol Border Gateway Protocol (MP-BGP) to transmit VPN routes and carry VPN attributes and labels.
- Manage VPN routes, including setup of multiple routing tables and VPN routing information maintenance.

4. VRF

A VPN routing and forwarding (VRF) instance addresses local route conflicts.

Each connection between a PE and a CE is associated with a VRF instance. One PE can have several VRF instances to exchange routing information with CEs. You can consider each VRF instance as a virtual router, which connects to a CE to receive routing information from the CE or notify the CE of the VPN routing information. VRF instances address the problem of conflict local routes on a PE due to the adoption of the same address space by different VPNs.

One VRF instance includes the following:

- An independent routing table
- A group of interfaces that belong to the VRF instance
- A group of routing protocols that are used in the VRF instance

5. RD

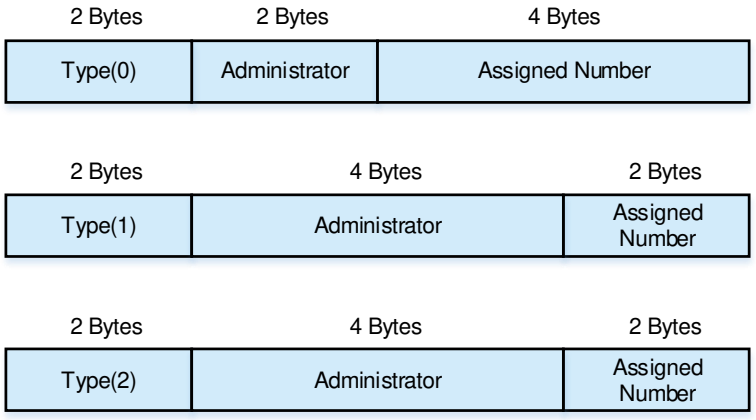
Route Distinguisher (RD) is an important VPN route attribute used to address route conflicts during transmission. If different VPNs use the same network address and advertise their routing information on the backbone network through BGP, the BGP chooses only the best route from the overlapped addresses to advertise. As a result, some VPNs cannot obtain their routing information. To resolve this problem, you can add different RD values to the overlapped addresses. Based on the different RD values carried in VPN information, the BGP can distinguish VPNs with the same network address, and therefore each VPN can obtain its own routing information.

RD is used to distinguish routing information of different VPNs with the same network address. If address overlapping does not exist between different VPNs, you do not need to configure RD values.

Generally speaking, a unique RD value is specified for each VPN. This ensures normal transmission of routing information of different VPNs on the backbone network. The RD value is generally defined as xx:xx, such as RD 1:100, among which 1 stands for the autonomous system (AS) number of the backbone network and 100 is a number specified by the user. A VPN route carries only one RD value.

An RD consists of three fields: **Type**, **Administrator**, and **Assigned Number**. Based on the value of the **Type** field, three encoding formats are adopted.

Figure 1-2 RD Structure



- When **Type** is set to **0**, the **Administrator** field has two bytes and is marked by a public AS number. The **Assigned Number** field has four bytes and is managed by the SP.
- When **Type** is set to **1**, the **Administrator** field has four bytes and uses a global IPv4 address. The **Assigned Number** field has two bytes and is managed by the SP.
- When **Type** is set to **2**, the **Administrator** field has four bytes and is marked by a four-byte AS number. The **Assigned Number** field has two bytes and is managed by the SP.

6. RT

Route-Target (RT) is an important VPN route attribute that enables a device to choose its route selection mode.

The RT attribute is further classified into Export Route-Target and Import Route-Target attributes. A PE adds the Export Route-Target attribute to the VPN routes received from CEs and then notifies other PEs of the VPN routes. The PE determines whether to import the routes received from other PEs to the VRF instance based on the Import Route-Target attribute. One principle is that when a PE receives a VPN route, the PE imports the route to the VRF instance only when at least one RT attribute carried in the route is the same as the Import

Route-Target attribute in the VRF instance of the PE. In this manner, you can flexibly control the advertising of VPN routes. A VPN route can carry multiple RT values.

Figure 1-3 RT Structure

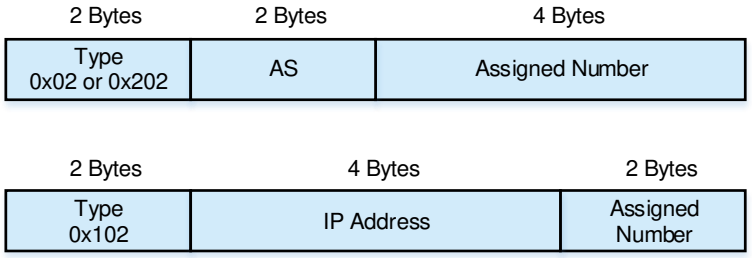


Figure 1-3 shows the decoding structure of RT carried in extended community attributes defined in BGP. The definition of RT is similar to that of RD. When **Type** is set to **0x02** or **0x202**, the AS number must be a public one. When **Type** is set to **0x102**, the IPv4 address must be a global one rather than a private address.

7. MP-BGP

The VPN routing information is transmitted on the backbone network through BGP. The Export Route-Target attribute is carried in extended community attributes defined in BGP.

The traditional BGP4 transmits only IPv4 route information and cannot carry VPN information that includes RD. Therefore, the BGP needs to be extended to introduce new attributes. MP-BGP introduces new attributes to support multiple protocols. MP-BGP can carry VPN information. A VPN route is composed of RD and IP address prefix. By adding RD values to VPN routes exchanged between PEs, the MP-BGP allows VPN users to change IPv4 routes to VPN-IPv4 routes and transmit them on the MPLS backbone network.

1.1.3 Inter-AS VPN Service Model

On an actual network, different sites of a VPN may be located in different ASs. The inter-AS VPN technology ensures communication between VPN sites in different ASs. In this case, VPN routes can be exchanged between different ASs.

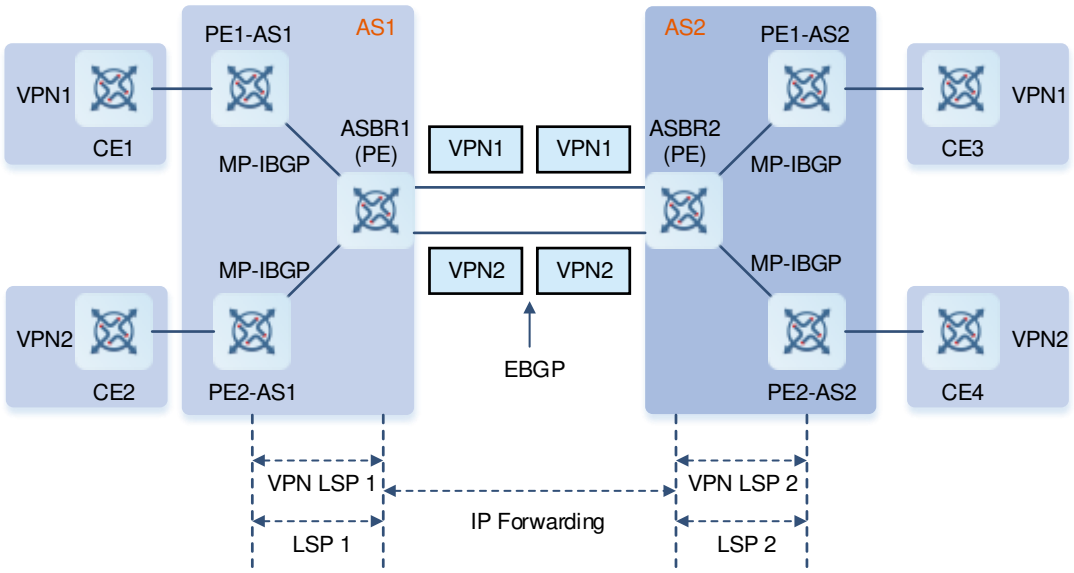
RFC 4364 provides the following three inter-AS VPN solutions:

- Option A: VRF-to-VRF mode
- Option B: single-hop MP-EBGP mode
- Option C: multi-hop MP-EBGP mode

1. Option A: VRF-to-VRF Mode

The VRF-to-VRF mode is also called the VRF back-to-back. An AS Border Router (ASBR) of an AS creates a VRF instance for each inter-AS VPN and binds the VRF instance to an interface to receive VPN routes from the local AS. The VRF instances on different ASBRs exchange VPN routes through bound interfaces. A VRF instance establishes an External Border Gateway Protocol (EBGP) connection with a VRF instance in another AS to exchange IPv4 routes.

Figure 1-4 VRF-to-VRF Inter-AS VPN



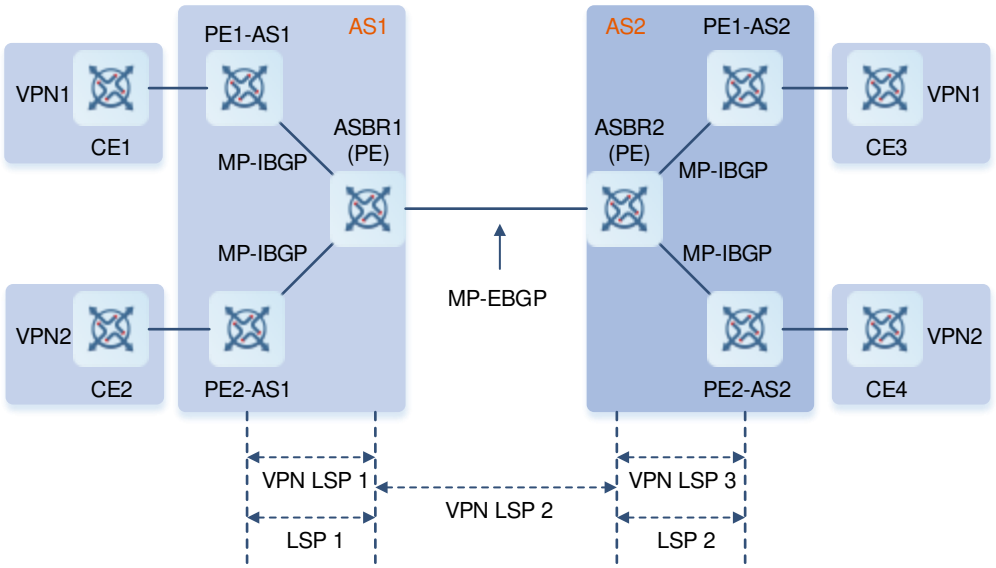
As shown in Figure 1-4, the VRF instances on two ASBRs establish a common EBGP session to exchange IPv4 routes, and the ASBRs and PEs in the same ASs establish MP-IBGP sessions to exchange VPN routes. For the VRF instance on an ASBR, the other VRF instance, with which an EBGP session is established, is equivalent to a CE. The Option A configuration solution is similar to a common intra-domain configuration solution. The ASBRs and PEs in the same ASs establish MP-IBGP sessions to exchange VPN routes. The VRF instances on different ASBRs establish EBGP sessions in BGP VRF address family mode to exchange IPv4 routes.

The VRF-to-VRF mode is easy to implement by directly using MP-IBGP. The service deployment is also simple. This configuration solution, however, requires an interface (generally a logical sub-interface) be configured for each inter-AS VPN on an ASBR one after another, complicating network expansion. In addition, the number of bound interfaces should be at least equal to the number of inter-AS VPNs. The separate creation of interfaces for each VPN poses high requirements on ASBRs. As a result, the Option A solution is generally applicable to networks with a small number of inter-AS VPNs.

2. Option B: Single-Hop MP-EBGP Mode

In the Option A solution, you need to configure a VRF instance for each VPN and bind the VRF instance to an interface on an ASBR. This is because VPN routes cannot be directly transmitted between EBGP peers and can be carried only through MP-IBGP. If VPN routes can be directly transmitted between EBGP peers, you do not need to configure VRF instances on ASBRs. The Option B solution extends MP-IBGP to enable VPN routes to be directly distributed between ASBRs. The Option B solution is called the single-hop MP-EBGP solution. Figure 1-5 shows the topology.

Figure 1-5 Option B Inter-AS VPN



The advantage of this single-hop MP-EBGP solution is that you do not need to configure a sub-interface for each VPN site on an ASBR or set up an inter-AS LSP. The VPN routes can be directly transmitted between single-hop MP-EBGP neighbors. The Option B solution is applicable to networks with lots of inter-AS VPN services. The VPN routing information, however, is maintained and spread by the ASBRs between ASs. A large number of VPN routes generate a heavy workload on the ASBRs. Since the ASBRs also generally forward IP packets on the public network, high requirements are imposed on these devices. In addition, the ASBRs cancel the RT filtering function for received VPN routes. The VPN routes on PEs may be spread to the ASBRs in another AS, resulting in VPN route leakage. The USPs, who exchange VPN routes, must reach trust agreements on route exchanging. The ASBRs should trust each other and perform corresponding route filtering policies. This increases ISPs' O&M costs.

Option B supports two implementation solutions: The ASBRs do not change the next hop of a VPN route, and the ASBRs change the next hop of a VPN route. The following describes the configuration procedures of these two implementation solutions.

- Solution 1: ASBRs do not change the next hop of a VPN route.

  When an ASBR receives a VPN route sent from an ASBR in another AS and sends the route to an MP-IBGP neighbor in the local AS, the ASBR does not change the next hop of the VPN route. In this solution, the PEs and ASBRs in the same ASs establish MP-IBGP sessions to exchange VPN routes, and ASBRs in different ASs establish MP-EBGP sessions to directly exchange VPN routes. When an ASBR sends a route received from an MP-EBGP neighbor to another MP-IBGP neighbor, it does not change the next hop of the route. Therefore, the PE in the AS must have a route to the next hop (ASBR in another AS). An ASBR can redistribute a route destined for the peer ASBR to the IGP domain in the local AS so that the address of the ASBR in another AS becomes reachable. LSPs can be established via LDP.

- Solution 2: ASBRs change the next hop of a VPN route.

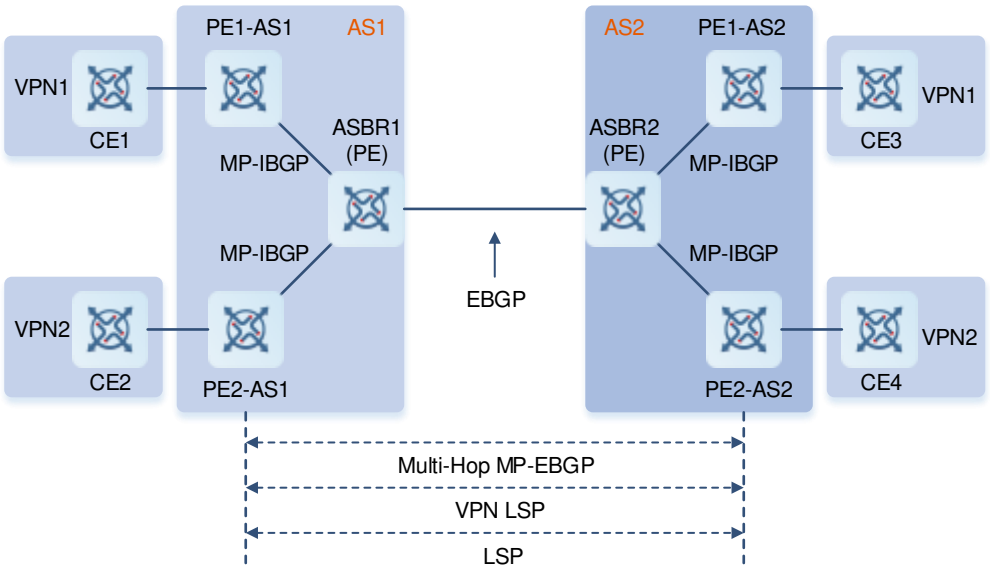
  When an ASBR receives a VPN route sent from an ASBR in another AS and sends the route to a PE in the local AS, the ASBR changes the next hop of the VPN route to the ASBR itself. In this solution, the PEs and ASBRs in the same ASs establish MP-IBGP sessions to exchange VPN routes, and ASBRs in different ASs establish MP-EBGP sessions to directly exchange VPN routes. When an ASBR receives a VPN route from

the other ASBR and advertises the VPN route to the MP-IBGP peer in the local AS, the ASBR changes the next hop of the VPN route to the ASBR itself.

3. Option C: Multi-Hop MP-EBGP Mode

Both Option A and Option B can meet inter-AS VPN networking requirements. However, the ASBRs need to maintain and advertise VPN routes. When each AS has considerable inter-AS VPN routes to be advertised, the ASBRs may become the bottleneck of further network expansion. To address this problem, the Option C solution is developed, that is, the multi-hop MP-EBGP solution. The multi-hop MP-EBGP solution enables PEs in different ASs to establish multi-hop MP-EBGP sessions to directly exchange VPN routes. In this mode, ASBRs do not need to maintain or distribute VPN routes.

Figure 1-6 Option C Multi-Hop MP-EBGP



In multi-hop MP-EBGP mode, only PEs rather than ASBRs are required to store VPN information. However, configuration of multi-hop MP-EBGP is complex. The solution is applicable to scenarios in which large-scale inter-AS VPN services are required.

To facilitate scale expansion in Option C, each AS is generally deployed with a route reflector (RR). The RRs in two ASs establish a multi-hop MP-EBGP session to exchange VPN routes. Judged from deployment, Option C can be referred to as the solution of multi-hop MP-EBGP session setup between RRs.

Option C has the following two implementation solutions:

- Solution 1: Enable label switching for IPv4 routes only between EBGP neighbors.

  In this solution, ASBRs need to run IGP (such as OSPF or RIP) to redistribute BGP routes so that each device in an AS has routes to PEs in another AS. In an AS, you can use the LDP to distribute labels for routes to PEs in another AS and set up LSPs. On the directly-connected ASBRs in two ASs, enable label switching for IPv4 routes. In this manner, BGP serves as the MPLS signaling protocol to distribute labels to routes destined to PEs in another AS and sets up inter-AS LSPs.

- Solution 2: Enable label switching for IPv4 routes between EBGP and IBGP neighbors.

  In solution 1, the IGP and LDP in one AS are required to maintain the PE routes in another AS. That is, inter-AS PE routes should be advertised to each device in the other AS. In view of the AS security in actual

applications, the PE routes of another AS are generally not advertised to each device in the local AS. These routes need to be owned only by the BGP and therefore are transparent to the IGP and LDP in the local AS. To achieve this, you can enable label switching for IPv4 routes between EBGP and IBGP neighbors.

This solution differs from solution 1 in that the IGP on an ASBR is not required to redistribute BGP routes and the LDP is not required to distribute labels to BGP routes and only needs to set up LSPs in the local AS. However, label switching for IPv4 routes needs to be enabled between both IBGP and EBGP neighbors to set up inter-AS LSPs. In addition, PEs are required to push three consecutive layers of labels.

1.1.4 OSPF VPN Extended Features

Open Shortest Path First (OSPF) is a widely used IGP. In most of the existing application scenarios, VPN users select OSPF as the internal routing protocol. If OSPF is also used between a PE and a CE, you do not need to run other routing protocols. This simplifies CE configuration and management.

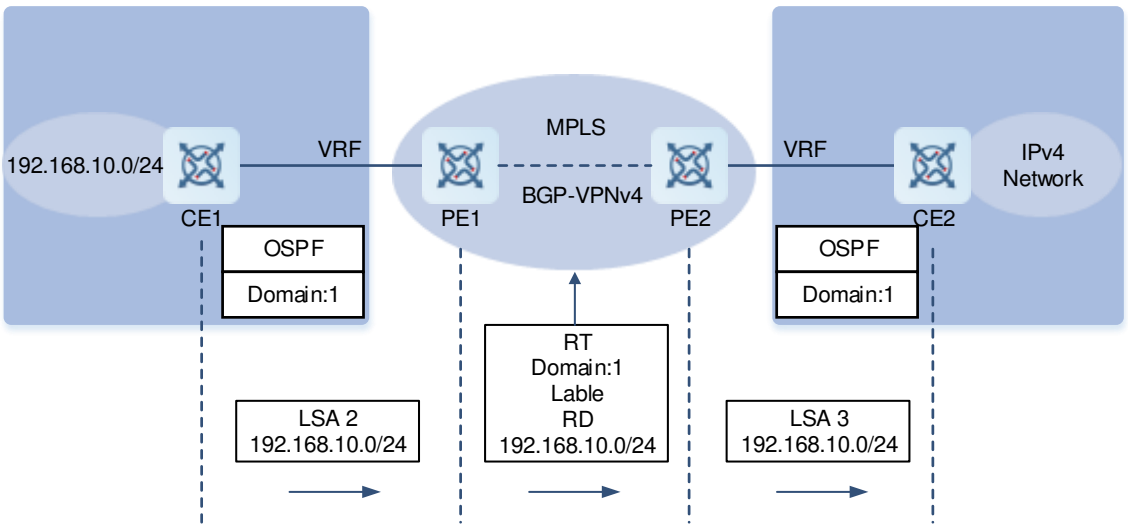
The following describes OSPF extended features between PEs and CEs.

1. Domain ID

A domain ID refers to the ID of an OSPF domain to which a route belongs. When a CE learns an OSPF route in a VPN site and this route is advertised to a PE as a type 1/2/3 link-state advertisement (LSA) and is redistributed to the BGP domain to form a VPN route, the domain ID is also redistributed to the BGP domain along with the route and advertised as an extended community attribute of the VPN route. When another PE receives this VPN route and redistributes it to a VRF-associated OSPF process, the domain ID is redistributed to the VRF-associated OSPF process along with the route. If the VRF-associated OSPF process confirms that the domain ID in the route is the same as that of the local VRF-associated OSPF process, it advertises the route to the CE as an internal route. If the VRF-associated OSPF process confirms that the domain ID in the route is different from that of the local VRF-associated OSPF process, the VRF-associated OSPF process advertises the route to the CE as an external route.

As shown in Figure 1-7, CE1 advertises routes to the same OSPF domain to PE1 as type 2 LSAs. PE1 converts them into VPN routes and advertises them to PE2. After receiving the routes, PE2 redistributes them to the VRF-associated OSPF process. The domain ID of the VRF-associated OSPF process is the same as that of the VPN routes. Therefore, the sites with the same domain ID are advertised to the VPN site as internal routes.

Figure 1-7 Domain ID

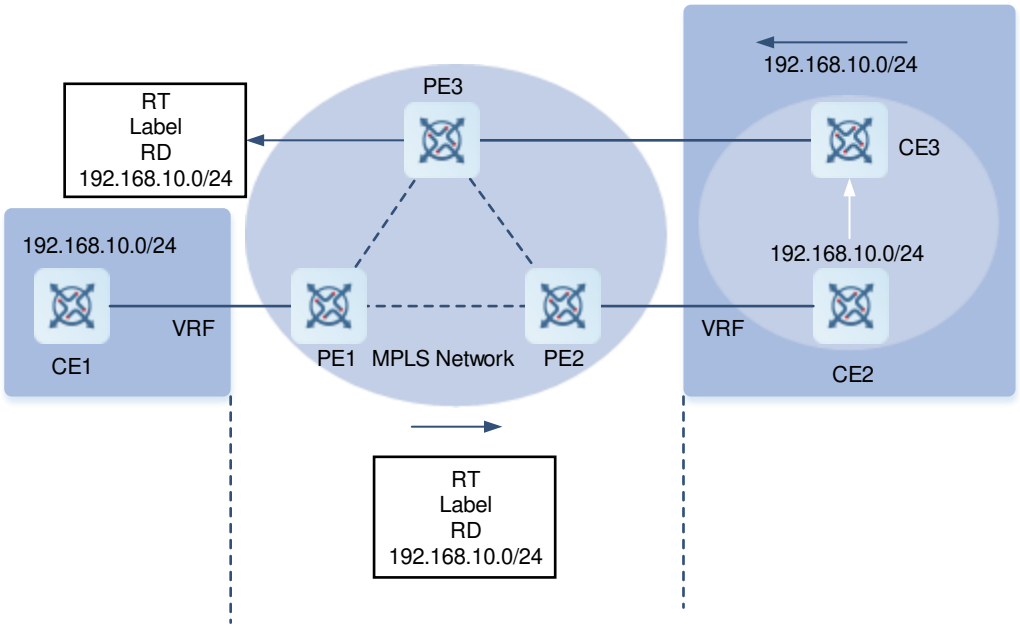


2. DN Bit

The DN bit is a loop detection technology between a PE and a CE running OSPF. In some scenarios, loops may arise when OSPF runs between a PE and a CE. For example, multiple PEs are connected to one VPN site. If one PE advertises learned VPN routes to the VPN site, which advertises the routes to another PE via OSPF, a loop may occur.

As shown in Figure 1-8, PE1 advertises the 192.168.10.0/24 route to PE2 and PE3, CE2 advertises the route to CE3 via OSPF, and CE3 advertises the route to PE3. PE3 preferentially selects the route that is redistributed by OSPF instead of BGP, converts this route into a VPNv4 route, and advertises it. As a result, a loop may occur.

Figure 1-8 DN Bit



To prevent such possible loops, the DN bit is set in an optional field of a type 3/5/7 LSA advertised from a PE to a CE. If another PE receives an LSA with the DN bit contained in an optional field, the OSPF of the PE will not use this LSA for OSPF route calculation.

3. VPN Route Tag

The VPN router tag is another loop detection technology between a PE and a CE running OSPF. When OSPF runs between a PE and a CE, the VRF-associated OSPF process of the PE has a router tag by default, which is called the VPN router tag. When the VRF-associated OSPF process of the PE imports a VPN route, converts it into a type 5/7 LSA, and advertises it to a CE, the LSA carries the VPN router tag. When multiple PEs are connected to one VPN site, the type 5/7 LSA received by a PE contains the VPN router tag, and the VPN router tag is the same as that of the OSPF process, the LSA is not used for OSPF route calculation.

4. Area Deployment

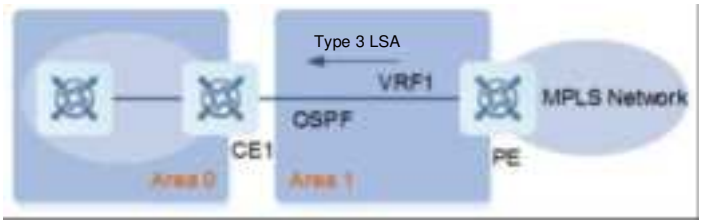
In normal cases, the link between a PE and a CE may belong to any OSPF area. If it belongs to a non-zero area, the PE is an ABR for the OSPF area where the CE resides. This may cause problems because the ABR running OSPF has the following features:

- The ABR calculates type 3 LSAs in the backbone area only.

- The ABR forwards only type 3 LSAs in the backbone area to a non-backbone area.

As shown in Figure 1-9, if the link between the PE and CE1 belongs to a non-zero area, the PE redistributes the VPNv4 routes advertised by MP-BGP neighbors to the OSPF domain, restores them to type 3 LSAs, and advertises them to CE1. CE1 does not calculate LSAs in non-backbone areas. These LSAs are not advertised to routers in Area 0, and sites of a VPN cannot learn routes of other sites. Therefore, exercise caution during OSPF area deployment when the link between a PE and a CE belongs to a non-zero area.

Figure 1-9 Area Deployment

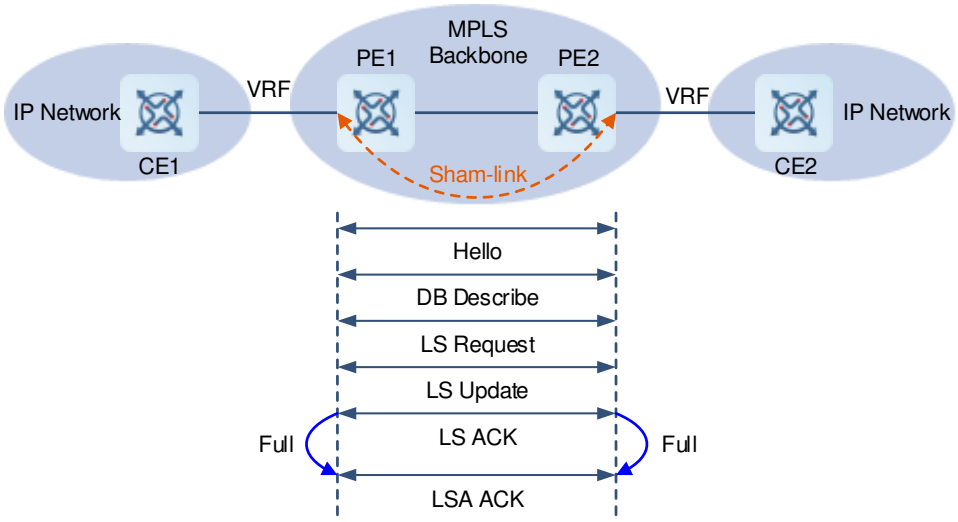


In general, if OSPF is run between a PE and a CE to exchange VPN routes in L3VPN applications, do not deploy backbone areas at VPN internal sites. If a router at a VPN internal site belongs to a backbone area in addition to a PE, at least one router at the VPN internal site must be connected to the PE and the link between the CE and PE must belong to Area 0. In this way, inter-area routes and external routes can be transmitted between the PE and the VPN site.

5. Sham Link

A sham link is not a real link but a virtual link established between VRF instances of two PEs. Like a normal OSPF link, a sham link has its OSPF interfaces and can send OSPF packets, establish neighbor relationships, and send LSAs. When LSAs are flooded on a sham link, the types of all OSPF routes do not change, as shown in Figure 1-10.

Figure 1-10 Sham Link



The purposes of establishing sham links between VRF-associated OSPF processes of different PEs are as follows:

- When the MP-IBGP is used to carry private routes, it only transfers routes. After the routes reach the peer PE and are restored, the MP-IBGP imports the original OSPF routes in a best-effort manner, and the OSPF topology information cannot be communicated properly. With a sham link, an OSPF link can be established to interconnect OSPF processes at each site and establish a complete topology.
- Different sites in the same VPN exchange information through the MPLS backbone network. However, a link is connected between these sites within the VPN to ensure that these VPN sites can communicate with each other through this link when the MPLS backbone network is unavailable. This link is called a backdoor link. If two VPN sites belong to the same OSPF area and one backdoor link is connected between the sites, routes inside the two sites are exchanged through either the MPLS backbone network or the backdoor link. Routes exchanged through the MPLS backbone network are inter-domain routes while routes exchanged through the backdoor link are intra-domain routes. The intra-domain routes advertised by the backdoor link are prior to the inter-domain routes advertised by the MPLS backbone network. Therefore, routes inside two sites are preferentially forwarded through the backdoor link, which is against the intention of the backdoor link connection for VPN users. In this case, a sham link is also required.

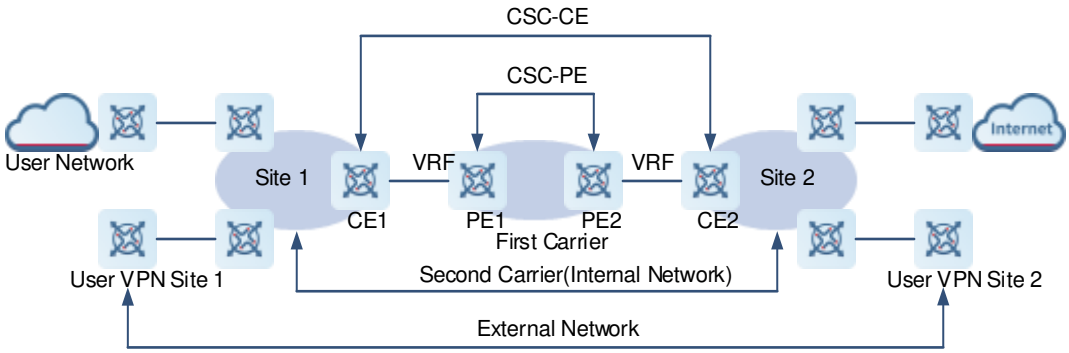
1.1.5 CSC Service Model

In the basic MPLS VPN, each site is a traditional IP network with simple network structure. However, some special VPN users are also service providers and they rent the VPN service from a MPLS VPN service provider to offer users with specific services. In this case, the MPLS VPN service provider is called the provider carrier or first carrier, and the VPN user who is also a service provider is called the customer carrier or second carrier. This networking model is called the Carrier's Carrier (CSC) model.

As shown in Figure 1-11, the user network and user VPN sites are connected to the second carrier, and the second carrier VPN sites are connected to the first carrier.

- LDP and BGP run between site 1 and site 2 to implement basic L3VPN features for the second carrier.
- First carrier features are implemented between PE1 and PE2.

Figure 1-11 CSC Model



1. Concepts of CSC Model

- First carrier

  First carrier is also called provider carrier that provides MPLS VPN services to second carriers. To allow a second carrier to provide services to its own users, the PEs of the first carrier must support the CSC technology. A PE of the first carrier that provides the CSC service to second carriers is called CSC-PE.

- Second carrier

  Second carrier is also called customer carrier that rents MPLS L3VPN services from a first carrier to build its own internal network and provide services to users over its own network. A CE of the second carrier that connects to the first carrier is called CSC-CE.

- Internal route

  Internal routes are routes of the internal network of the second carrier. Internal routes are used to implement interconnection between internal networks of second carriers and are maintained by the PE of the first carrier and the second carriers.

- External route

  As a service provider, the internal network of a second carrier may be connected to multiple third-party networks. The routes from the second carrier to the third-party networks are called external routes. If the second carrier provides traditional IP services to users, the external routes include routes of the user network. If the second carrier is connected to the Internet, the external routes include Internet routes. If the second carrier provides MPLS VPN services to users, the external routes include the VPN routes of users.

  Generally, there are a large number of external routes. To maintain fine scalability, the first carrier does not maintain external routes and the second carriers maintain external routes by themselves.

- VPN tunnel

  VPN tunnels are LSP tunnels established between private network devices. In the CSC model, LSP tunnels between the devices of second carriers are VPN tunnels.

- Route and label distribution between PEs and CEs

  To support a VPN tunnel, a PE of the first carrier (CSC-PE) and a CE of a second carrier (CSC-CE) must distribute label binding information to each other. The routing protocol for exchanging internal routes and the protocol for distributing labels for the internal routes vary depending on whether the CSC-PE and CSC-CE are in the same AS.

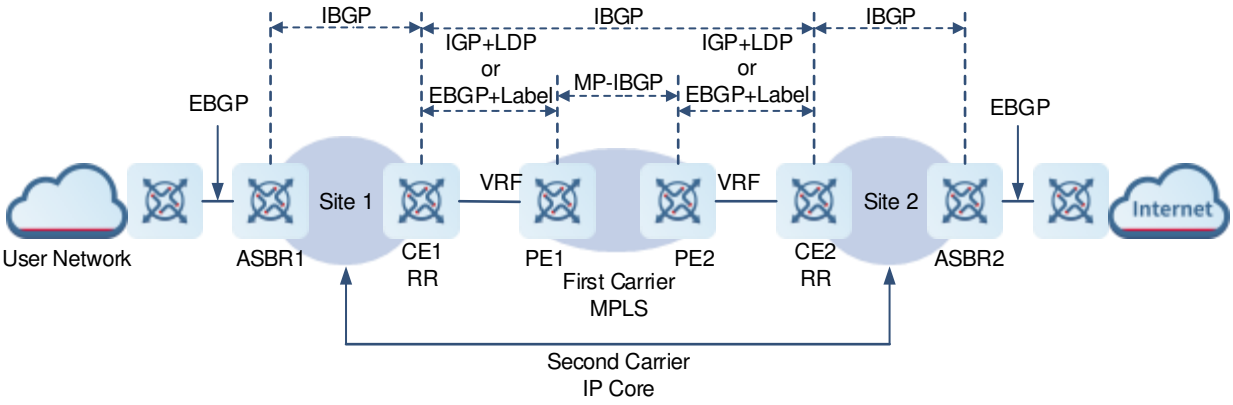
  - If the CSC-PE and CSC-CE are in the same AS, IGP is used for exchanging internal routes and LDP is used for exchanging label binding information.
  - If the CSC-PE and CSC-CE are in different ASs, EBGP is used for exchanging internal routes and exchanging labels for internal IPv4 internal routes.

2. Application Scenarios of CSC Model

A second carrier may be a common ISP or an MPLS service provider.

- Second carrier with IP core

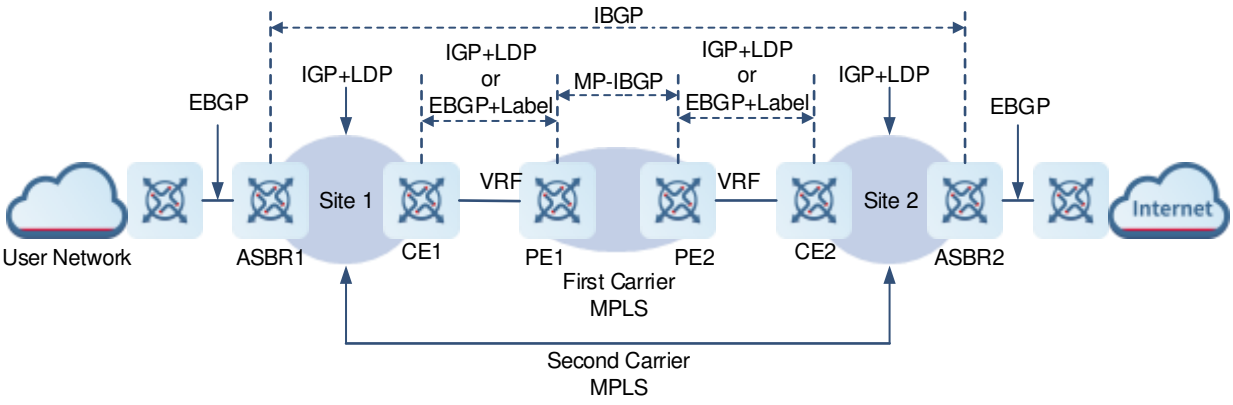
Figure 1-12 Second Carrier with IP Core



As shown in Figure 1-12, the second carrier uses an IP core network to provide network access services to users. ASBR1, ASBR2, CE1, and CE2 establish IBGP neighbor relationships and exchange external routes. CE1 and CE2 are route reflectors (RRs) that reflect external routes between sites. Internet access traffic of users flows into the second carrier network from ASBR1 and out of the second carrier network from ASBR2. Traffic from CE1 is forwarded over the private network LSP tunnel to reach CE2.

- Second carrier with MPLS core

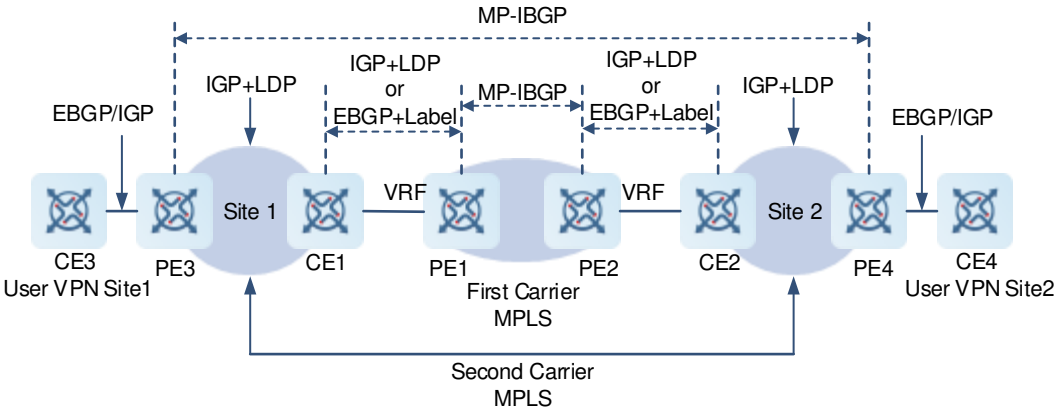
Figure 1-13 Second Carrier with MPLS Core



As shown in Figure 1-13, the second carrier uses an MPLS core network to provide network access services to users. ASBR1 and ASBR2 establish IBGP neighbor relationships and exchange external routes. Internet access traffic of users flows into the second carrier network from ASBR1 and out of the second carrier network from ASBR2. Traffic from ASBR1 is forwarded over the private network LSP tunnel to reach ASBR2.

- Second VPN provider with MPLS core

Figure 1-14 Second VPN Provider with MPLS Core



As shown in Figure 1-14, the second carrier uses an MPLS core network to provide MPLS L3VPN services to users. PE3 and PE4 establish MP-IBGP neighbor relationships and exchange VPN routes of users. The private network LSP between PE3 and PE4 is used as the outer tunnel of the user VPN.

1.1.6 6VPE Service Model

IPv6 VPN Provider Edge Router (6VPE) is a technology that uses IPv4 BGP/MPLS VPN to provide VPN services for IPv6 networks. In 6VPE mode, CEs use addresses in the IPv6 address family and the MPLS backbone network is still an IPv4 network. 6VPE is an IPv6 extension of IPv4 BGP/MPLS VPN.

The 6VPE technology helps retain the existing network base and upgrades PEs to smoothly transit from IPv4 access to IPv6 access and from IPv4 VPN services to IPv6 VPN services. Regardless of the network or service, 6VPE is an IPv4-to-IPv6 evolution solution with less investment.

In addition to IPv4 MPLS backbone networks, 6VPE supports IPv6 MPLS backbone networks. The two have similar principles and configurations. This document describes 6VPE in case of IPv4 MPLS backbone networks.

1. Route Distribution Process

The route distribution process in the control plane of a 6VPE device includes the following steps:

(1) Routing relationship between PEs and Ps are established using IPv4 IGP (such as RIP, OSPFv2, IS-IS, and BGP) or IPv6 IGP (such as RIPng and OSPFv3) to advertise the loopback address of the 6VPE device to all Ps and other 6VPE devices on the network.

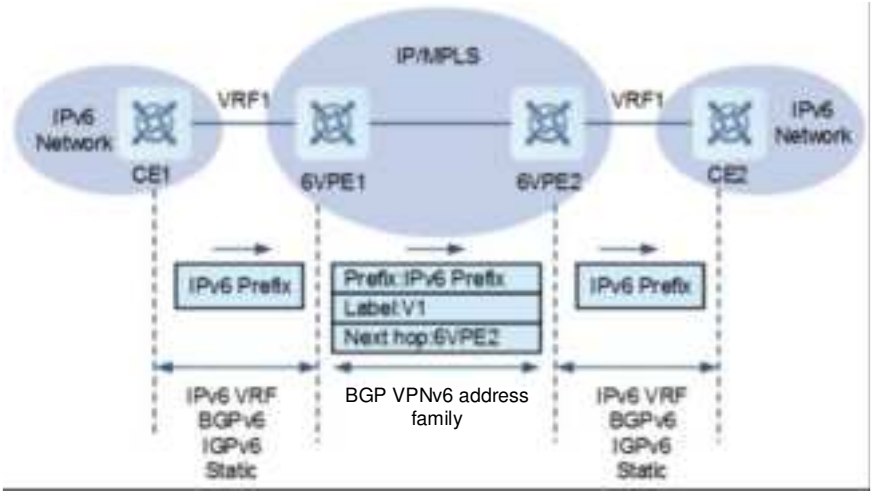
(2) The IPv4 or IPv6 label distribution protocol (such as LDP) is used to establish LSPs, that is, MPLS tunnels between 6VPE devices.

(3) 6VPE devices exchange routing information (IPv6 routing protocol or statically configured routes) between VRF instances and IPv6 CEs.

(4) Through BGP extended attributes, 6VPE devices advertise route reachability information and distribute labels for address prefixes in VPNs. If a 6VPE device connects to a user network using IPv6, the route reachability information uses the newly defined VPNv6 address family.

(5) 6VPE devices advertise IPv6 route reachability information to CEs using an IPv6 routing protocol.

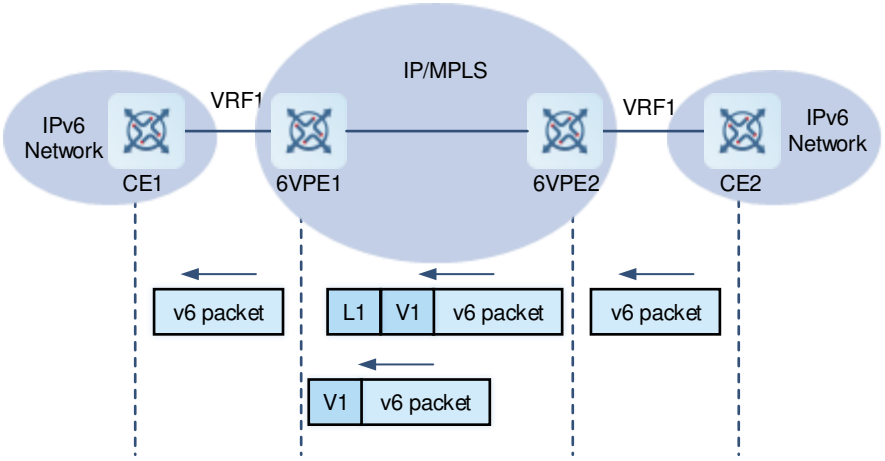
Figure 1-15 Control Plane Exchange



2. Packet Forwarding Process

When forwarding IPv6 packets of a VRF instance, a 6VPE device encapsulates MPLS labels for packets that need to enter the backbone network tunnels and forwards them to the egress 6VPE device along the LSP.

Figure 1-16 Forwarding Plane Exchange



1.1.7 6PE Service Model

IPv6 Provider Edge (6PE) is an IPv6 transition technology that allows CEs in IPv6 islands (non-VPN service) to communicate with each other through the existing IPv4 MPLS backbone network.

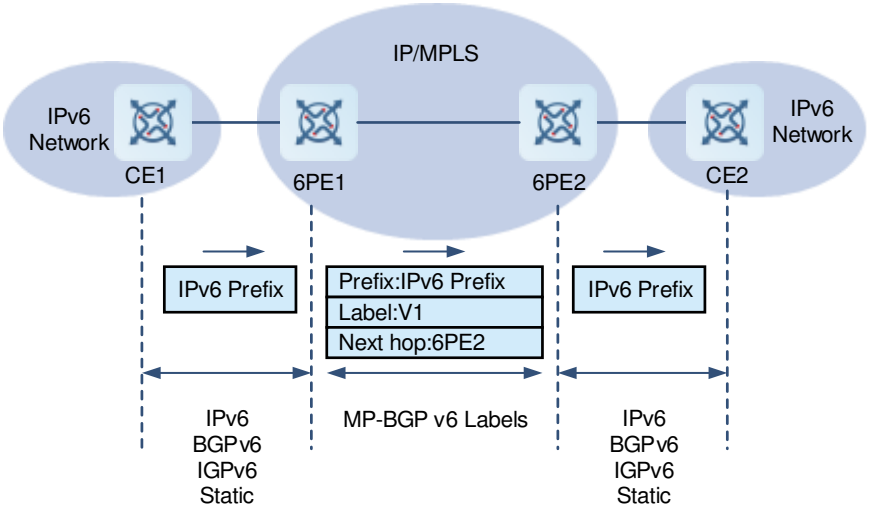
With the 6PE technology, ISPs can use existing IPv4 MPLS backbone networks to provide access services for scattered IPv6 networks. ISPs need to perform IPv6 upgrade only for PEs and do not need to upgrade or reconfigure the existing IPv4 MPLS backbone networks. This protects investment on existing backbone networks. For ISPs, the 6PE technology is an efficient and low-risky solution for IPv6 transition.

In addition to IPv4 MPLS backbone networks, 6PE supports IPv6 MPLS backbone networks.

1. Route Distribution Process

As shown in Figure 1-17, PEs and CEs run IPv6 and have IPv6 connections established. The backbone network is an IP MPLS network (IPv4 or IPv6 MPLS). This document uses the IPv4 MPLS network as an example. 6PE devices learn IPv6 routes from CEs, distribute labels to the IPv6 routes, and transfer the routes to other 6PE devices through BGP sessions.

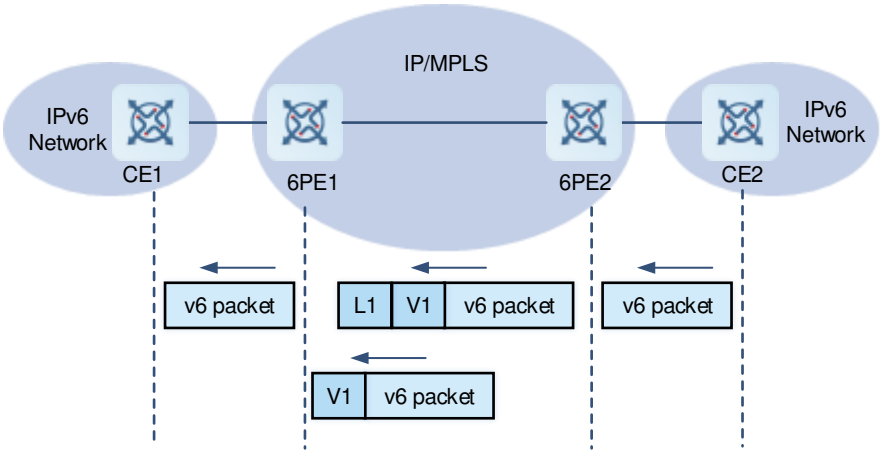
Figure 1-17 Route Distribution



2. Packet Forwarding Process

When forwarding IPv6 packets, a 6PE device encapsulates MPLS labels for packets that need to enter the backbone network tunnels, and forwards them to the egress PE along the LSP.

Figure 1-18 Packet Forwarding



1.1.8 Protocols and Standards

- RFC 4364: BGP/MPLS IP Virtual Private Networks (VPNs)
- RFC 4659: BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN
- RFC 4577: OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)

- RFC 4798: 6PE

1.2 IPv4 MPLS L3VPN Configuration Task Summary

IPv4 MPLS L3VPN configuration includes the following tasks:

(1) Configuring Basic IPv4 MPLS L3VPN Functions

a Configuring an MPLS Network

b Configuring a VPN Routing Instance

c Configuring VPN Route Exchange Between PEs

d Configuring VPN Route Exchange Between PEs and CEs

e (Optional) Configuring the Label Distribution Mode for VPN Routes

f (Optional) Configuring the Import and Export Policies for VPN Routes

g (Optional) Configuring Static L3VPN FTN and ILM

(2) (Optional) Configuring the Inter-AS VPN Service Model – Option A

(3) (Optional) Configuring the Inter-AS VPN Service Model – Option B (ASBRs Do Not Change the Next Hops of VPN Routes)

a Configuring Route Exchange Between PEs and CEs

b Configuring IGP and MPLS Signaling Protocol in an AS

c Configuring an ASBR to Cancel the Default RT Filtering Function

d Configuring PEs and ASBRs in the Same AS to Exchange VPN Routing Information

e Establishing an MP-EBGP Session Between ASBRs in Different ASs

f (Optional) Configuring Route Map Rules to Filter VPN Routes

g Configuring an IGP to Redistribute ASBR Routes of Another AS

(4) (Optional) Configuring Inter-AS VPN Service Model – Option B (ASBRs Change the Next Hops of VPN Routes)

a Configuring Route Exchange Between PEs and CEs

b Configuring IGP and MPLS Signaling Protocol in an AS

c Configuring an ASBR to Cancel the Default RT Filtering Function

d Establishing an MP-IBGP Session Between an ASBR and a PE and Changing the Next Hop Address to the ASBR Address

e Establishing an MP-EBGP Session Between ASBRs

f (Optional) Configuring Route Map Rules to Filter VPN Routes

(5) (Optional) Configuring Inter-AS VPN Service Model – Option C (Enabling Label Switching for IPv4 Routes Only with EBGP Neighbors)

a Configuring Route Exchange Between PEs and CEs in the Same ASs

b Configuring IGP and MPLS Signaling Protocol in an AS

c Establishing an EBGP Session Between ASBRs to Distribute Labels to IPv4 Routes

d Configuring an ASBR to Redistribute PE Routes Learned from the EBGP Domain to the IGP Domain

e Configuring a Multi-Hop MP-EBGP Session

(6) (Optional) Configuring the Inter-AS VPN Service Model – Option C (Enabling Label Switching for IPv4 Routes with EBGP and IBGP Neighbors)

a Configuring Route Exchange Between PEs and CEs in the Same ASs

b Configuring IGP and MPLS Signaling Protocol in an AS

c Establishing an IBGP Session Between a PE and an ASBR to Distribute Labels to IPv4 Routes

d Establishing an EBGP Session Between ASBRs to Distribute Labels to IPv4 Routes

e Configuring a Multi-Hop MP-EBGP Session

(7) (Optional) Configuring the Inter-AS VPN Service Model – Option C (Establishing a Multi-Hop MP-EBGP Session Between RRs)

a Configuring Route Exchange Between PEs and CEs

b Configuring IGP and MPLS Signaling Protocol in an AS

c Establishing an MP-IBGP Session Between an RR and a PE and Enabling Label Switching for IPv4 Routes

d Establishing an IBGP Session Between an RR and an ASBR and Enabling Label Switching for IPv4 Routes

e Establishing an EBGP Session Between ASBRs to Distribute Labels for IPv4 Routes

f Configuring a Multi-Hop MP-EBGP Session

(8) (Optional) Configuring OSPF VPN Extended Features

a Configuring the Domain ID

b Configuring the VPN Route Tag

c Configuring a Sham Link

d Configuring Loop Detection for a VRF-associated OSPF Process

e Configuring Extended Community Attributes of VPN Routes

f Disabling Loop Detection Based on the DN Bit Carried in LSAs

g Disabling Loop Detection Based on the Route Tag Carried in LSAs

1.3 Configuring Basic IPv4 MPLS L3VPN Functions

1.3.1 Overview

After basic IPv4 MPLS L3VPN functions are configured, BGP/MPLS VPN services can be provided in an AS on an ISP's network.

1.3.2 Restrictions and Guidelines

- MPLS L3VPN supports only Layer 3 routing interfaces.
- The LDP router ID must be 32 bits.
- The BGP router ID must be 32 bits.
- The **mpls ldp enable** and **label-switching** commands must be configured for PE interfaces used to connect to the public network.

1.3.3 Configuration Tasks

Configuration of basic IPv4 MPLS L3VPN functions includes the following tasks:

(1) Configuring an MPLS Network

(2) Configuring a VPN Routing Instance

(3) Configuring VPN Route Exchange Between PEs

(4) Configure VPN route exchange between PEs and CEs. Configure one of the following tasks.

- Running BGP Between PEs and CEs
- Running OSPF Between PEs and CEs
- Running RIP Between PEs and CEs
- Configuring Static Routes Between PEs and CEs

(5) (Optional) Configuring the Label Distribution Mode for VPN Routes

(6) (Optional) Configuring the Import and Export Policies for VPN Routes

(7) (Optional) Configuring Static L3VPN FTN and ILM

1.3.4 Configuring an MPLS Network

1. Overview

To use MPLS on the backbone network, you must configure the MPLS LDP on the Ps and PEs to establish public tunnels. This means that you have to configure LDP on MPLS devices and enable MPLS forwarding on each interface.

2. Procedure on Ps and PEs

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable MPLS forwarding globally.

mpls enable

MPLS forwarding is disabled globally by default.

To implement MPLS forwarding on a device, enable MPLS forwarding globally first.

(4) Enable LDP and enter the LDP configuration mode.

mpls router ldp

LDP is disabled by default.

(5) Configure the LDP router ID.

ldp router-id { *ipv4-address* | **interface** *interface-type interface-number* [**force**] }

The system router ID is used as the LDP router ID by default.

Generally, the loopback interface address is used as the router ID. If **force** is specified, the new router ID is forced to take effect immediately. Otherwise, the new router ID does not take effect immediately.

(6) Exit the LDP configuration mode.

exit

(7) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

  interface *ethernet-type interface-number*

- Enter the Layer 3 link aggregation configuration mode.

  interface aggregateport *interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

  interface *ethernet-type interface-number.subnumber*

- Enter the Layer 3 aggregated sub-interface configuration mode.

  interface aggregateport *interface-number.subnumber*

- Enter the switch virtual interface (SVI) configuration mode.

  interface vlan *interface-number*

(8) Configure an IP address for the interface.

ip address *ipv4-address mask-length*

No IP address is configured for an interface by default.

(9) Enable labeled MPLS packet forwarding on the public interface.

label-switching

Forwarding labeled MPLS packets is disabled on an interface by default.

(10) Enable LDP on the interface.

mpls ldp enable

LDP is disabled on an interface by default.

1.3.5 Configuring a VPN Routing Instance

1. Overview

To configure an VPN routing instance, define a VRF instance, configure RD and RT values for the VRF instance, and associate the VRF instance with an interface.

2. Restrictions and Guidelines

- VRF instances need to be configured on PEs rather than CEs or Ps.
- If the RD value of a VRF instance is defined on a PE or the PE is enabled with BGP VRF, the RD value cannot be modified or deleted. You can only delete and re-create the VRF instance to set the RD value.
- VRF instances on a PE must have unique RD values.
- When the **ip vrf forwarding** *vrf-name* command is run on an interface configured with an IP address, the configured IP address is deleted. In this case, you need to re-configure an IP address for the interface in interface configuration mode.

3. Procedure on PEs

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a VRF instance and enter the VRF instance configuration mode.

ip vrf *vrf-name*

No VRF instance is created by default.

(4) Configure the RD value.

rd *rd-value*

No RD value is configured by default.

(5) Configure the RT value.

route-target { **both** | **export** | **import** } *rt-value*

No RT value is configured by default.

(6) Exit the VRF instance configuration mode.

exit

(7) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

interface *ethernet-type interface-number*

- Enter the Layer 3 link aggregation configuration mode.

  interface aggregateport *interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

  interface *ethernet-type interface-number.subnumber*

- Enter the Layer 3 aggregated sub-interface configuration mode.

  interface aggregateport *interface-number.subnumber*

- Enter the SVI interface configuration mode.

  interface vlan *interface-number*

- Enter the tunnel interface configuration mode.

  interface tunnel *interface-number*

- Enter the loopback interface configuration mode.

  interface loopback *interface-number*

(8) Associate the VRF instance with an interface.

ip vrf forwarding *vrf-name*

An interface does not belong to any VRF instance by default.

(9) Configure an IP address for the interface.

ip address *ipv4-address mask-length*

No IP address is configured for an interface by default.

1.3.6 Configuring VPN Route Exchange Between PEs

1. Overview

PEs exchange routing information through BGP, and only common IPv4 routing information is exchanged by default. If a PE needs to exchange VPN routing information with another PE, enter the VPN address family configuration mode and enable VPN route exchange with peer PEs.

2. Procedure on PEs

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a BGP domain and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(4) Configure a BGP session.

neighbor *ipv4-address* **remote-as** *asn-number*

No BGP peer is configured by default.

(5) Configure the interface address used to establish the MP-IBGP session as the source address. Generally, the loopback interface address is used as the source address.

neighbor *ipv4-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

(6) Enter the BGP VPNv4 address family configuration mode.

address-family vpnv4

No VPNv4 address family is defined by default.

(7) Activate VPN route exchange in the BGP session.

neighbor *ipv4-address* **activate**

VPN route exchange is enabled in an IPv4 unicast address family by default.

1.3.7 Running BGP Between PEs and CEs

1. Overview

Establish BGP sessions between PEs and between PEs and CEs to connect to CEs in private networks using BGP. Redistribute private routes and BGP routes on CEs to ensure private network connectivity.

2. Restrictions and Guidelines

- During PE configuration in BGP IPv4 VRF address family configuration mode, if no RD value is specified for a VRF instance and **address-family ipv4 vrf** *vrf-nam*e is run to enter the address family configuration mode of the VRF instance, the system displays a prompt indicating that no RD value is configured and does not allow users to enter the address family configuration mode.
- Any dynamic or static routing protocol may run in users' private networks.

3. Procedure on PEs

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a BGP domain and enter the BGP configuration mode.

router bgp *pe-asn-number*

BGP is disabled by default.

(4) Configure the associated VRF instance and enter the BGP IPv4 VRF address family configuration mode.

address-family ipv4 **vrf** *vrf-name*

No IPv4 VRF address family is configured by default.

(5) Establish EBGP sessions with CEs.

neighbor *ipv4-address* **remote-as** *ce-asn-num*

No BGP peer is configured by default.

4. Procedure on CEs

On CEs, establish BGP sessions with PEs to import private network routes to BGP. Any dynamic or static routing protocol may run in users' private networks. OSPF is used as an example here.

(1) Enter the privileged EXEC mode.

enable

(1) Enter the global configuration mode.

configure terminal

(2) Create a BGP domain and enter the BGP configuration mode.

router bgp *ce-as-number*

BGP is disabled by default.

(3) Establish EBGP sessions with PEs.

neighbor *ipv4-address* **remote-as** *pe-as-number*

No BGP peer is configured by default.

(4) Configure BGP to redistribute OSPF routes.

redistribute ospf *ospf-id*

The route redistribution function is disabled by default.

(5) Exit the BGP configuration mode.

exit

(6) Enter the OSPF configuration mode.

router ospf *process–id*

The OSPF routing process is disabled by default.

(7) Configure OSPF to redistribute BGP routes.

redistribute bgp subnets

The route redistribution function is disabled by default.

1.3.8 Running OSPF Between PEs and CEs

1. Overview

If OSPF is run in users' private networks, continue to run OSPF between PEs and CEs without extra configuration on CEs.

To run OSPF between a PE and a CE, you must configure an OSPF process for the corresponding VRF instance on the PE. The PE then uses the OSPF process to exchange routing information with the CE. By redistributing BGP routes, the OSPF sends the VPN routes received from other PEs to the CE. By redistributing OSPF routes, the BGP sends the VPN routing information from the CE to the PE to other PE peers.

2. Restrictions and Guidelines

OSPF must be enabled for interfaces on CEs used to connect to PEs, and no passive ports or other route exchange restrictions are set.

3. Procedure on PEs

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create an OSPF process and enter the OSPF configuration mode.

router ospf *process-id* [**vrf** *vrf-name*]

The OSPF routing process is disabled by default.

(4) Configure links added to the OSPF area.

network *ipv4-address mask-length* **area** *area-id*

No interface IP address is added to the OSPF area by default.

(5) Configure OSPF to redistribute BGP routes.

redistribute bgp subnets

Route redistribution is not configured by default.

(6) Exit the OSPF configuration mode.

exit

(7) Enable BGP and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(8) Enter the BGP IPv4 VRF configuration mode.

address-family ipv4 **vrf** *vrf-name*

No IPv4 VRF address family is configured by default.

(9) Redistribute OSPF routes.

redistribute ospf *process-id*

The route redistribution function is disabled by default.

1.3.9 Running RIP Between PEs and CEs

1. Overview

If Routing Information Protocol (RIP) is run in users' private networks, continue to run RIP between PEs and CEs without extra configuration on CEs.

The VRF instance on a PE uses RIP to exchange routing information with a CE. By redistributing BGP routes, the RIP sends the VPN routes received from other PEs to the CE. By redistributing RIP routes, the BGP sends the VPN routing information from the CE to the PE to other PE peers.

2. Restrictions and Guidelines

RIP must be enabled for the interface on a CE used to connect to a PE, and route exchange is not restricted.

3. Procedure on PEs

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create an RIP instance and enter the RIP configuration mode.

router rip

(4) Enter the RIP IPv4 VRF address family configuration mode.

address-family ipv4 **vrf** *vrf-name*

No IPv4 VRF address family is configured by default.

(5) Configure the RIP version number.

version 2

By default, route update packets of RIPv1 and RIPv2 can be received, but only route update packets of RIPv1 are sent.

(6) Configure RIP used to communicate with a CE.

network *network-number* [*wildcard*]

No local network is advertised by default.

(7) Configure RIP to redistribute BGP routes.

redistribute bgp

Route redistribution is not configured by default.

(8) Exit the address family configuration mode.

exit

(9) Enable BGP and enter the BGP configuration mode.

router bgp as-number

BGP is disabled by default.

(10) Enter the BGP IPv4 VRF configuration mode.

address-family ipv4 **vrf** *vrf-name*

No IPv4 VRF address family is configured by default.

(11) Redistribute RIP routes.

redistribute rip

The route redistribution function is disabled by default.

1.3.10 Configuring Static Routes Between PEs and CEs

1. Overview

Configure a static route from a PE to a CE and redistribute the static route to the BGP VRF instance. Configure a static route from a CE to a PE and import the static route to the private network. In simple network environments, static routes are generally configured.

2. Procedure on PEs

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure a static route.

ip route [**vrf** *vrf-name*] *network mask* { *ipv4-address* | *interface* [*ipv4-address*] } [*distance*] [**tag** *tag*] [**permanent** | { **track** *object-number* | **arp** }] [**weight** *number*] [**description** *description-text*] [**disabled** | **enabled**] [**global**]

No static route is configured by default.

(4) Enter the BGP configuration mode.

r**outer bgp** as-number

BGP is disabled by default.

(5) Enter the BGP IPv4 VRF address family configuration mode.

address-family ipv4 **vrf** *vrf-name*

No IPv4 VRF address family is configured by default.

(6) Redistribute static routes.

redistribute static

The route redistribution function is disabled by default.

1.3.11 Configuring the Label Distribution Mode for VPN Routes

1. Overview

Two label distribution modes are used in L3VPN applications: route-based and VRF-based label distribution. Route-based label distribution features a fast forwarding speed. A device only needs to query the ILM table to forward packets to the next hop. However, the ILM table needs to have a large capacity. VRF-based label distribution requires only one label distributed to each VRF instance, and all routes in the VRF instance share

the label. This reduces the ILM table capacity. However, the forwarding speed is slow. This is because the system needs to look up the table twice in the forwarding process. It first looks up the ILM table to find the VRF instance where the packet is located, and then forwards the packet based on the destination IP address in the routing table of the VRF instance.

2. Restrictions and Guidelines

- When you modify the label distribution mode, the MP-BGP cancels all routes advertised in the VPN and re-advertises the routes.
- In VRF instance configuration mode, you can run the **alloc-label** command to modify the default label distribution mode. You can also choose different distribution modes for different VRF instances. **per-vrf** indicates that a label is distributed to all routes of a VRF instance. When advertising VPN routes, the MP-BGP uses the same label for all routes. **per-route** indicates that a label is distributed to each route of a VRF instance. When advertising VPN routes, the MP-BGP uses a different label for each route.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a VRF instance and enter the VRF instance configuration mode.

ip vrf *vrf-name*

No VRF instance is created by default.

(4) Configure the label distribution mode for VPN routes.

alloc-label { **per-nexthop** | **per-vrf** | **per-route** }

An L3VPN adopts the VRF-based label distribution mode by default. When you modify the label distribution mode, the MP-BGP cancels all routes advertised in the VPN and re-advertises the routes.

1.3.12 Configuring the Import and Export Policies for VPN Routes

1. Overview

In most situations, you can define the import route-target attribute in VRF configuration mode to determine the routes to be imported into a VRF instance and define the export route-target attribute to determine the RTs to be carried in the routes. These configurations are valid to all routes. In certain application scenarios that require accurate control on the import and export of VPN routes, however, you need to adopt policies.

2. Restrictions and Guidelines

The rule defined by the **import map** command takes effect after the import extended community attributes defined in the VRF instance. That is, only after these routes match the extended community attributes defined by the **route-target import** command in the VRF instance, the VPN routes received from the remote device can be filtered again by rules defined by the **import map** command.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a VRF instance and enter the VRF instance configuration mode.

ip vrf *vrf-name*

No VRF instance is created by default.

(4) Configure a policy for importing remote VPN routes to the local VPN routes based on the rules defined in the route map.

import map *routemap-name*

(5) Configure the extended group attributes that the local end distributes to remote VPN routes based on the rules defined in the route map.

export map *routemap-name*

1.3.13 Configuring Static L3VPN FTN and ILM Entries

1. Overview

In most situations, the MP-BGP distributes labels to private routes and the public LSP is generated by running the LDP on a public network. You can also configure a static LSP to distribute labels to private routes and set up private LSPs.

2. Restrictions and Guidelines

The configured static private FTN and ILM entries take effect only after the corresponding public LSP is set up. To set up a public LSP, refer to 1.3.4 Configuring an MPLS Network. You can set up a public LSP through LDP or configure it statically.

3. Procedure on PEs

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure a static private FTN entry.

mpls static l3vpn-ftn vrf-name { ipv4-address/mask-length | ipv6-address/prefix-length } **out-label** label **remote-pe** remote-pe-ipv4-addres

No L3VPN FTN entry is configured by default.

If the egress of a forwarding equivalence class (FEC) is another PE, you must specify the private label and the egress PE. The address of the egress PE is then used to match the public LSP.

(4) Configure a static private FTN entry.

(IPv4 network)

mpls static l3vpn-ftn vrf-name ipv4-address/mask-length **local-forward nexthop** interface-type interface-number nexthop-ipv4-address

(IPv6 network)

mpls static l3vpn-ftn vrf-name ipv6-address/prefix-length **local-forward nexthop** interface-type interface-number nexthop-ipv6-address

No L3VPN FTN entry is configured by default.

If the egress of an FEC is the local PE, you must specify the outbound interface on the local PE and the next-hop address (the outbound interface and the next hop is generally in another VRF instance). You can use this command when the local PE has several VRF instances that belong to the same VPN.

(5) Configure an ILM entry for the L3VPN. You need to specify the incoming label, the outbound interface, and the next-hop address.

(IPv4 network)

mpls static ilm in-label *in-label* **forward-action pop-l3vpn-nexthop** *vrf-name* **nexthop** *interface-type interface-number nexthop-ipv4-address* **fec** *ipv4-address*/*mask-length*

(IPv6 network)

mpls static ilm in-label *in-label* **forward-action pop-l3vpn-nexthop** *vrf-name* **nexthop** *interface-type interface-number nexthop-ipv6-address* **fec** *ipv6-address*/*prefix-length*

No L3VPN ILM entry is configured by default.

1.4 Configuring the Inter-AS VPN Service Model – Option A

1.4.1 Overview

In Option A solution, ASBRs establish a VRF instance for each VPN that needs to traverse domains, and bind interfaces for these VRF instances. VRF instances between ASBRs exchange VPN routes by using these interfaces. With configuration in this section, you can create an MPLS L3VPN across multiple ASs.

1.4.2 Restrictions and Guidelines

- The Option A solution requires an ASBR to configure one interface (usually logical sub-interface) for each inter-AS VPN and bind the interface to the inter-AS VPN. The number of bound interfaces should be at least equivalent to the number of inter-AS VPNs, and the VPNs need to be configured one after another on the ASBR. Therefore, the extensibility is poor. The sub-interface creation for each VPN raises higher requirements for the ASBR. This solution is generally applicable to networks with few inter-AS VPNs.
- The LDP router ID must be 32 bits.
- The BGP router ID must be 32 bits.
- An ASBR needs to configure an interface for each VPN and bind the interface to the VPN.

1.4.3 Procedure

Configure Option A in the method similar to 1.3 Configuring Basic IPv4 MPLS L3VPN Functions.

1.5 Configuring the Inter-AS VPN Service Model – Option B (ASBRs Do Not Change the Next Hops of VPN Routes)

1.5.1 Overview

When an ASBR receives a VPN route sent from an ASBR in another AS and sends it to an MP-IBGP neighbor in the local AS, the ASBR does not change the next hop of the VPN route. This is called the Option B solution with the next hop unchanged. In this solution, the ASBR and PE in the local AS establish an MP-IBGP session to exchange VPN routes. An MP-EBGP session can also be established between two ASBRs in different ASs to directly exchange VPN routes. An ASBR does not change the next hop of a route received from an MP-EBGP neighbor when it sends the route to an MP-IBGP neighbor. Therefore, the route to the next hop (ASBR in another AS) must exist on the PE in the local AS. An ASBR can redistribute a route destined for the peer ASBR to the IGP domain in the local AS so that the address of the ASBR in another AS becomes reachable. LSPs can be established via LDP.

1.5.2 Configuration Tasks

Configuration of the inter-AS VPN service model Option B (VPN route next hop unchanged) includes the following tasks:

(1) Configuring Route Exchange Between PEs and CEs

(2) Configuring IGP and MPLS Signaling Protocol in an AS

(3) Configuring an ASBR to Cancel the Default RT Filtering Function

(4) Configuring PEs and ASBRs in the Same AS to Exchange VPN Routing Information

(5) Establishing an MP-EBGP Session Between ASBRs in Different ASs

(6) (Optional) Configuring Route Map Rules to Filter VPN Routes

(7) Configuring an IGP to Redistribute ASBR Routes of Another AS

1.5.3 Configuring Route Exchange Between PEs and CEs

The procedure is similar to "Configuring Route Exchange Between PEs and CEs" in 1.3 Configuring Basic IPv4 MPLS L3VPN Functions.

1.5.4 Configuring IGP and MPLS Signaling Protocol in an AS

The procedure is similar to 1.3.4 Configuring an MPLS Network.

1.5.5 Configuring an ASBR to Cancel the Default RT Filtering Function

1. Restrictions and Guidelines

By default, a PE rejects a VPN route sent by another PE (or ASBR) if the route is not imported by any VRF instance on the PE. To enable an ASBR to receive all VPN routes from other PEs (or ASBRs) no matter whether these routes are imported into a local VRF instance, you should disable this default RT filtering function on the ASBR.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP and enter the BGP configuration mode.

router bgp asn-number

BGP is disabled by default.

(4) Disable RT filtering.

no bgp default route-target filter

The RT filtering function is enabled by default.

1.5.6 Configuring PEs and ASBRs in the Same AS to Exchange VPN Routing Information

The procedure is similar to 1.3.6 Configuring VPN Route Exchange Between PEs.

1.5.7 Establishing an MP-EBGP Session Between ASBRs in Different ASs

1. Overview

Establish a directly-connected single-hop MP-EBGP session between two inter-AS ASBRs to advertise VPN routes.

2. Restrictions and Guidelines

- You must run the **label-switching** command to enable labeled MPLS packet forwarding on the interfaces that connect two ASBRs so that the links between the ASBRs can forward MPLS packets.
- If the ASBRs do not use directly-connected addresses to establish an MP-EBGP session but use the loopback interface addresses with 32-bit mask as the source addresses to establish an MP-EBGP session, you must run the **neighbor ebgp-multihop** command to enable the multi-hop EBGP function. In addition, you must configure a static route on each ASBR to the loopback interface addresses on the peer and enable LDP or configure a static FTN entry (with an outgoing label 3, indicating that the ASBR is the penultimate hop).

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(4) Configure an EBGP session between ASBRs.

neighbor *asbr-address* **remote-as** *asbr-asn-number*

No BGP peer is configured by default.

(5) Enter the BGP VPNv4 address family configuration mode.

address-family vpnv4

No VPNv4 address family is defined by default.

(6) Enable VPN route exchange with peers.

neighbor *asbr-address* **activate**

VPN route exchange is enabled in an IPv4 unicast address family by default.

1.5.8 Configuring Route Map Rules to Filter VPN Routes

1. Overview

In view of the AS security in actual applications, you can configure policies on ASBRs to send or receive only certain VPN routes. You can realize this purpose by filtering the RT extended community attributes of VPN routes. In addition, all VPN routes are saved because the default RT filtering function is disabled on the ASBRs. In this case, you can configure VPN route policies on the ASBRs to receive only inter-AS VPN routes sent from the local AS, lessening the capacity pressure of the ASBRs.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a rule for the extended community attribute list.

ip extcommunity-list standard [*extcommunity-name* | *extcommunity-number*] { **permit** | **deny** } **rt** *rt-value*

No standard extended community attribute list is configured by default.

(4) Create a route map rule and enter the route map configuration mode.

route-map *route-map-name* **permit** [*number*]

(5) Configure the RT matching rule for a route map.

match extcommunity [*extcommunity-name* | *extcommunity-number*]

No extended community attribute list is matched by default.

(6) Exit the route map configuration mode.

exit

(7) Enable BGP and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(8) Enter the BGP VPNv4 address family configuration mode.

address-family vpnv4

No VPNv4 address family is defined by default.

(9) Filter the VPN routes received from ASBRs in another AS.

neighbor *peer-address* **route-map** *route-map-name* **in**

(10) Filter the VPN routes sent to ASBRs in another AS.

neighbor *peer-address* **route-map** *route-map-name* **out**

1.5.9 Configuring an IGP to Redistribute ASBR Routes of Another AS

1. Overview

Since an ASBR does not change the next hops of VPN routes sent to the IBGP peer, the next hop addresses of VPN routes learned by the PE in the local AS are the ASBR addresses in another AS. Therefore, you must configure the PE to learn the routes to the next hop address. For a single-hop directly-connected MP-EBGP session between two ASBRs where BGP is enabled to carry labels (through IPv4 routes or VPN routes), the MP-BGP automatically generates a host route with 32-bit mask and an FTN entry (with the outgoing label 3) to the peer on each ASBR. In this manner, the tunnel egress is not terminated on the local ASBR. Therefore, as long as the ASBRs redistribute the host route to the IGP in the local AS, the PEs can learn routes to the ASBR in the other AS.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable an IGP, for example, OSPF, RIP, or IS-IS.

router *igp*

(4) Redistribute routes to a directly-connected network segment.

redistribute connected subnets

1.6 Configuring Inter-AS VPN Service Model – Option B (ASBRs Change the Next Hops of VPN Routes)

1.6.1 Overview

When an ASBR receives a VPN route sent from an ASBR in another AS and then sends it to a PE in the local AS, the ASBR changes the next hop of the VPN route to the ASBR itself. This is called the Option B solution with the next hop changed. In this solution, a PE and an ASBR in the same AS establish an MP-IBGP session to exchange VPN routes and an MP-EBGP session can be established between the two ASBRs to exchange VPN routes. When an ASBR receives a VPN route from the other ASBR and advertises the VPN route to the MP-IBGP peer in the local AS, the ASBR changes the next hop of the VPN route to the ASBR itself.

1.6.2 Restrictions and Guidelines

1.6.3 Configuration Tasks

Configuration of the inter-AS VPN Option B service model (VPN route next hop changed) includes the following tasks:

(1) Configuring Route Exchange Between PEs and CEs

(2) Configuring IGP and MPLS Signaling Protocol in an AS

(3) Configuring an ASBR to Cancel the Default RT Filtering Function

(4) Establishing an MP-IBGP Session Between an ASBR and a PE and Changing the Next Hop Address to the ASBR Address

(5) Establishing an MP-EBGP Session Between ASBRs

(6) (Optional) Configuring Route Map Rules to Filter VPN Routes

1.6.4 Configuring Route Exchange Between PEs and CEs

The procedure is similar to "Configuring Route Exchange Between PEs and CEs" in 1.3 Configuring Basic IPv4 MPLS L3VPN Functions.

1.6.5 Configuring IGP and MPLS Signaling Protocol in an AS

The procedure is similar to 1.3.4 Configuring an MPLS Network.

1.6.6 Configuring an ASBR to Cancel the Default RT Filtering Function

By default, a PE rejects a VPN route sent by another PE (or ASBR) if the route is not imported by any VRF instance on the PE. To enable an ASBR to receive all VPN routes from other PEs (or ASBRs) no matter whether these routes are imported into a local VRF instance, you should disable this default RT filtering function on the ASBR.

The procedure is similar to 1.5.5 Configuring an ASBR to Cancel the Default RT Filtering Function.

1.6.7 Establishing an MP-IBGP Session Between an ASBR and a PE and Changing the Next Hop Address to the ASBR Address

1. Restrictions and Guidelines

By default, an ASBR does not change the next hops of the VPN routes received from an MP-EBGP peer when the ASBR sends the routes to the MP-IBGP peer. You can configure the ASBR to forcibly change the next hops of the VPN routes to the ASBR itself. In this manner, the PEs in the local AS are not required to learn the address of the peer ASBR. This is the major difference with solution 1 (next hop unchanged).

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(4) Establish an IBGP session with a PE.

neighbor *pe-address* **remote-as** *as-number*

No BGP peer is configured by default.

(5) Specify the local loopback interface address as the source address to establish an IBGP session.

neighbor *pe-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

Generally, the loopback interface address is used as the source address.

(6) Enter the BGP VPNv4 address family configuration mode.

address-family vpnv4

No VPNv4 address family is defined by default.

(7) Enable VPN route exchange with peers.

neighbor *pe-address* **activate**

(8) Configure an ASBR to change the next hops to its own address when sending VPN routes to the IBGP neighbor.

neighbor *pe-address* **next-hop-self**

1.6.8 Establishing an MP-EBGP Session Between ASBRs

The procedure is similar to 1.5.7 Establishing an MP-EBGP Session Between ASBRs in Different ASs.

1.6.9 Configuring Route Map Rules to Filter VPN Routes

The procedure is similar to 1.5.8 Configuring Route Map Rules to Filter VPN Routes.

1.7 Configuring Inter-AS VPN Service Model – Option C (Enabling Label Switching for IPv4 Routes Only with EBGP Neighbors)

1.7.1 Overview

In this solution, ASBRs need to run IGP (such as OSPF or RIP) to redistribute BGP routes so that each device in an AS has routes to PEs in another AS. In an AS, you can use the LDP to distribute labels for routes to PEs in another AS and set up LSPs. On the directly-connected ASBRs in two ASs, enable label switching for IPv4 routes. BGP serves as the MPLS signaling protocol to distribute labels for routes to PEs in another AS and set up inter-AS LSPs.

1.7.2 Configuration Tasks

Configuration of the inter-AS VPN service model – Option C (enabling label switching for IPv4 routes only between EBGP neighbors) includes the following tasks:

(1) Configuring Route Exchange Between PEs and CEs in the Same ASs

(2) Configuring IGP and MPLS Signaling Protocol in an AS

(3) Establishing an EBGP Session Between ASBRs to Distribute Labels to IPv4 Routes

(4) Configuring an ASBR to Redistribute PE Routes Learned from the EBGP Domain to the IGP Domain

(5) Configuring a Multi-Hop MP-EBGP Session

1.7.3 Configuring Route Exchange Between PEs and CEs in the Same ASs

The procedure is similar to "Configuring Route Exchange Between PEs and CEs" in 1.3 Configuring Basic IPv4 MPLS L3VPN Functions.

1.7.4 Configuring IGP and MPLS Signaling Protocol in an AS

The procedure is similar to 1.3.4 Configuring an MPLS Network.

1.7.5 Establishing an EBGP Session Between ASBRs to Distribute Labels to IPv4 Routes

1. Overview

Establish an EBGP session between inter-AS ASBRs and enable label distribution for IPv4 routes. To import PE routes to the BGP domain, run the **network** command in the BGP IPv4 address family mode or run the IGP route redistribution command. In view of the AS security in actual applications, you are generally required to configure IPv4 route distribution policies on ASBRs. By configuring route map rules, you can control the routes sent to neighbors and specify whether the routes carry labels. Similar control is available for receiving routes.

2. Restrictions and Guidelines

- You must run the **label-switching** command to enable labeled MPLS packet forwarding on the interfaces that connect two ASBRs so that the links between the ASBRs can forward MPLS packets.
- If the ASBRs do not use directly-connected addresses but use the loopback interface addresses with 32-bit mask as the source addresses to establish an MP-EBGP session, you must run the **neighbor ebgp-multihop** command to enable the multi-hop EBGP function. In addition, you must configure a static route on each ASBR to the loopback interface addresses on the peer and enable LDP or configure a static FTN entry (with an outgoing label 3, indicating that the ASBR is the penultimate hop).
- In actual applications, an ASBR is generally required to distribute labels only to PE routes that carry inter-AS VPN services. You can run the **set mpls-label** command in route map mode to achieve this purpose. Create a route map rule and run the **set mpls-label** command in route map configuration mode to distribute labels to routes and advertise only inter-AS PE routes to the peer ASBR. For details about configurations related to the routing policies, see *Configuring Routing Policies*.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP and enter the BGP configuration mode.

router bgp *asn-num*ber

BGP is disabled by default.

(4) Establish an EBGP session with the ASBR in another AS.

neighbor *asbr-address* **remote-as** *asbr-asn-num*

No BGP peer is configured by default.

(5) Enter the BGP IPv4 address family configuration mode.

address-family ipv4

(6) Configure the ASBR to exchange labeled IPv4 routes with the ASBR peer in the other AS.

neighbor *asbr-address* **send-label**

Forwarding labeled MPLS packets is disabled by default.

(7) (Optional) Configure PE addresses to be imported into the BGP routing table in the local AS, that is, host routes to each PE in the AS.

network *pe-address* **mask** *mask*

(8) (Optional) Configure a route distribution policy to control the routes sent to neighbors by defining a route map rule.

neighbor *asbr-address* **route-map** *route-map-name* **out**

(9) (Optional) Configure a route distribution policy to receive only labeled routes by defining a route map rule.

neighbor *asbr-address* **route-map** *route-map-name* **in**

1.7.6 Configuring an ASBR to Redistribute PE Routes Learned from the EBGP Domain to the IGP Domain

1. Overview

When an ASBR learns a route to the PE in another AS from the peer ASBR and needs to inform other PEs in the local AS of the route and to set up an LSP to the PE in the other AS, the ASBR can redistribute routes learned from the EBGP domain to the IGP domain and enable the LDP to distribute labels to BGP routes.

2. Restrictions and Guidelines

You can run the **redistribute bgp subnets route-map** *routemap-name* command in IGP configuration mode to control routes learned from the EBGP domain that need to be redistributed to the IGP domain. You can also run the **advertise-labels for bgp-routes acl** *acl-name* command in LDP configuration mode to control the BGP routes for which labels are distributed.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the IGP configuration mode.

router *igp*

(4) Redistribute BGP routes. Route filtering by using route map rules is optional.

redistribute bgp subnets [**route-map** *routemap-name*]

(5) Exit the IGP configuration mode.

exit

(6) Enter the LDP configuration mode.

mpls router ldp

LDP is disabled by default.

(7) Configure the LDP router ID.

ldp router-id { *ipv4-address* | **interface** *interface-type interface-number* [**force**] }

The system router ID is used as the LDP router ID by default.

Generally, the loopback interface address is used as the router ID. If **force** is specified, the new router ID is forced to take effect immediately. Otherwise, the new router ID does not take effect immediately.

(8) Distribute labels to BGP routes. Route filtering by using ACL rules is optional.

advertise-labels for bgp-routes [**acl** *acl-name*]

By default, the LDP distributes labels only to IGP routes rather than BGP routes. To enable the LDP to distribute labels to BGP routes, you can run the **advertise-labels for bgp-routes** command.

1.7.7 Configuring a Multi-Hop MP-EBGP Session

1. Overview

Establish a multi-hop MP-EBGP session between a PE that needs inter-AS VPN services and a PE in another AS to exchange VPN routes.

2. Restrictions and Guidelines

In a multi-hop MP-EBGP session, IPv4 routes do not need to be exchanged or at least the routes of the two addresses used to set up the BGP session should not be exchanged. Otherwise, a PE has two routes to the PE in another AS. One route is advertised by the ASBR in the local AS, and the other is advertised by the multi-hop EBGP session. According to BGP specifications, an EBGP route has a higher priority over an IGBP route by default. In this case, the BGP preferentially chooses the route advertised by the multi-hop BGP session, and this results in flapping of routes on this PE to the PE in another AS and causes unreachable VPN routes.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(4) Establish a multi-hop EBGP session with a PE in another AS.

neighbor *ebgp-peer-address* **remote-as** *ebgp-as-number*

No BGP peer is configured by default.

(5) Use the loopback interface address to establish a neighbor relationship with the EBGP peer.

neighbor *ebgp-peer-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

Generally, the loopback interface address is used as the source address.

(6) Configure multi-hop attributes.

neighbor *ebgp-peer-address* **ebgp-multihop**

(7) Enter the BGP VPNv4 address family configuration mode.

address-family vpnv4

No VPNv4 address family is defined by default.

(8) Enable VPN route exchange with peers.

neighbor *ebgp-peer-address* **activate**

(9) Exit the BGP VPN address family configuration mode.

exit

(10) Enter the BGP IPv4 address family configuration mode.

address-family ipv4

(11) Disable the IPv4 route exchange function.

no neighbor *ebgp-peer-address* **activate**

1.8 Configuring the Inter-AS VPN Service Model – Option C (Enabling Label Switching for IPv4 Routes with EBGP and IBGP Neighbors)

1.8.1 Overview

In the Option C solution (enabling label switching for IPv4 routes only with EBGP neighbors), the IGP and LDP in one AS need to maintain the PE routes in another AS. That is, inter-AS PE routes should be advertised to each device in an AS. In view of the AS security in actual applications, the PE routes of another AS are generally not advertised to each device in the local AS. These routes need to be owned only by the BGP and therefore are transparent to the IGP and LDP in the local AS. To achieve this, you can enable label switching for IPv4 routes with EBGP and IBGP neighbors.

This solution differs from the Option C solution (enabling label switching for IPv4 routes only with EBGP neighbors) in that the IGP on an ASBR is not required to redistribute BGP routes, and the LDP is not required to distribute labels to BGP routes and only needs to set up LSPs in the local AS. However, you need to enable label switching for IPv4 routes with both IBGP and EBGP neighbors to set up inter-AS LSPs. In addition, the PEs are required to push three consecutive layers of labels.

1.8.2 Configuration Tasks

Configuration of the inter-AS VPN service model – Option C (enabling label switching for IPv4 routes with EBGP and IBGP neighbors) includes the following tasks:

(1) Configuring Route Exchange Between PEs and CEs in the Same ASs

(2) Configuring IGP and MPLS Signaling Protocol in an AS

(3) Establishing an IBGP Session Between a PE and an ASBR to Distribute Labels to IPv4 Routes

(4) Establishing an EBGP Session Between ASBRs to Distribute Labels to IPv4 Routes

(5) Configuring a Multi-Hop MP-EBGP Session

1.8.3 Configuring Route Exchange Between PEs and CEs in the Same ASs

The procedure is similar to "Configuring Route Exchange Between PEs and CEs" in 1.3 Configuring Basic IPv4 MPLS L3VPN Functions.

1.8.4 Configuring IGP and MPLS Signaling Protocol in an AS

The procedure is similar to 1.3.4 Configuring an MPLS Network.

1.8.5 Establishing an IBGP Session Between a PE and an ASBR to Distribute Labels to IPv4 Routes

1. Overview

This configuration procedure is the main difference between the inter-AS VPN Option C solution (enabling IPv4 route exchange between EBGP and IBGP neighbors) and the inter-AS VPN Option C solution (enabling IPv4 route exchange only between EBGP neighbors). The IBGP session between an ASBR and a PE is used to transmit the PE routes of another AS, and the BGP distributes labels to the PE routes. The PE routes that are learned by EBGP from another AS are not redistributed to the IGP in the local AS.

2. Restrictions and Guidelines

Before you enable label switching for IPv4 routes for an IBGP session with an IBGP peer, you must run the **neighbor update-source** command to specify the source address of the IBGP session, and this source address must be the address of the loopback interface. Otherwise, the inter-AS LSP cannot be set up.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

config terminal

(3) Enable BGP and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(4) Establish an IBGP session with an ASBR (PE).

neighbor *peer-address* **remote-as** *as-number*

No BGP peer is configured by default.

(5) Use the loopback interface address as the source address to establish the BGP session with an ASBR (PE) peer.

neighbor *peer-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

Generally, the loopback interface address is used as the source address.

(6) Enter the IPv4 address family configuration mode.

address-family ipv4

(7) Configure labeled IPv4 route exchange with an ASBR (PE) peer.

neighbor *peer-address* **send-label**

Forwarding labeled MPLS packets is disabled by default.

1.8.6 Establishing an EBGP Session Between ASBRs to Distribute Labels to IPv4 Routes

The procedure is similar to 1.7.5 Establishing an EBGP Session Between ASBRs to Distribute Labels to IPv4 Routes.

1.8.7 Configuring a Multi-Hop MP-EBGP Session

The procedure is similar to 1.7.7 Configuring a Multi-Hop MP-EBGP Session.

1.9 Configuring the Inter-AS VPN Service Model – Option C (Establishing a Multi-Hop MP-EBGP Session Between RRs)

1.9.1 Overview

In the traditional Option C solution, the inter-AS VPN sites should be connected in full mesh mode. The addition of a single VPN site requires the setup of MP-MBGP connections with the PEs in other ASs, hindering the expansion of VPN sites. To solve this problem, you can deploy an RR in each AS and establish multi-hop MP-EBGP sessions between the RRs to exchange VPN routes.

1.9.2 Configuration Tasks

Configuration of the inter-AS VPN service model – Option C (establishing a multi-hop MP-EBGP session between RRs) includes the following tasks:

(1) Configuring Route Exchange Between PEs and CEs

(2) Configuring IGP and MPLS Signaling Protocol in an AS

(3) Establishing an MP-IBGP Session Between an RR and a PE and Enabling Label Switching for IPv4 Routes

(4) Establishing an IBGP Session Between an RR and an ASBR and Enabling Label Switching for IPv4 Routes

(5) Establishing an EBGP Session Between ASBRs to Distribute Labels for IPv4 Routes

(6) Configuring a Multi-Hop MP-EBGP Session

1.9.3 Configuring Route Exchange Between PEs and CEs

The procedure is similar to "Configuring Route Exchange Between PEs and CEs" in 1.3 Configuring Basic IPv4 MPLS L3VPN Functions.

1.9.4 Configuring IGP and MPLS Signaling Protocol in an AS

The procedure is similar to 1.3.4 Configuring an MPLS Network.

1.9.5 Establishing an MP-IBGP Session Between an RR and a PE and Enabling Label Switching for IPv4 Routes

1. Overview

Establish an MP-IBGP session between a PE and an RR to exchange VPN routes and enable label switching for IPv4 routes in this session.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(4) Establish an IBGP session.

neighbor peer-address **remote-as** as-number

No BGP peer is configured by default.

(5) Use the loopback interface address as the source address to establish an IBGP session.

neighbor *peer-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

Generally, the loopback interface address is used as the source address.

(6) Enter the BGP IPv4 address family configuration mode.

address-family ipv4

(7) Enable IPv4 route exchange.

neighbor *peer-address* **activate**

(8) Enable label switching for IPv4 routes.

neighbor *peer-address* **send-label**

Forwarding labeled MPLS packets is disabled by default.

(9) On the RR, configure all PE peers as the IPv4 RR clients.

neighbor *peer-address* **route-reflector-client**

(10) Exit the BGP IPv4 address family configuration mode.

exit

(11) Enter the BGP VPNv4 address family configuration mode.

address-family vpnv4

No VPNv4 address family is defined by default.

(12) Enable VPN route exchange with peers.

neighbor *peer-address* **activate**

(13) On the RR, configure all PE peers as the VPN RR clients.

neighbor *peer-address* route-reflector-client

1.9.6 Establishing an IBGP Session Between an RR and an ASBR and Enabling Label Switching for IPv4 Routes

1. Overview

Establish an MP-IBGP session between an ASBR and an RR to receive routes to the PEs in the local AS from the RR and send routes to the PEs in another AS to the RR. In addition, enable label switching for IPv4 routes in the session.

2. Restrictions and Guidelines

For the IBGP session between an RR and an ASBR, you are generally not required to set the ASBR as the RR client unless the ASBR also serves as a PE.

3. Procedure

(1) Enter the global configuration mode.

config terminal

(2) Enable BGP and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(3) Establish an IBGP session.

neighbor *peer-address* **remote-as** *asn-number*

No BGP peer is configured by default.

(4) Use the loopback interface address as the source address to establish an IBGP session.

neighbor *peer-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

Generally, the loopback interface address is used as the source address.

(5) Enter the BGP IPv4 address family configuration mode.

address-family ipv4

(6) Enable IPv4 route exchange.

neighbor *peer-address* **activate**

(7) Enable label switching for IPv4 routes.

neighbor *peer-address* **send-label**

Forwarding labeled MPLS packets is disabled by default.

1.9.7 Establishing an EBGP Session Between ASBRs to Distribute Labels for IPv4 Routes

The procedure is similar to 1.7.5 Establishing an EBGP Session Between ASBRs to Distribute Labels to IPv4 Routes.

1.9.8 Configuring a Multi-Hop MP-EBGP Session

1. Overview

Establish a multi-hop MP-EBGP session between the RRs in two ASs to exchange inter-AS VPN routes. In addition, disable the transmission of IPv4 routing information for the session. The PE routes are advertised to another AS through an ASBR.

2. Restrictions and Guidelines

- When advertising a route to an EBGP peer, the device with EBGP enabled modifies the next hop of the route as its own address. Upon receipt of the VPN route, the PE in another AS considers the next hop of the route as the RR. As a result, all inter-AS VPN traffic is transmitted through the RR. This is not the optimal forwarding path and has high requirements on the forwarding performance of the RR. To avoid the preceding situation, you can run the **neighbor next-hop-unchanged** command in the VPNv4 address family mode to configure the ASBR not to change the next hop of a VPNv4 route sent to the BGP peer when you establish a multi-hop MP-EBGP session on the RR.
- In a multi-hop MP-EBGP session, IPv4 routes do not need to be exchanged or at least the routes of the two addresses used to establish the BGP session should not be exchanged. Otherwise, a PE has two routes to a PE in another AS. One route is advertised by the ASBR in the local AS, and the other is advertised by the multi-hop EBGP session. According to BGP specifications, an EBGP route has a higher priority over an IGBP route by default. In this case, the BGP preferentially chooses the route advertised by the multi-hop BGP and this results in flapping of routes on this PE to the PE in another AS and causes unreachable VPN routes.

3. Procedure

(1) Enter the global configuration mode.

config terminal

(2) Enable BGP and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(3) Establish an EBGP session.

neighbor rr-address **remote-as** ebgp-asn-number

No BGP peer is configured by default.

(4) Use the loopback interface address as the source address to establish an EBGP session.

neighbor *rr-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

Generally, the loopback interface address is used as the source address.

(5) Configure multi-hop EBGP attributes.

neighbor *rr-address* **ebgp-multihop**

BGP connections can be established only with directly-connected EBGP peers by default.

(6) Enter the BGP IPv4 address family configuration mode.

address-family ipv4

(7) Disable IPv4 route exchange for the session.

no neighbor *rr-address* **activate**

(8) Exit the BGP IPv4 address family configuration mode.

exit

(9) Enter the BGP VPNv4 address family configuration mode.

address-family vpnv4

No VPNv4 address family is defined by default.

(10) Enable VPN route exchange with the RR in another AS.

neighbor *rr-address* **activate**

(11) (Optional) Configure the device not to change the next hop when advertising VPN routes to the peer.

neighbor *rr-address* **next-hop-unchanged**

The function is disabled by default. The device does not change the next hop when advertising routes to the IBGP peer.

1.10 Configuring OSPF VPN Extended Features

1.10.1 Overview

When OSPF is run in a VPN, run OSPF between PEs and CEs to simplify CE configuration and management.

1.10.2 Restrictions and Guidelines

The LDP router ID must be 32 bits.

1.10.3 Configuration Tasks

Configuration of OSPF VPN extended features includes the following tasks:

(1) Configuring the Domain ID

(2) Configuring the VPN Route Tag

(3) Configuring a Sham Link

(4) Configuring Loop Detection for a VRF-associated OSPF Process

(5) Configuring Extended Community Attributes of VPN Routes

(6) Disabling Loop Detection Based on the DN Bit Carried in LSAs

(7) Disabling Loop Detection Based on the Route Tag Carried in LSAs

1.10.4 Configuring the Domain ID

1. Overview

A domain ID is used to identify the domain to which an OSPF process belongs. Generally, all OSPF processes associated with VRF instances in a VPN are configured with the same domain ID.

2. Restrictions and Guidelines

- A VRF-associated OSPF process can be configured with multiple domain IDs. However, there is only one primary domain ID, and the others are secondary domain IDs. The only primary domain ID is configured by running the **domain-id** *value* command, while secondary domain IDs are configured by running the **domain-**

id *hex-value* **secondary** command. When OSPF routes are converted to VPN routes and advertised, the VPN routes contain only the primary domain ID.

- The primary and secondary domain IDs can be configured by running **domain-id** *ipv4-address* or **domain-id type** { **0005** | **0105** | **0205** | **8005** } **value**.
- Different VRF-associated OSPF processes can have the same domain ID. However, VRF-associated OSPF processes in the same VPN must be configured with the same domain ID to guarantee the correctness of route advertisement.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Create an OSPF process and enter the OSPF configuration mode.

router ospf *process-id* [**vrf** *vrf-name*]

The OSPF routing process is disabled by default.

(3) Configure the domain ID of an OSPF process.

domain-id { *ipv4-address* [**secondary**] | **null** | **type** { 0005 | 0105 | 0205 | 8005 } **value** *hex-value* [**secondary**] }

The default domain ID of an OSPF process is **null**, and the type value is **0x0005**.

This command takes effect only to OSPF processes associated with VRF instances.

1.10.5 Configuring the VPN Route Tag

1. Overview

In L3VPN applications, if a VPN site connects to multiple PEs, the VPN routes learned by a PE through MP-BGP are advertised to the VPN site in type 5/7 LSAs. Such routes may also be learned by other PEs connecting to this VPN site and then advertised, hence causing loops. To avoid such loops, the same VPN route tag must be configured for VRF-associated OSPF processes connecting to the same VPN site on PEs. When a VRF-associated OSPF process sends a type 5/7 LSA to the VPN site, this LSA carries the VPN route tag. When other PEs receive such a type 5/7 LSA containing the VPN route tag and the route tag is the same as the route tag of the corresponding OSPF process, the LSA is not used for OSPF route calculation.

2. Restrictions and Guidelines

Generally, OSPF processes belonging to the same VPN must be configured with the same VPN route tag.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Create an OSPF process and enter the OSPF configuration mode.

router ospf *process-id* [**vrf** *vrf-name*]

The OSPF routing process is disabled by default.

(3) Configure a VPN route tag for the OSPF process.

domain-tag *value*

The VPN route tag of an OSPF process is the AS number of the local BGP by default.

This command takes effect only to OSPF processes associated with VRF instances.

1.10.6 Configuring a Sham Link

1. Overview

A sham link is mainly used in a scenario where there is a backdoor link between VPN sites. If you expect to transmit VPN data still through the MPLS backbone network, you can establish a sham link between the VRF-associated OSPF processes of two PEs. Both OSPF processes can establish OSPF neighbors and distribute LSA packets through this sham link.

However, OSPF routes advertised through the sham link do not carry the VPN route tag and cannot be used for forwarding. Packets are still forwarded through BGP VPNv4 routes. In actual configuration, ensure that routes learned through the sham link are also learned through MP-BGP.

The source addresses used to establish the sham link must be redistributed to the BGP VPNv4 route but cannot participate in route calculation of the VRF-associated OSPF processes.

2. Restrictions and Guidelines

- The sham link configuration must be performed on two PEs. The link cannot be established if only one PE is configured.
- To establish a sham link between two PEs, the following conditions must be met:
  - The area IDs of the sham link configured on two PEs must be identical.
  - The <source address, destination address> of the sham link configured on one PE must be the same as the <destination address, source address> of the sham link configured on the other PE.
  - The source and destination addresses used to establish a sham link on the PEs must be 32-bit loopback addresses bound to VRF instances.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Create an OSPF process and enter the OSPF configuration mode.

router ospf *process-id* [**vrf** *vrf-name*]

The OSPF routing process is disabled by default.

(3) Configure the area ID, source address, and destination address of a sham link.

area *area-id* **sham-link** *source-address destination-address* [**cost** *number*] [**dead-interval** *dead-interval*] [**hello-interval** *hello-interval*] [**retransmit-interval** *retransmit-interval*] [**transmit-delay** ***transmit-delay***] [**authentication** [**message-digest** | **null** | **key-chain** *kechain-name*]] [**authentication-key** [**0** | **7**] *key* | **message-digest-key** *key-id* [**md5** | **hmac-md5** | **hmac-sha256**] [**0** | **7**] *key*]

The sham link is disabled by default. A sham link is not authenticated by default.

This command takes effect only to OSPF processes associated with VRF instances.

1.10.7 Configuring Loop Detection for a VRF-associated OSPF Process

1. Overview

In some application scenarios, the loop detection function of a VRF-associated OSPF process needs to be disabled. For example, when a VPN uses a multi-VPN-instance CE (MCE) to exchange VPN routes with a PE via OSPF, you must run the **capability vrf-lite** command on the MCE to disable the loop detection function of the VRF-associated OSPF process so that the VPN site can learn the routes of other VPN sites.

2. Restrictions and Guidelines

Before configuring loop detection, you must create a VRF instance.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Create an OSPF process and enter the OSPF configuration mode.

router ospf *process-id* [**vrf** *vrf-name*]

The OSPF routing process is disabled by default.

(3) Enable the loop detection function for an OSPF process associated with a VRF instance.

capability vrf-lite [**auto**]

A VRF-associated OSPF process supports PE-CE OSPF extended features by default, which include LSA conversion based on the domain ID, DN bit, and VPN route tag. If you do not expect a VRF-associated OSPF process to support PE-CE OSPF extended features, run the **capability vrf-lite** command.

This command takes effect only to OSPF processes associated with VRF instances.

1.10.8 Configuring Extended Community Attributes of VPN Routes

1. Overview

When OSPF routes are redistributed to the BGP domain to form VPN routes, the extended community attributes of OSPF routes are carried, including Router-ID and Route-Type. You can manually configure the type of Router-ID and Route-type to be compatible with implementation of different vendors.

For example, some vendors support only the Router-ID type 0x0107. When interconnecting with such vendors, run the **extcommunity-type** command to set the Router-ID type to 0x0107.

Some vendors support only the Router-Type type 0x8000. When interconnecting with such vendors, run the **extcommunity-type** command to set the Router-Type type to 0x8000.

2. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Create an OSPF process and enter the OSPF configuration mode.

router ospf *process-id* [**vrf** *vrf-name*]

The OSPF routing process is disabled by default.

(3) Configure OSPF route extended community attributes Router-ID and Route-Type.

extcommunity-type { router-id { 0107 | 8001 } | route-type { 0306 | 8000 } }

The default Router-ID is **0x0107**, and the default Route-Type is **0x0306**.

This command takes effect only to OSPF processes associated with VRF instances.

1.10.9 Disabling Loop Detection Based on the DN Bit Carried in LSAs

1. Overview

In L3VPN CE dual-homing scenarios, route calculation based on the DN bit is suppressed between PEs to prevent loops. In some scenarios, PEs may be allowed to learn routes from each other without generating loops. In these scenarios, you can cancel DN bit check. When a PE connects to an MCE, the MCE needs to calculate routes advertised by the PE and does not check the DN bit. OSPF type 3/5/7 LSAs can carry the DN bit.

2. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Create an OSPF process and enter the OSPF configuration mode.

router ospf *process-id* [**vrf** *vrf-name*]

The OSPF routing process is disabled by default.

(3) Disable loop detection using the DN bit in LSAs.

disable-dn-bit-check [summary | ase | nssa]

Loop detection based on the DN bit carried in LSAs is enabled by default.

This command takes effect only to OSPF processes associated with VRF instances.

1.10.10 Disabling Loop Detection Based on the Route Tag Carried in LSAs

1. Overview

In L3VPN CE dual-homing scenarios, when a PE receives an LSA with a route tag same as that of its own, the PE does not use the route tag to calculate routes. In this way, loops can be prevented. In some scenarios, PEs may be allowed to learn routes from each other without generating loops. In these scenarios, you can set different route tags for different PEs or disable route tag check. When a PE connects to an MCE, the MCE needs to calculate routes advertised by the PE and does not check the route tag. OSPF type 5/7 LSAs can carry the route tag.

2. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Create an OSPF process and enter the OSPF configuration mode.

router ospf *process-id* [**vrf** *vrf-name*]

The OSPF routing process is disabled by default.

(3) Disable loop detection based on the route tag carried in LSAs.

disable-tag-check

Loop detection based on the route tag carried in LSAs is enabled by default.

This command takes effect only to OSPF processes associated with VRF instances.

1.11 IPv6 MPLS L3VPN Configuration Task Summary

IPv6 MPLS L3VPN configuration includes the following tasks:

(1) Configuring the 6VPE Service Model

(2) Configuring the 6PE Service Model

1.12 Configuring the 6VPE Service Model

1.12.1 Overview

The 6VPE technology uses IPv4 BGP/MPLS VPN on an IPv4 backbone network to provide VPN services for IPv6 networks.

1.12.2 Configuration Tasks

6VPE service model configuration includes the following tasks:

(1) Configuring a Public Network Tunnel

(2) Configuring the VRF Instance of a 6VPE Device

(3) Configuring the IPv6 Address of a 6VPE Device Under a VRF Instance

(4) Configuring a BGP Session Between 6VPE Devices

(5) Configuring a 6VPE Device to Distribute IPv6 Routes Under a VRF Instance

(6) Configuring Routes Between a CE and a 6VPE Device

1.12.3 Configuring a Public Network Tunnel

1. Overview

On a public network, an LSP must be set up to carry users' service traffic. To run MPLS on the backbone network, you must run LDP on the Ps and PEs to establish public tunnels. This means that you have to configure LDP on MPLS devices and enable MPLS forwarding on each interface.

2. Restrictions and Guidelines

Currently, LDP supports only IPv4 public networks and does not support IPv6 public networks.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Enable MPLS forwarding globally.

mpls enable

MPLS forwarding is disabled globally by default.

To implement MPLS forwarding on a device, enable MPLS forwarding globally first.

(3) Enable LDP and enter the LDP configuration mode.

mpls router ldp

LDP is disabled by default.

(4) Configure the LDP router ID.

ldp router-id { *ipv4-address* | **interface** *interface-type interface-number* [**force**] }

The system router ID is used as the LDP router ID by default.

Generally, the loopback interface address is used as the router ID. If **force** is specified, the new router ID is forced to take effect immediately. Otherwise, the new router ID does not take effect immediately.

(5) Exit the LDP configuration mode.

exit

(6) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

interface *ethernet-type interface-number*

- Enter the Layer 3 link aggregation configuration mode.

interface aggregateport *interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

interface *ethernet-type interface-number.subnumber*

- Enter the Layer 3 aggregated sub-interface configuration mode.

interface aggregateport *interface-number.subnumber*

- Enter the SVI configuration mode.

interface vlan *interface-number*

(7) Set an IP address.

ip address ipv4-address mask-length

No IP address is configured for an interface by default.

(8) Enable labeled MPLS packet forwarding on the public interface.

label-switching

Forwarding labeled MPLS packets is disabled on an interface by default.

(9) Enable LDP on an interface.

mpls ldp enable

LDP is disabled on an interface by default.

1.12.4 Configuring the VRF Instance of a 6VPE Device

(1) Enter the global configuration mode.

configure terminal

(2) Create a multiprotocol VRF instance and enter the multiprotocol VRF instance configuration mode.

vrf definition *vrf-name*

vrf-name cannot exceed 31 characters.

(3) Configure the RD value.

rd *rd-value*

(4) Enable the VRF instance to support the IPv6 address family and enter the IPv6 address family configuration mode of a multiprotocol or global VRF .

address-family ipv6

The IPv6 address family is disabled by default.

(5) Configure the RT value.

route-target { **both** | **export** | **import** } *rt-value*

(6) Exit the IPv4 address family configuration mode of a multiprotocol or global VRF .

exit-address-family

1.12.5 Configuring the IPv6 Address of a 6VPE Device Under a VRF Instance

1. Restrictions and Guidelines

- A PE can forward IPv6 packets only after the **ipv6 enable** command is configured to enable IPv6 forwarding for a private interface on the PE.
- To save IPv6 addresses, you can use an automatically generated link-local address instead of a global IPv6 address for the interface on a PE used to connect to a CE.

2. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

  interface *ethernet-type interface-number*

- Enter the Layer 3 link aggregation configuration mode.

  interface aggregateport *interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

  interface *ethernet-type interface-number.subnumber*

- Enter the Layer 3 aggregated sub-interface configuration mode.

  interface aggregateport *interface-number.subnumber*

- Enter the SVI configuration mode.

  interface vlan *interface-number*

(3) Associate an interface with a VRF instance.

vrf forwarding *vrf_name*

(4) Enable IPv6 forwarding on the interface.

ipv6 enable

(5) (Optional) Configure the IPv6 address of the interface.

ipv6 address *ipv6-address/prefix-length*

1.12.6 Configuring a BGP Session Between 6VPE Devices

(1) Enter the global configuration mode.

configure terminal

(2) Create a BGP domain and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(3) Configure a BGP session.

neighbor *ipv4-address* **remote-as** *asn-number*

No BGP peer is configured by default.

(4) Use an interface address as the source address to establish a BGP session.

neighbor *ipv4-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

Generally, the loopback interface address is used as the source address.

(5) Enter the BGP VPNv6 address family configuration mode.

address-family vpnv6 unicast

(6) Activate IPv6 route exchange in the BGP session.

neighbor *ipv4-address* **activate**

(7) Exit the BGP VPNv6 address family configuration mode.

exit-address-family

1.12.7 Configuring a 6VPE Device to Distribute IPv6 Routes Under a VRF Instance

(1) Enter the global configuration mode.

configure terminal

(2) Create a BGP domain and enter the BGP configuration mode.

router bgp *asn-number*

BGP is disabled by default.

(3) Enter the BGP IPv6 address family configuration mode under a VRF instance.

address-family ipv6 vrf *vrf-name*

The IPv6 unicast address family configuration mode is used by default.

(4) (Optional) Establish a BGP session with a CE.

neighbor *ipv6-address* **remote-as** *ce-as-num*

No BGP peer is configured by default.

(5) (Optional) Configure IPv6 prefixes in the local AS that need to be imported to the BGP routing table.

network *ipv6-prefix*

(6) (Optional) Import network information in direct or static routes to BGP.

redistribute { **connected** | **static** } [**route-map** *map-tag*]

(7) Exit the BGP IPv6 address family configuration mode under the VRF instance.

exit-address-family

(8) Enter the BGP scope configuration mode.

scope vrf *vrf-name*

(9) Enter the BGP scope IPv6 address family configuration mode.

address-family ipv6 unicast

The IPv6 unicast address family configuration mode is used by default.

(10) (Optional) Establish a BGP session with a CE.

neighbor *ipv6-address* **remote-as** *ce-as-number*

No BGP peer is configured by default.

(11) (Optional) Configure IPv6 prefixes in the local AS that need to be imported to the BGP routing table.

network *ipv6-prefix*

(12) (Optional) Import network information in direct routes to BGP.

redistribute { **connected** | **static** } [**route-map** *map-tag*]

1.12.8 Configuring Routes Between a CE and a 6VPE Device

Routes between a CE and a 6VPE device can be IPv6 static or dynamic routes. When IPv6 dynamic routes are used, BGP routes of the 6VPE device and dynamic routes between the 6VPE and CE need to be imported to each other.

For details about IPv6 dynamic route configuration, see *Configuring RIPng*, *Configuring OSPFv3*, *Configuring IS-IS*, and *Configuring BGP*.

1.13 Configuring the 6PE Service Model

1.13.1 Overview

Network SPs use existing IPv4 MPLS backbone networks to provide access services (non-VPN services) for scattered IPv6 networks in case of IPv6 islands.

The 6VPE technology uses IPv4 BGP/MPLS VPN on an IPv4 backbone network to provide VPN services for IPv6 networks.

1.13.2 Configuration Tasks

6PE service model configuration includes the following tasks:

(1) Configuring a Public Network Tunnel

(2) Configuring the IPv6 Address of a 6PE Device

(3) Configuring a BGP Session Between 6PE Devices

(4) Configuring Routes Between a CE and a 6VPE Device

1.13.3 Configuring a Public Network Tunnel

1. Overview

On a public network, an LSP must be set up to carry users' service traffic. To run MPLS on the backbone network, you must run LDP on the Ps and PEs to establish public tunnels. This means that you have to configure LDP on MPLS devices and enable MPLS forwarding on each interface.

2. Restrictions and Guidelines

Currently, LDP supports only IPv4 public networks and does not support IPv6 public networks.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Enable MPLS forwarding globally.

mpls enable

MPLS forwarding is disabled globally by default.

To implement MPLS forwarding on a device, enable MPLS forwarding globally first.

(3) Enable LDP and enter the LDP configuration mode.

mpls router ldp

LDP is disabled by default.

(4) Configure the LDP router ID.

ldp router-id { *ipv4-address* | **interface** *interface-type interface-number* [**force**] }

The system router ID is used as the LDP router ID by default.

Generally, the loopback interface address is used as the router ID. If **force** is specified, the new router ID is forced to take effect immediately. Otherwise, the new router ID does not take effect immediately.

(5) Return to the global configuration mode.

exit

(6) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

  interface *ethernet-type interface-number*

- Enter the Layer 3 link aggregation configuration mode.

  interface aggregateport *interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

  interface *ethernet-type interface-number.subnumber*

- Enter the Layer 3 aggregated sub-interface configuration mode.

  interface aggregateport *interface-number.subnumber*

- Enter the SVI configuration mode.

  interface vlan *interface-number*

(7) Configure an IP address for the interface.

ip address *ipv4-address mask-length*

No IP address is configured for an interface by default.

(8) Enable labeled MPLS packet forwarding on the public interface.

label-switching

Forwarding labeled MPLS packets is disabled on an interface by default.

(9) Enable LDP on the interface.

mpls ldp enable

LDP is disabled on an interface by default.

(1) Enable the IPv6 capability on the interface.

ipv6 enable

The IPv6 capability is disabled on the interface by default.

1.13.4 Configuring the IPv6 Address of a 6PE Device

1. Overview

Configure an IPv6 address for the interface on the 6PE device used to connect to a CE.

2. Restrictions and Guidelines

To save IPv6 addresses, you can use an automatically generated link-local address instead of a global IPv6 address for the interface on a PE used to connect to a CE.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

  interface *ethernet-type interface-number*

- Enter the Layer 3 link aggregation configuration mode.

  interface aggregateport *interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

  interface *ethernet-type interface-number.subnumber*

- Enter the Layer 3 aggregated sub-interface configuration mode.

  interface aggregateport *interface-number.subnumber*

- Enter the SVI configuration mode.

  interface vlan *interface-number*

(3) Enable IPv6 forwarding on the interface.

ipv6 enable

(4) Configure an IPv6 address for the interface.

ipv6 address ipv6-address/prefix-length

No IPv6 address is configured for the interface by default.

1.13.5 Configuring a BGP Session Between 6PE Devices

(1) Enter the global configuration mode.

configure terminal

(2) Create a BGP domain and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(3) Configure a BGP session.

neighbor *ipv4-address* **remote-as** *as-number*

No BGP peer is configured by default.

Currently, a 6PE device only supports BGP sessions using IPv4 addresses.

(4) Use an interface address as the source address to establish an MP-IBGP session.

neighbor *ipv4-address* **update-source** *interface-type interface-number*

The optimal local interface is used as the outbound interface by default.

Generally, the loopback interface address is used as the source address.

(5) Enter the BGP IPv6 address family configuration mode.

address-family ipv6 **unicast**

The IPv6 unicast address family configuration mode is used by default.

(6) Activate IPv6 route exchange in the BGP session.

neighbor *ipv4-address* **activate**

(7) Activate labeled IPv6 route exchange in the BGP session.

neighbor *ipv4-address* **send-label**

Forwarding labeled MPLS packets is disabled by default.

1.13.6 Configuring Routes Between a CE and a 6VPE Device

Routes between a CE and a 6PE device can be IPv6 static or dynamic routes. When IPv6 dynamic routes are used, BGP4+ routes of the 6PE device and dynamic routes between the 6PE and CE need to be imported to each other.

For details about IPv6 dynamic route configuration, see *Configuring RIPng*, *Configuring OSPFv3*, *Configuring IS-IS*, and *Configuring BGP*.

1.14 Configuring a CSC Service Model

1.14.1 Overview

In the CSC service model, first carriers provide the VPN service to second carriers, and second carriers provide the IP access service and VPN service to users.

1.14.2 Restrictions and Guidelines

- The router ID for LDP and BGP must contain 32 bits.

1.14.3 Configuration Tasks

The CSC service model configuration includes the following tasks:

(1) Configuring Basic BGP/MPLS VPN Features (First Carrier)

(2) Configuring PEs and CEs to Distribute Labels Using LDP (First Carrier)

(3) Configuring PEs and CEs to Distribute Labels Using EBGP (First Carrier)

(4) Configuring the IP Core to Provide the Internet Service (Second Carrier)

(5) Configuring the MPLS Core to Provide the Internet Service (Second Carrier)

(6) Configuring the MPLS Core to Provide the VPN Service (Second Carrier)

(7) Configuring the Second Carrier to Provide User Access

1.14.4 Configuring Basic BGP/MPLS VPN Features (First Carrier)

1. Restrictions and Guidelines

- To configure the CSC model, the per-route label distribution mode must be used for each VRF instance. Therefore, you need to run the **alloc-label per-route** command in the VRF instance configuration mode to select the per-route label distribution mode.
- When the second carrier is an Internet provider with IP core, if PEs and CEs exchange internal routes using EBGP and exchange external routes using BGP and the CEs are RRs, a route map must be configured on the PEs and CEs to filter external routes, preventing external routers from being leaked to the PEs of the first carrier.

1. Procedure

(1) Configure an MPLS network. For details, see 1.3.4 Configuring an MPLS Network.

(2) Configure a VRF instance. For details, see 1.3.5 Configuring a VPN Routing Instance.

(3) Configure MP-IBGP neighbors. For details, see 1.3.6 Configuring VPN Route Exchange Between PEs.

(4) Configure route exchange between PEs and CEs. Choose one of the following based on the routing protocol:

- Run BGP between PEs and CEs. For details, see 1.3.7 Running BGP Between PEs and CEs.
- Run OSPF between PEs and CEs. For details, see 1.3.8 Running OSPF Between PEs and CEs.
- Run RIP between PEs and CEs. For details, see 1.3.9 Running RIP Between PEs and CEs.
- Configure static routes between PEs and CEs. For details, see 1.3.10 Configuring Static Routes Between PEs and CEs.

1.14.5 Configuring PEs and CEs to Distribute Labels Using LDP (First Carrier)

1. Overview

When the CSC feature is configured, if the PEs and CEs of the first carrier exchange routes using IGP, LDP must be configured to distribute labels.

2. Restrictions and Guidelines

- The configuration on the PE is similar to that on the CE. The difference is that the PE uses BGP to distribute labels to routes.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) On the PE and CE, enable LDP in a VRF instance and enter the LDP configuration mode.

mpls router ldp *vrf-name*

LDP is disabled by default.

(3) On the PE and CE, configure the LDP router ID.

ldp router-id { *ipv4-address* | **interface** *interface-type interface-number* [**force**] }

The system router ID is used as the LDP router ID by default.

(4) On the PE, distribute labels to BGP routes.

advertise-labels [**for acl** [**ipv6**] *acl-name* [**to** *peer-acl-name*] | **for bgp-routes** [**acl** [**ipv6**] *bgp-routes-acl-name*] | **for default-route** | **for host-routes**]

Labels are distributed to BGP routes by default.

(5) On the PE and CE, exit the LDP configuration mode.

exit

(6) On the PE and CE, enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.
  interface *ethernet-type interface-number*
- Enter the Layer 3 aggregate interface configuration mode.
  interface aggregateport *interface-number*
- Enter the Layer 3 Ethernet sub-interface configuration mode.
  interface *ethernet-type interface-number.subnumber*
- Enter the Layer 3 aggregate sub-interface configuration mode.
  interface aggregateport *interface-number.subnumber*
- Enter the SVI interface configuration mode.
  interface vlan *interface-number*

(7) On the PE and CE, enable labeled MPLS packet forwarding on the interface.

label-switching

Labeled MPLS packet forwarding is disabled on an interface by default.

(8) On the PE and CE, enable LDP on the interface.

mpls ldp enable

LDP is disabled on an interface by default.

1.14.6 Configuring PEs and CEs to Distribute Labels Using EBGP (First Carrier)

1. Overview

When the CSC feature is configured, if the PEs and CEs of the first carrier exchange routes using EBGP, EBGP must be configured to distribute labels.

2. Restrictions and Guidelines

- The configuration on the PE is similar to that on the CE.

3. Procedure

(1) Enter the global configuration mode.

configure terminal

(2) On the PE and CE, enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.
  interface *ethernet-type interface-number*
- Enter the Layer 3 aggregate interface configuration mode.
  interface aggregateport *interface-number*
- Enter the Layer 3 Ethernet sub-interface configuration mode.
  interface *ethernet-type interface-number.subnumber*
- Enter the Layer 3 aggregate sub-interface configuration mode.
  interface aggregateport *interface-number.subnumber*
- Enter the SVI interface configuration mode.
  interface vlan *interface-number*

(3) On the PE and CE, enable labeled MPLS packet forwarding on the interface.

label-switching

Labeled MPLS packet forwarding is disabled on an interface by default.

(4) On the PE and CE, exit the interface configuration mode.

exit

(5) Enable BGP and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(6) Enter the BGP IPv4 address family configuration mode.

address-family ipv4

(7) Configure the device to send MPLS labeled routes to the specified neighbor.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **send-label**

1.14.7 Configuring the IP Core to Provide the Internet Service (Second Carrier)

1. Overview

As shown in Figure 1-12, the second carrier uses an IP core network to provide network access services to users. ASBR1, ASBR2, CE1, and CE2 establish IBGP neighbor relationships to exchange external routes.

2. Restrictions and Guidelines

- ASBRs and CEs establish IBGP neighbor relationships to exchange external routes. CEs function as RRs to reflect external routes between sites.

3. Prerequisites

Before performing operations in this section, configure IGP on the second carrier network to ensure interconnection of the second carrier network.

4. Configuring an Internal IBGP Session in a Site

Establish an IBGP session between the ASBR and CSC-CE in a site and configure the CSC-CE as the RR.

(1) Enter the global configuration mode.

configure terminal

(2) Enable BGP and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(3) Configure a BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **remote-as** { *as-number* | **route-map-name** *map-tag* }

No BGP peer is configured by default.

(4) Configure the CE in the CSC as the RR client.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **route-reflector-client**

The RR feature is disabled by default.

(5) Configure a source address for the BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **update-source** { *interface-type interface-number* | *address* }

The optimal local interface is used as the outbound interface by default.

(6) Configure the ASBR to change the next hop to its own address when advertising routes to the BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **next-hop-self**

By default, the ASBR changes the next hop to the local BGP speaker when advertising routes to the EBGP peer and does not change the next hop when advertising routes to the IBGP peer.

5. Configuring IBGP Sessions Between CSC-CEs of Different Sites

Establish fully-connected IBGP sessions between CSC-CEs of different sites to transmit external routes of the sites.

(1) Enter the global configuration mode.

configure terminal

(2) Enable BGP and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(3) Configure a BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **remote-as** { *as-number* | **route-map-name** *map-tag* }

No BGP peer is configured by default.

(4) Configure the source address for the BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **update-source** { *interface-type interface-number* | *address* }

(5) (Optional) Configure the CSC-CE of another site as the RR client.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **route-reflector-client**

The RR feature is disabled by default.

(6) Exit the BGP configuration mode.

exit

(7) Enable the device to parse the next hop in a BGP route to an LSP tunnel.

recursive-route lookup lsp

The capability of parsing the next hop in a BGP route to an LSP tunnel is disabled by default.

6. Configuring a Route Map for Route Filtering

When internal routes are exchanged using BGP, as the CSC-CEs are responsible for transmitting both external routes and internal routes, you must ensure that the EBGP sessions between the CSC-CEs and CSC-PEs transmit only internal routes and IBGP sessions between the CSC-CEs as well as between the CSC-CEs and ASBRs transmit only external routes. Otherwise, route loop or disorder may occur. To prevent this problem, you need to run the **neighbor route-map** { **in** | **out** } command on the IBGP neighbors and EBGP neighbors to filter corresponding routes. For ease of use, AS path filtering rules are used. You can also use other rules.

(1) Enter the global configuration mode.

configure terminal

(2) Configure AS path filtering rules.

ip as-path access-list *path-list-number* { **permit** | **deny** } *regular-expression*

No AS path filtering rule exists by default.

(3) Create a route map and enter the route map configuration mode.

route-map *route-map-name* [**permit** | **deny**] [*sequence-number*]

No route map is configured by default.

(9) Match routes based on the AS_PATH attribute.

match as-path *as-path-acl-list-number*&<1-10>

No AS_PATH attribute list is matched by default.

(4) Enable BGP and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(5) Apply the route map to a BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **route-map** *map-tag* { **in** | **out** }

Route mapping is not performed on received or sent routes by default.

1.14.8 Configuring the MPLS Core to Provide the Internet Service (Second Carrier)

1. Overview

As shown in Figure 1-13, the second carrier uses an MPLS core network. IBGP neighbor relationships are established directly between the ASBRs to exchange external routes, so that the CSC-CEs do not need to transmit external routes.

2. Configuring an MPLS Network in Each Site

Configure an MPLS network. For details, see 1.3.4 Configuring an MPLS Network.

Note

LDP must be enabled on the CSC-CE to establish sessions with other devices in the same site so as to establish an MPLS network. If the CSC-CE and CSC-PE learn routes using BGP, you must run the **advertise-labels for bgp-routes** command on the CSC-CE to enable label distribution to BGP routes through LDP.

3. Establishing IBGP Sessions Between ASBRs of Different Sites

Establish BGP sessions between the ASBRs inside a site and between the ASBRs of different sites to transmit external routes.

Note

To reduce the configuration cost of fully-connected IBGP sessions, an RR role can be set inside a site. A BGP session can be established between the ASBR and RR in the site and a BGP session can be established between the RRs of different sites.

(2) Enter the global configuration mode.

configure terminal

(1) Enable BGP and enter the BGP configuration mode.

router bgp *as-number*

BGP is disabled by default.

(2) Configure a BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **remote-as** { *as-number* | **route-map-name** *map-tag* }

No BGP peer is configured by default.

(3) Configure a source address for the BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **update-source** { *interface-type interface-number* | *address* }

The optimal local interface is used as the outbound interface by default.

(4) Configure the ASBR to change the next hop to its own address when advertising external routes to the BGP peer.

neighbor { *neighbor-ipv4-address* | *peer-group-name* } **next-hop-self**

By default, the ASBR changes the next hop to the local BGP speaker when advertising routes to the EBGP peer and does not change the next hop when advertising routes to the IBGP peer.

(5) Enable the device to parse the next hop in a BGP route to an LSP tunnel.

recursive-route lookup lsp

The capability of parsing the next hop in a BGP route to an LSP tunnel is disabled by default.

1.14.9 Configuring the MPLS Core to Provide the VPN Service (Second Carrier)

1. Overview

As shown in Figure 1-14, the second carrier uses an MPLS core network to provide MPLS L3VPN services to users. MP-IBGP neighbor relationships are established between the PEs of the second carrier to exchange VPN routes of users.

2. Configuring an MPLS Network in Each Site

Configure an MPLS network. For details, see 1.3.4 Configuring an MPLS Network.

Note

LDP must be enabled on the CSC-CE to establish sessions with other devices in the same site so as to establish an MPLS network. If the CSC-CE and CSC-PE learn routes using BGP, you must run the **advertise-labels for bgp-routes** command on the CSC-CE to enable label distribution to BGP routes through LDP.

3. Establishing MP-IBGP Neighbors Between PEs of Different Sites

Configure MP-IBGP neighbors. For details, see 1.3.6 Configuring VPN Route Exchange Between PEs.

Note

To reduce the configuration cost of fully-connected MP-IBGP sessions, an RR role can be set inside a site. An MP-IBGP session can be established between the PE and RR in the site and an MP-IBGP session can be established between the RRs of different sites.

1.14.10 Configuring the Second Carrier to Provide User Access

The configuration in this section is related to services provided by the second carrier but not the CSC model. If the second carrier provides IP services to users, see RIP, OSPFv2, IS-IS, and BGP in "IP Routing Configuration". If the second carrier provides MPLS VPN services to users, see "MPLS Configuration".

1.15 Monitoring

Run the **clear** commands to clear information.

Caution

Running the **clear** commands may lose vital information and thus interrupt services.

Run the **show** commands to check the running status of a configured function to verify the configuration effect.

Run the **debug** command to output debugging information.

Caution

The output debugging information occupies system resources. Therefore, disable the debugging function immediately after use.

Table 1-1 MPLS L3VPN Monitoring

Command	Purpose
clear bgp vpnv4 unicast { * | *as-number* | *peer-address* } [**soft**] [**in** | **out**]	Resets the VPNv4 unicast address family of BGP.
clear bgp vpnv4 unicast dampening [*ipv4-address* [*mask-length*]]	Clears VPNv4 route flapping information and restores suppressed VPNv4 routes.
clear bgp vpnv4 unicast external [**soft**] [**in** | **out**]	Resets all EBGP connections of the VPNv4 address family.
clear bgp vpnv4 unicast flap-statistics [*ipv4-address* [*mask-length*]]	Clears route flapping statistics of the VPNv4 address family.
clear bgp vpnv4 unicast peer-group *peer-group-name* [**soft**] [**in** | **out**]	Resets VPNv4 address families of all members in a BGP peer group.
clear bgp vpnv6 unicast { * | *as-number* | *peer-address* } [**soft**] [**in** | **out**]	Resets the VPNv6 unicast address family of BGP.
clear bgp vpnv6 unicast dampening	Clears VPNv6 route flapping information and restores suppressed VPNv6 routes.
clear bgp vpnv6 unicast external [**soft**] [**in** | **out**]	Resets all EBGP connections of the VPNv6 address family.
clear bgp vpnv6 unicast flap-statistics	Clears route flapping statistics of the VPNv6 address family.
clear bgp vpnv6 unicast peer-group *peer-group-name* [**soft**] [**in** | **out**]	Resets VPNv6 address families of all members in a BGP peer group.
clear ip bgp vrf *vrf-name* { * | *address* | *as-num* } [**soft**] [**in** | **out**]	Resets the BGP session of a VRF instance.
show bgp ipv4 unicast labels	Displays IPv4 routing information with the MPLS label learned and sent by BGP.
show bgp ipv6 unicast labels	Displays IPv6 routing information with the MPLS label learned and sent by BGP.
show bgp vpnv4 unicast all [*network* | **neighbor** [*peer-address*] | **summary** | **label**]	Displays all VPNv4 routing information or neighbor information learned and sent by BGP.
show bgp vpnv4 unicast vrf *vrf-name* [*network* | **summary** | **label**]	Displays VPNv4 routing information or neighbor information learned and sent by BGP under a VRF instance.
show bgp vpnv4 unicast rd *rd-value* [*network* | **summary** | **label**]	Displays VPNv4 routing information or neighbor information learned and sent by BGP under an RD.

Command	Purpose
show bgp vpnv6 unicast all [*network* | **neighbor** [*peer-address*] | **summary** | **label**]	Displays VPNv6 routing information or neighbor information learned and sent by BGP.
show bgp vpnv6 unicast vrf *vrf-name* [*network* | **summary** | **label**]	Displays VPNv6 routing information or neighbor information learned and sent by BGP under a VRF instance.
show bgp vpnv6 unicast rd *rd-value* [*network* | **summary** | **label**]	Displays VPNv6 routing information or neighbor information learned and sent by BGP under an RD.
show ip extcommunity-list [*extcommunity-list-num* | *extcommunity-list-name*]	Displays configurations of an extended community attribute list.
show ip ospf [*process-id*] **sham-links** [**area** *area-id*]	Displays OSPF sham link information.
show ip vrf [**brief** | **detail** | **interfaces**] [*vrf-name*]	Displays configured single-protocol VRF instance information.
show vrf brief [*vrf-name*]	Displays the brief VRF instance information (including single-protocol and multiprotocol VRF instances).
show vrf ipv4 [*vrf-name*]	Displays brief IPv4 address family information of VRF instances (including single-protocol and multiprotocol VRF instances)
show vrf ipv6 [*vrf-name*]	Displays brief IPv6 address family information of a multiprotocol VRF instance.
show vrf detail [*vrf-name*]	Displays detailed VRF instance information (including single-protocol VRF instances and multi-protocol VRF instances).
show mpls forwarding-table	Displays L3VPN forwarding entries.
debug ip bgp mpls	Debugs BGP MPLS.
debug mpls	Debugs MPLS entry internal processing.
debug mpls msg [**send** | **recv**]	Debugs MPLS messages.

1.16 IPv4 MPLS L3VPN Configuration Examples

1.16.1 Configuring Basic IPv4 MPLS L3VPN Functions (Intranet)

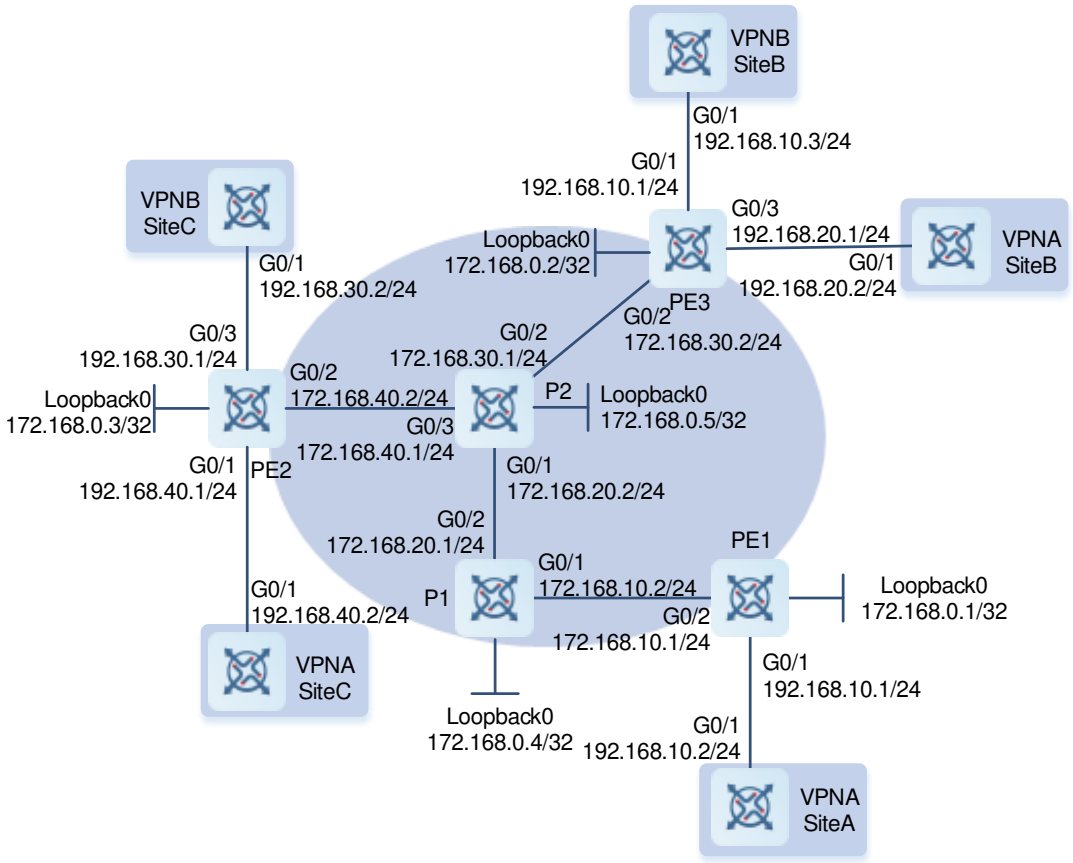
1. Requirements

There are two VPNs: VPNA and VPNB. VPNA has user sites at SiteA, SiteB, and SiteC, VPNB has user sites at SiteB and SiteC, and VPNA SiteA and VPNB SiteB have address overlapping. The requirements are as follows:

- Users at different sites of VPNA can communicate with each other.
- Users at different sites of VPNB can communicate with each other.
- Users in VPNA and VPNB cannot communicate with each other.

2. Topology

Figure 1-19 Configuring Basic IPv4 MPLS L3VPN Functions (Intranet)



3. Notes

- Configure interface IP addresses and OSPF on MPLS backbone network nodes to ensure communication between them.
- Enable MPLS forwarding and LDP and establish LDP LSPs on MPLS backbone network nodes.
- Configure VPN routing instances, define RD and RT values, and associate VRF instances with interfaces on PEs.

- Configure MP-IBGP neighbors on PEs to exchange VPN routing information.
- Establish EBGP sessions between different VPN sites and PEs to exchange VPN routing information.

4. Procedure

(1) Configure interface IP addresses and OSPF on MPLS backbone network nodes to ensure communication between them.

Configure PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# interface loopback 0
PE1(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
PE1(config-if-Loopback 0)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# no switchport
PE1(config-if-GigabitEthernet 0/2)# ip address 172.168.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# exit
PE1(config)# router ospf 10
PE1(config-router)# network 172.168.0.1 0.0.0.0 area 0
PE1(config-router)# network 172.168.10.0 0.0.0.255 area 0
PE1(config-router)# exit
```

Configure P1.

```
P1> enable
P1# configure terminal
P1(config)# interface loopback 0
P1(config-if-Loopback 0)# ip address 172.168.0.4 255.255.255.255
P1(config-if-Loopback 0)# exit
P1(config)# interface gigabitethernet 0/1
P1(config-if-GigabitEthernet 0/1)# no switchport
P1(config-if-GigabitEthernet 0/1)# ip address 172.168.10.2 255.255.255.0
P1(config-if-GigabitEthernet 0/1)# exit
P1(config)# interface gigabitethernet 0/2
P1(config-if-GigabitEthernet 0/2)# no switchport
P1(config-if-GigabitEthernet 0/2)# ip address 172.168.20.1 255.255.255.0
P1(config-if-GigabitEthernet 0/2)# exit
P1(config)# router ospf 10
P1(config-router)# network 172.168.0.4 0.0.0.0 area 0
P1(config-router)# network 172.168.10.0 0.0.0.255 area 0
P1(config-router)# network 172.168.20.0 0.0.0.255 area 0
P1(config-router)# exit
```

Configure P2.

```
P2> enable
P2# configure terminal
P2(config)# interface loopback 0
P2(config-if-Loopback 0)# ip address 172.168.0.5 255.255.255.255
```

```
P2(config-if-Loopback 0)# exit
P2(config)# interface gigabitethernet 0/1
P2(config-if-GigabitEthernet 0/1)# no switchport
P2(config-if-GigabitEthernet 0/1)# ip address 172.168.20.2 255.255.255.0
P2(config-if-GigabitEthernet 0/1)# exit
P2(config)# interface gigabitethernet 0/2
P2(config-if-GigabitEthernet 0/2)# no switchport
P2(config-if-GigabitEthernet 0/2)# ip address 172.168.30.1 255.255.255.0
P2(config-if-GigabitEthernet 0/2)# exit
P2(config)# interface gigabitethernet 0/3
P2(config-if-GigabitEthernet 0/3)# no switchport
P2(config-if-GigabitEthernet 0/3)# ip address 172.168.40.1 255.255.255.0
P2(config-if-GigabitEthernet 0/3)# exit
P2(config)# router ospf 10
P2(config-router)# network 172.168.0.5 0.0.0.0 area 0
P2(config-router)# network 172.168.20.0 0.0.0.255 area 0
P2(config-router)# network 172.168.30.0 0.0.0.255 area 0
P2(config-router)# network 172.168.40.0 0.0.0.255 area 0
P2(config-router)# exit
```

Configure PE2.

```
PE2> enable
PE2# configure terminal
PE2(config)# interface loopback 0
PE2(config-if-Loopback 0)# ip address 172.168.0.3 255.255.255.255
PE2(config-if-Loopback 0)# exit
PE2(config)# interface gigabitethernet 0/2
PE2(config-if-GigabitEthernet 0/2)# no switchport
PE2(config-if-GigabitEthernet 0/2)# ip address 172.168.40.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/2)# exit
PE2(config)# router ospf 10
PE2(config-router)# network 172.168.0.3 0.0.0.0 area 0
PE2(config-router)# network 172.168.40.0 0.0.0.255 area 0
PE2(config-router)# exit
```

Configure PE3.

```
PE3> enable
PE3# configure terminal
PE3(config)# interface loopback 0
PE3(config-if-Loopback 0)# ip address 172.168.0.2 255.255.255.255
PE3(config-if-Loopback 0)# exit
PE3(config)# interface gigabitethernet 0/2
PE3(config-if-GigabitEthernet 0/2)# no switchport
PE3(config-if-GigabitEthernet 0/2)# ip address 172.168.30.2 255.255.255.0
PE3(config-if-GigabitEthernet 0/2)# exit
PE3(config)# router ospf 10
PE3(config-router)# network 172.168.0.2 0.0.0.0 area 0
```

```
PE3(config-router)# network 172.168.30.0 0.0.0.255 area 0
PE3(config-router)# exit
```

(2) Enable MPLS forwarding and LDP and establish LDP LSPs on MPLS backbone network nodes.

Configure PE1.

```
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# label-switching
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/2)# exit
```

Configure P1.

```
P1(config)# mpls enable
P1(config)# mpls router ldp
P1(config-mpls-router)# ldp router-id interface loopback 0 force
P1(config-mpls-router)# exit
P1(config)# interface gigabitethernet 0/1
P1(config-if-GigabitEthernet 0/1)# label-switching
P1(config-if-GigabitEthernet 0/1)# mpls ldp enable
P1(config-if-GigabitEthernet 0/1)# exit
P1(config)# interface gigabitethernet 0/2
P1(config-if-GigabitEthernet 0/2)# label-switching
P1(config-if-GigabitEthernet 0/2)# mpls ldp enable
P1(config-if-GigabitEthernet 0/2)# exit
```

Configure P2.

```
P2(config)# mpls enable
P2(config)# mpls router ldp
P2(config-mpls-router)# ldp router-id interface loopback 0 force
P2(config-mpls-router)# exit
P2(config)# interface gigabitethernet 0/1
P2(config-if-GigabitEthernet 0/1)# label-switching
P2(config-if-GigabitEthernet 0/1)# mpls ldp enable
P2(config-if-GigabitEthernet 0/1)# exit
P2(config)# interface gigabitethernet 0/2
P2(config-if-GigabitEthernet 0/2)# label-switching
P2(config-if-GigabitEthernet 0/2)# mpls ldp enable
P2(config-if-GigabitEthernet 0/2)# exit
P2(config)# interface gigabitethernet 0/3
P2(config-if-GigabitEthernet 0/3)# label-switching
P2(config-if-GigabitEthernet 0/3)# mpls ldp enable
P2(config-if-GigabitEthernet 0/3)# exit
```

Configure PE2.

```
PE2(config)# mpls enable
```

```
PE2(config)# mpls router ldp
PE2(config-mpls-router)# ldp router-id interface loopback 0 force
PE2(config-mpls-router)# exit
PE2(config)# interface gigabitethernet 0/2
PE2(config-if-GigabitEthernet 0/2)# label-switching
PE2(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE2(config-if-GigabitEthernet 0/2)# exit
```

Configure PE3.

```
PE3(config)# mpls enable
PE3(config)# mpls router ldp
PE3(config-mpls-router)# ldp router-id interface loopback 0 force
PE3(config-mpls-router)# exit
PE3(config)# interface gigabitethernet 0/2
PE3(config-if-GigabitEthernet 0/2)# label-switching
PE3(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE3(config-if-GigabitEthernet 0/2)# exit
```

(3) Configure VPN routing instances, define RD and RT values, and associate VRF instances with interfaces on PEs.

Configure PE1.

```
PE1(config)# ip vrf VPNA
PE1(config-vrf)# rd 1:100
PE1(config-vrf)# route-target both 1:100
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# no switchport
PE1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPNA
PE1(config-if-GigabitEthernet 0/1)# ip address 192.168.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# exit
```

Configure PE2.

```
PE2(config)# ip vrf VPNA
PE2(config-vrf)# rd 1:100
PE2(config-vrf)# route-target both 1:100
PE2(config-vrf)# exit
PE2(config)# ip vrf VPNB
PE2(config-vrf)# rd 1:200
PE2(config-vrf)# route-target both 1:200
PE2(config-vrf)# exit
PE2(config)# interface gigabitethernet 0/1
PE2(config-if-GigabitEthernet 0/1)# no switchport
PE2(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPNA
PE2(config-if-GigabitEthernet 0/1)# ip address 192.168.40.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/1)# exit
PE2(config)# interface gigabitethernet 0/3
PE2(config-if-GigabitEthernet 0/3)# no switchport
```

```
PE2(config-if-GigabitEthernet 0/3)# ip vrf forwarding VPNB
PE2(config-if-GigabitEthernet 0/3)# ip address 192.168.30.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/3)# exit
```

Configure PE3.

```
PE3(config)# ip vrf VPNA
PE3(config-vrf)# rd 1:100
PE3(config-vrf)# route-target both 1:100
PE3(config-vrf)# exit
PE3(config)# ip vrf VPNB
PE3(config-vrf)# rd 1:200
PE3(config-vrf)# route-target both 1:200
PE3(config-vrf)# exit
PE3(config)# interface gigabitethernet 0/1
PE3(config-if-GigabitEthernet 0/1)# no switchport
PE3(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPNB
PE3(config-if-GigabitEthernet 0/1)# ip address 192.168.10.1 255.255.255.0
PE3(config-if-GigabitEthernet 0/1)# exit
PE3(config)# interface gigabitethernet 0/3
PE3(config-if-GigabitEthernet 0/3)# no switchport
PE3(config-if-GigabitEthernet 0/3)# ip vrf forwarding VPNA
PE3(config-if-GigabitEthernet 0/3)# ip address 192.168.20.1 255.255.255.0
PE3(config-if-GigabitEthernet 0/3)# exit
```

(4) Configure MP-IBGP neighbors on PEs to exchange VPN routing information.

Configure PE1.

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 172.168.0.2 remote-as 1
PE1(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE1(config-router)# neighbor 172.168.0.3 remote-as 1
PE1(config-router)# neighbor 172.168.0.3 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 172.168.0.2 activate
PE1(config-router-af)# neighbor 172.168.0.3 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

Configure PE2.

```
PE2(config)# router bgp 1
PE2(config-router)# neighbor 172.168.0.1 remote-as 1
PE2(config-router)# neighbor 172.168.0.1 update-source loopback 0
PE2(config-router)# neighbor 172.168.0.2 remote-as 1
PE2(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 172.168.0.1 activate
PE2(config-router-af)# neighbor 172.168.0.2 activate
PE2(config-router-af)# exit-address-family
```

```
PE2(config-router)# exit
```

Configure PE3.

```
PE3(config)# router bgp 1
PE3(config-router)# neighbor 172.168.0.1 remote-as 1
PE3(config-router)# neighbor 172.168.0.1 update-source loopback 0
PE3(config-router)# neighbor 172.168.0.3 remote-as 1
PE3(config-router)# neighbor 172.168.0.3 update-source loopback 0
PE3(config-router)# address-family vpnv4
PE3(config-router-af)# neighbor 172.168.0.1 activate
PE3(config-router-af)# neighbor 172.168.0.3 activate
PE3(config-router-af)# exit-address-family
PE3(config-router)# exit
```

(5) Establish EBGP sessions between different VPN sites and PEs to exchange VPN routing information.

Configure PE1.

```
PE1(config)# router bgp 1
PE1(config-router)# address-family ipv4 vrf VPNA
PE1(config-router-af)# neighbor 192.168.10.2 remote-as 65001
PE1(config-router-af)# neighbor 192.168.10.2 activate
PE1(config-router-af)# redistribute connected
PE1(config-router-af)# exit-address-family
PE1(config-router)# end
```

Configure PE2.

```
PE2(config)# router bgp 1
PE2(config-router)# address-family ipv4 vrf VPNA
PE2(config-router-af)# neighbor 192.168.40.2 remote-as 65003
PE2(config-router-af)# neighbor 192.168.40.2 activate
PE2(config-router-af)# redistribute connected
PE2(config-router-af)# exit-address-family
PE2(config-router)# address-family ipv4 vrf VPNB
PE2(config-router-af)# neighbor 192.168.30.2 remote-as 65005
PE2(config-router-af)# neighbor 192.168.30.2 activate
PE2(config-router-af)# redistribute connected
PE2(config-router-af)# exit-address-family
PE2(config-router)# end
```

Configure PE3.

```
PE3(config)# router bgp 1
PE3(config-router)# address-family ipv4 vrf VPNA
PE3(config-router-af)# neighbor 192.168.20.2 remote-as 65002
PE3(config-router-af)# neighbor 192.168.20.2 activate
PE3(config-router-af)# redistribute connected
PE3(config-router-af)# exit-address-family
PE3(config-router)# address-family ipv4 vrf VPNB
PE3(config-router-af)# neighbor 192.168.10.3 remote-as 65004
PE3(config-router-af)# neighbor 192.168.10.3 activate
```

```
PE3(config-router-af)# redistribute connected
PE3(config-router-af)# exit-address-family
PE3(config-router)# end
```

VPNA SiteA configuration

```
VPNA-SITEA> enable
VPNA-SITEA# configure terminal
VPNA-SITEA(config)# interface gigabitethernet 0/1
VPNA-SITEA(config-if-GigabitEthernet 0/1)# no switchport
VPNA-SITEA(config-if-GigabitEthernet 0/1) ip address 192.168.10.2
255.255.255.0
VPNA-SITEA(config-if-GigabitEthernet 0/1) exit
VPNA-SITEA(config)# router bgp 65001
VPNA-SITEA(config-router)# neighbor 192.168.10.1 remote-as 1
VPNA-SITEA(config-router)# neighbor 192.168.10.1 activate
VPNA-SITEA(config-router)# redistribute connected
VPNA-SITEA(config-router)# end
```

VPNA SiteB configuration

```
VPNA-SITEB> enable
VPNA-SITEB# configure terminal
VPNA-SITEB(config)# interface gigabitethernet 0/1
VPNA-SITEB(config-if-GigabitEthernet 0/1)# no switchport
VPNA-SITEB(config-if-GigabitEthernet 0/1) ip address 192.168.20.2
255.255.255.0
VPNA-SITEB(config-if-GigabitEthernet 0/1) exit
VPNA-SITEB(config)# router bgp 65002
VPNA-SITEB(config-router)# neighbor 192.168.20.1 remote-as 1
VPNA-SITEB(config-router)# neighbor 192.168.20.1 activate
VPNA-SITEB(config-router)# redistribute connected
VPNA-SITEB(config-router)# end
```

VPNA SiteC configuration

```
VPNA-SITEC> enable
VPNA-SITEC# configure terminal
VPNA-SITEC(config)# interface gigabitethernet 0/1
VPNA-SITEC(config-if-GigabitEthernet 0/1)# no switchport
VPNA-SITEC(config-if-GigabitEthernet 0/1) ip address 192.168.40.2
255.255.255.0
VPNA-SITEC(config-if-GigabitEthernet 0/1) exit
VPNA-SITEC(config)# router bgp 65003
VPNA-SITEC(config-router)# neighbor 192.168.40.1 remote-as 1
VPNA-SITEC(config-router)# neighbor 192.168.40.1 activate
VPNA-SITEC(config-router)# redistribute connected
VPNA-SITEC(config-router)# end
```

VPNB SiteB configuration

```
VPNB-SITEB> enable
```

```
VPNB-SITEB# configure terminal
VPNB-SITEB(config)# interface gigabitethernet 0/1
VPNB-SITEB(config-if-GigabitEthernet 0/1)# no switchport
VPNB-SITEB(config-if-GigabitEthernet 0/1)# ip address 192.168.10.3
255.255.255.0
VPNB-SITEB(config-if-GigabitEthernet 0/1)# exit
VPNB-SITEB(config)# router bgp 65004
VPNB-SITEB(config-router)# neighbor 192.168.10.1 remote-as 1
VPNB-SITEB(config-router)# neighbor 192.168.10.1 activate
VPNB-SITEB(config-router)# redistribute connected
VPNB-SITEB(config-router)# end
```

VPNB SiteC configuration

```
VPNB-SITEC> enable
VPNB-SITEC# configure terminal
VPNB-SITEC(config)# interface gigabitethernet 0/1
VPNB-SITEC(config-if-GigabitEthernet 0/1)# no switchport
VPNB-SITEC(config-if-GigabitEthernet 0/1) ip address 192.168.30.2
255.255.255.0
VPNB-SITEC(config-if-GigabitEthernet 0/1) exit
VPNB-SITEC(config)# router bgp 65005
VPNB-SITEC(config-router)# neighbor 192.168.30.1 remote-as 1
VPNB-SITEC(config-router)# neighbor 192.168.30.1 activate
VPNB-SITEC(config-router)# redistribute connected
VPNB-SITEC(config-router)# end
```

5. Verification

After the configuration is completed, run the **ping** command to detect the connectivity between sites.

- Verify that PE1 can ping PE2, PE3, P1, P2, and VPNA SiteA.
- Verify that PE2 can ping PE1, PE3, P1, P2, VPNA SiteC, and VPNB SiteC.
- Verify that PE3 can ping PE1, PE2, P1, P2, VPNA SiteB, and VPNB SiteB.
- Verify that VPNA SiteA can ping VPNA SiteB and VPNA SiteC and cannot ping VPNB SiteB or VPNB SiteC.
- Verify that VPNA SiteB can ping VPNA SiteA and VPNA SiteC and cannot ping VPNB SiteB or VPNB SiteC.
- Verify that VPNA SiteC can ping VPNA SiteA and VPNA SiteB and cannot ping VPNB SiteB or VPNB SiteC.
- Verify that VPNB SiteB can ping VPNB SiteC and cannot ping VPNA SiteA, VPNA SiteB, or VPNA SiteC.
- Verify that VPNB SiteC can ping VPNB SiteB and cannot ping VPNA SiteA, VPNA SiteB, or VPNA SiteC.
- Verify that P1 can ping PE1, PE2, PE3, and P2 but cannot ping VPN sites.
- Verify that P2 can ping PE1, PE2, PE3, and P1 but cannot ping VPN sites.

6. Configuration Files

- PE1 configuration file

```
hostname PE1
!
ip vrf VPNA
```

```
 rd 1:100
 route-target both 1:100
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.10.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.10.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.1 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 neighbor 172.168.0.3 remote-as 1
 neighbor 172.168.0.3 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.2 activate
  neighbor 172.168.0.3 activate
  exit-address-family
  !
 address-family ipv4 vrf VPNA
  neighbor 192.168.10.2 remote-as 65001
  neighbor 192.168.10.2 activate
  redistribute connected
  exit-address-family
!
router ospf 10
 network 172.168.0.1 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- P1 configuration file

```
hostname P1
```

```
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.10.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.20.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.4 255.255.255.255
!
router ospf 10
 network 172.168.0.4 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
 network 172.168.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- P2 configuration file

```
hostname P2
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.20.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.30.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/3
 no switchport
 ip address 172.168.40.1 255.255.255.0
```

```
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.5 255.255.255.255
!
router ospf 10
 network 172.168.0.5 0.0.0.0 area 0
 network 172.168.20.0 0.0.0.255 area 0
 network 172.168.30.0 0.0.0.255 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- PE2 configuration file

```
hostname PE2
!
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
ip vrf VPNB
 rd 1:200
 route-target both 1:200
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.40.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.40.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/3
 no switchport
 ip vrf forwarding VPNB
 ip address 172.168.30.1 255.255.255.0
!
interface Loopback 0
 ip address 172.168.0.3 255.255.255.255
```

```
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
  neighbor 172.168.0.2 activate
  exit-address-family
  !
 address-family ipv4 vrf VPNA
  neighbor 192.168.40.2 remote-as 65003
  neighbor 192.168.40.2 activate
  redistribute connected
  exit-address-family
  !
 address-family ipv4 vrf VPNB
  neighbor 192.168.30.2 remote-as 65005
  neighbor 192.168.30.2 activate
  exit-address-family
  redistribute connected
!
router ospf 10
 network 172.168.0.3 0.0.0.0 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- PE3 configuration file

```
hostname PE3
!
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
ip vrf VPNB
 rd 1:200
 route-target both 1:200
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPNB
```

```
 ip address 192.168.10.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.30.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/3
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.20.1 255.255.255.0
!
interface Loopback 0
 ip address 172.168.0.2 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 neighbor 172.168.0.3 remote-as 1
 neighbor 172.168.0.3 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
  neighbor 172.168.0.3 activate
  exit-address-family
  !
 address-family ipv4 vrf VPNA
  neighbor 192.168.20.2 remote-as 65002
  neighbor 192.168.20.2 activate
  redistribute connected
  exit-address-family
  !
 address-family ipv4 vrf VPNB
  neighbor 192.168.10.3 remote-as 65004
  neighbor 192.168.10.3 activate
  exit-address-family
  redistribute connected
!
router ospf 10
 network 172.168.0.2 0.0.0.0 area 0
 network 172.168.30.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- VPNA SiteA configuration file

```
hostname VPNA-SITEA
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.10.2 255.255.255.0
!
router bgp 65001
 neighbor 192.168.10.1 remote-as 1
address-family ipv4
  redistribute connected
 neighbor 192.168.10.1 activate
  exit-address-family
!
```

- VPNA SiteB configuration file

```
hostname VPNA-SITEB
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.20.2 255.255.255.0
!
router bgp 65002
 neighbor 192.168.20.1 remote-as 1
address-family ipv4
  redistribute connected
 neighbor 192.168.20.1 activate
  exit-address-family
!
```

- VPNA SiteC configuration file

```
hostname VPNA-SITEC
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.40.2 255.255.255.0
!
router bgp 65003
 neighbor 192.168.40.1 remote-as 1
address-family ipv4
  redistribute connected
 neighbor 192.168.40.1 activate
  exit-address-family
!
```

- VPNB SiteB configuration file

```
hostname VPNB-SITEB
!
interface GigabitEthernet 0/1
```

```
 no switchport
 ip address 192.168.10.3 255.255.255.0
!
router bgp 65004
 neighbor 192.168.10.1 remote-as 1
address-family ipv4
  redistribute connected
 neighbor 192.168.10.1 activate
  exit-address-family
!
```

- VPNB SiteC configuration file

```
hostname VPNB-SITEC
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.30.2 255.255.255.0
!
router bgp 65005
 neighbor 192.168.30.1 remote-as 1
address-family ipv4
  redistribute connected
 neighbor 192.168.30.1 activate
  exit-address-family
!
```

7. Common Errors

The router ID is not 32 bits. As a result, the LDP session or BGP neighbor relationship fails to be established.

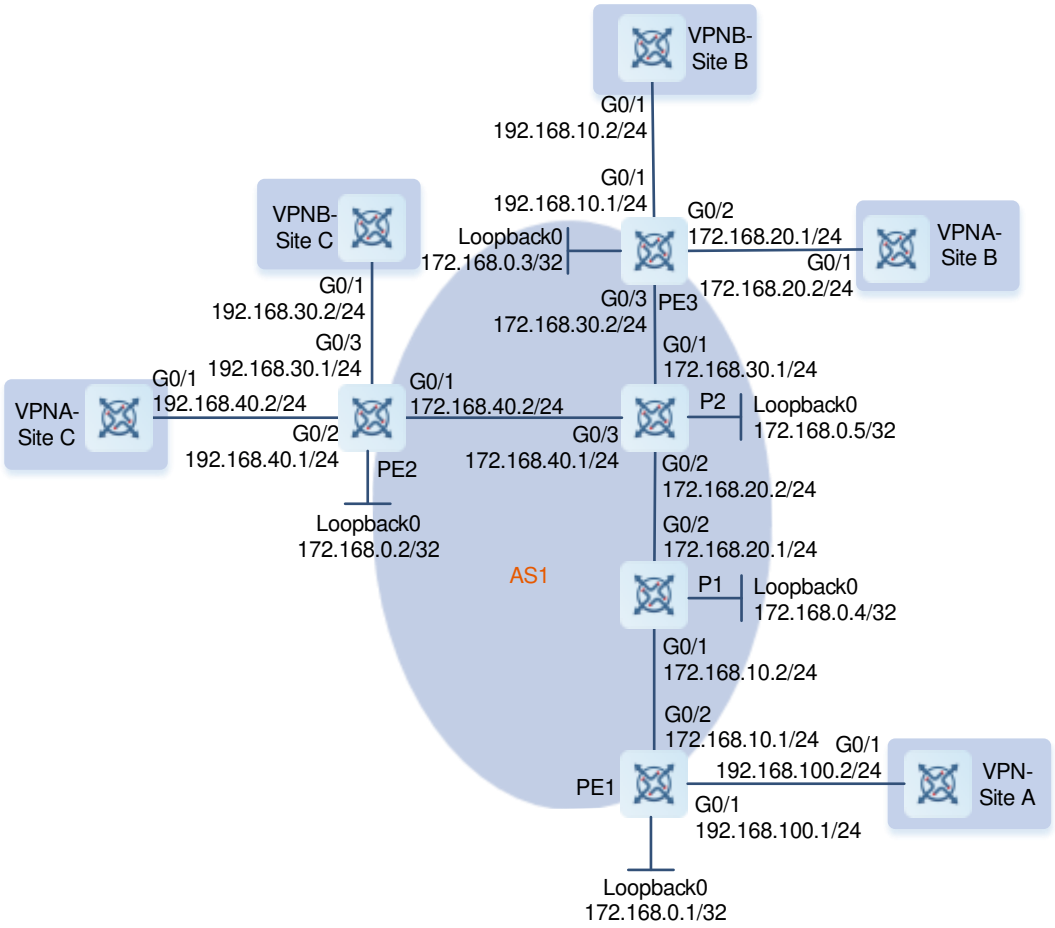
1.16.2 Configuring Basic IPv4 MPLS L3VPN Functions (Extranet)

1. Requirements

There are two VPNs: VPNA and VPNB. It is required that users within a VPN can communicate with each other, users in different VPNs cannot communicate with each other, and the two VPNs can access shared resources. As shown in Figure 1-20, VPNA and VPNB sites can access resources of VPN SiteA.

2. Topology

Figure 1-20 Configuring Basic IPv4 MPLS L3VPN Functions (Extranet)



3. Notes

- Configure interface IP addresses and OSPF on MPLS backbone network nodes to ensure communication between them.
- Enable MPLS forwarding and LDP and establish LDP LSPs on MPLS backbone network nodes.
- Configure VPN routing instances, define RD and RT values, and associate VRF instances with interfaces on PEs.
- Configure MP-IBGP neighbors on PEs to exchange VPN routing information.
- Configure OSPF between different VPN sites and PEs to exchange VPN routing information.

4. Procedure

(1) Configure interface IP addresses and OSPF on MPLS backbone network nodes to ensure communication between them.

Configure PE1.

```
PE1> enable
PE1# configure terminal
```

```
PE1(config)# interface loopback 0
PE1(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
PE1(config-if-Loopback 0)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# no switchport
PE1(config-if-GigabitEthernet 0/2)# ip address 172.168.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# exit
PE1(config)# router ospf 1
PE1(config-router)# network 172.168.0.1 0.0.0.0 area 0
PE1(config-router)# network 172.168.10.0 0.0.0.255 area 0
PE1(config-router)# exit
```

Configure P1.

```
P1> enable
P1# configure terminal
P1(config)# interface loopback 0
P1(config-if-Loopback 0)# ip address 172.168.0.4 255.255.255.255
P1(config-if-Loopback 0)# exit
P1(config)# interface gigabitethernet 0/1
P1(config-if-GigabitEthernet 0/1)# no switchport
P1(config-if-GigabitEthernet 0/1)# ip address 172.168.10.2 255.255.255.0
P1(config-if-GigabitEthernet 0/1)# exit
P1(config)# interface gigabitethernet 0/2
P1(config-if-GigabitEthernet 0/2)# no switchport
P1(config-if-GigabitEthernet 0/2)# ip address 172.168.20.1 255.255.255.0
P1(config-if-GigabitEthernet 0/2)# exit
P1(config)# router ospf 1
P1(config-router)# network 172.168.0.4 0.0.0.0 area 0
P1(config-router)# network 172.168.10.0 0.0.0.255 area 0
P1(config-router)# network 172.168.20.0 0.0.0.255 area 0
P1(config-router)# exit
```

Configure P2.

```
P2> enable
P2# configure terminal
P2(config)# interface loopback 0
P2(config-if-Loopback 0)# ip address 172.168.0.5 255.255.255.255
P2(config-if-Loopback 0)# exit
P2(config)# interface gigabitethernet 0/1
P2(config-if-GigabitEthernet 0/1)# no switchport
P2(config-if-GigabitEthernet 0/1)# ip address 172.168.30.1 255.255.255.0
P2(config-if-GigabitEthernet 0/1)# exit
P2(config)# interface gigabitethernet 0/2
P2(config-if-GigabitEthernet 0/2)# no switchport
P2(config-if-GigabitEthernet 0/2)# ip address 172.168.20.2 255.255.255.0
P2(config-if-GigabitEthernet 0/2)# exit
P2(config)# interface gigabitethernet 0/3
```

```
P2(config-if-GigabitEthernet 0/3)# no switchport
P2(config-if-GigabitEthernet 0/3)# ip address 172.168.40.1 255.255.255.0
P2(config-if-GigabitEthernet 0/3)# exit
P2(config)# router ospf 1
P2(config-router)# network 172.168.0.5 0.0.0.0 area 0
P2(config-router)# network 172.168.20.0 0.0.0.255 area 0
P2(config-router)# network 172.168.30.0 0.0.0.255 area 0
P2(config-router)# network 172.168.40.0 0.0.0.255 area 0
P2(config-router)# exit
```

Configure PE2.

```
PE2> enable
PE2# configure terminal
PE2(config)# interface loopback 0
PE2(config-if-Loopback 0)# ip address 172.168.0.2 255.255.255.255
PE2(config-if-Loopback 0)# exit
PE2(config)# interface gigabitethernet 0/1
PE2(config-if-GigabitEthernet 0/1)# no switchport
PE2(config-if-GigabitEthernet 0/1)# ip address 172.168.40.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/1)# exit
PE2(config)# router ospf 1
PE2(config-router)# network 172.168.0.2 0.0.0.0 area 0
PE2(config-router)# network 172.168.40.0 0.0.0.255 area 0
PE2(config-router)# exit
```

Configure PE3.

```
PE3> enable
PE3# configure terminal
PE3(config)# interface loopback 0
PE3(config-if-Loopback 0)# ip address 172.168.0.3 255.255.255.255
PE3(config-if-Loopback 0)# exit
PE3(config)# interface gigabitethernet 0/3
PE3(config-if-GigabitEthernet 0/3)# no switchport
PE3(config-if-GigabitEthernet 0/3)# ip address 172.168.30.2 255.255.255.0
PE3(config-if-GigabitEthernet 0/3)# exit
PE3(config)# router ospf 1
PE3(config-router)# network 172.168.0.3 0.0.0.0 area 0
PE3(config-router)# network 172.168.30.0 0.0.0.255 area 0
PE3(config-router)# exit
```

(2) Enable MPLS forwarding and LDP and establish LDP LSPs on MPLS backbone network nodes.

Configure PE1.

```
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/2
```

```
PE1(config-if-GigabitEthernet 0/2)# label-switching
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/2)# exit
```

Configure P1.

```
P1(config)# mpls enable
P1(config)# mpls router ldp
P1(config-mpls-router)# ldp router-id interface loopback 0 force
P1(config-mpls-router)# exit
P1(config)# interface gigabitethernet 0/1
P1(config-if-GigabitEthernet 0/1)# label-switching
P1(config-if-GigabitEthernet 0/1)# mpls ldp enable
P1(config-if-GigabitEthernet 0/1)# exit
P1(config)# interface gigabitethernet 0/2
P1(config-if-GigabitEthernet 0/2)# label-switching
P1(config-if-GigabitEthernet 0/2)# mpls ldp enable
P1(config-if-GigabitEthernet 0/2)# exit
```

Configure P2.

```
P2(config)# mpls enable
P2(config)# mpls router ldp
P2(config-mpls-router)# ldp router-id interface loopback 0 force
P2(config-mpls-router)# exit
P2(config)# interface gigabitethernet 0/1
P2(config-if-GigabitEthernet 0/1)# label-switching
P2(config-if-GigabitEthernet 0/1)# mpls ldp enable
P2(config-if-GigabitEthernet 0/1)# exit
P2(config)# interface gigabitethernet 0/2
P2(config-if-GigabitEthernet 0/2)# label-switching
P2(config-if-GigabitEthernet 0/2)# mpls ldp enable
P2(config-if-GigabitEthernet 0/2)# exit
P2(config)# interface gigabitethernet 0/3
P2(config-if-GigabitEthernet 0/3)# label-switching
P2(config-if-GigabitEthernet 0/3)# mpls ldp enable
P2(config-if-GigabitEthernet 0/3)# exit
```

Configure PE2.

```
PE2(config)# mpls enable
PE2(config)# mpls router ldp
PE2(config-mpls-router)# ldp router-id interface loopback 0 force
PE2(config-mpls-router)# exit
PE2(config)# interface gigabitethernet 0/1
PE2(config-if-GigabitEthernet 0/1)# label-switching
PE2(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE2(config-if-GigabitEthernet 0/1)# exit
```

Configure PE3.

```
PE3(config)# mpls enable
```

```
PE3(config)# mpls router ldp
PE3(config-mpls-router)# ldp router-id interface loopback 0 force
PE3(config-mpls-router)# exit
PE3(config)# interface gigabitethernet 0/3
PE3(config-if-GigabitEthernet 0/3)# label-switching
PE3(config-if-GigabitEthernet 0/3)# mpls ldp enable
PE3(config-if-GigabitEthernet 0/3)# exit
```

(3) Configure VPN routing instances, define RD and RT values, and associate VRF instances with interfaces on PEs.

Configure PE1.

```
PE1(config)# ip vrf VPN_EXTRA
PE1(config-vrf)# rd 1:100
PE1(config-vrf)# route-target both 1:100
PE1(config-vrf)# route-target both 1:200
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# no switchport
PE1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPN_EXTRA
PE1(config-if-GigabitEthernet 0/1)# ip address 192.168.100.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# exit
```

Configure PE2.

```
PE2(config)# ip vrf VPNA
PE2(config-vrf)# rd 1:100
PE2(config-vrf)# route-target both 1:100
PE2(config-vrf)# exit
PE2(config)# ip vrf VPNB
PE2(config-vrf)# rd 1:200
PE2(config-vrf)# route-target both 1:200
PE2(config-vrf)# exit
PE2(config)# interface gigabitethernet 0/2
PE2(config-if-GigabitEthernet 0/2)# no switchport
PE2(config-if-GigabitEthernet 0/2)# ip vrf forwarding VPNA
PE2(config-if-GigabitEthernet 0/2)# ip address 192.168.40.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/2)# exit
PE2(config)# interface gigabitethernet 0/3
PE2(config-if-GigabitEthernet 0/3)# no switchport
PE2(config-if-GigabitEthernet 0/3)# ip vrf forwarding VPNB
PE2(config-if-GigabitEthernet 0/3)# ip address 192.168.30.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/3)# exit
```

Configure PE3.

```
PE3(config)# ip vrf VPNA
PE3(config-vrf)# rd 1:100
PE3(config-vrf)# route-target both 1:100
PE3(config-vrf)# exit
```

```
PE3(config)# ip vrf VPNB
PE3(config-vrf)# rd 1:200
PE3(config-vrf)# route-target both 1:200
PE3(config-vrf)# exit
PE3(config)# interface gigabitethernet 0/1
PE3(config-if-GigabitEthernet 0/1)# no switchport
PE3(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPNB
PE3(config-if-GigabitEthernet 0/1)# ip address 192.168.10.1 255.255.255.0
PE3(config-if-GigabitEthernet 0/1)# exit
PE3(config)# interface gigabitethernet 0/2
PE3(config-if-GigabitEthernet 0/2)# ip vrf forwarding VPNA
PE3(config-if-GigabitEthernet 0/2)# ip address 192.168.20.1 255.255.255.0
PE3(config-if-GigabitEthernet 0/2)# exit
```

(4) Configure MP-IBGP neighbors on PEs to exchange VPN routing information.

Configure PE1.

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 172.168.0.2 remote-as 1
PE1(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE1(config-router)# neighbor 172.168.0.3 remote-as 1
PE1(config-router)# neighbor 172.168.0.3 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 172.168.0.2 activate
PE1(config-router-af)# neighbor 172.168.0.3 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

Configure PE2.

```
PE2(config)# router bgp 1
PE2(config-router)# neighbor 172.168.0.1 remote-as 1
PE2(config-router)# neighbor 172.168.0.1 update-source loopback 0
PE2(config-router)# neighbor 172.168.0.3 remote-as 1
PE2(config-router)# neighbor 172.168.0.3 update-source loopback 0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 172.168.0.1 activate
PE2(config-router-af)# neighbor 172.168.0.3 activate
PE2(config-router-af)# exit-address-family
PE2(config-router)# exit
```

Configure PE3.

```
PE3(config)# router bgp 1
PE3(config-router)# neighbor 172.168.0.1 remote-as 1
PE3(config-router)# neighbor 172.168.0.1 update-source loopback 0
PE3(config-router)# neighbor 172.168.0.2 remote-as 1
PE3(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE3(config-router)# address-family vpnv4
PE3(config-router-af)# neighbor 172.168.0.1 activate
```

```
PE3(config-router-af)# neighbor 172.168.0.2 activate
PE3(config-router-af)# exit-address-family
PE3(config-router)# exit
```

(5) Configure OSPF between different VPN sites and PEs to exchange VPN routing information.

Configure PE1.

```
PE1(config)# router ospf 10 vrf VPN_EXTRA
PE1(config-router)# network 192.168.100.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# exit
PE1(config)# router bgp 1
PE1(config-router)# address-family ipv4 vrf VPN_EXTRA
PE1(config-router-af)# redistribute ospf 10
PE1(config-router-af)# exit-address-family
PE1(config-router)# end
```

Configure PE2.

```
PE2(config)# router ospf 10 vrf VPNA
PE2(config-router)# network 192.168.40.0 0.0.0.255 area 0
PE2(config-router)# redistribute bgp subnets
PE2(config-router)# exit
PE2(config)# router ospf 20 vrf VPNB
PE2(config-router)# network 192.168.30.0 0.0.0.255 area 0
PE2(config-router)# redistribute bgp subnets
PE2(config-router)# exit
PE2(config)# router bgp 1
PE2(config-router)# address-family ipv4 vrf VPNA
PE2(config-router-af)# redistribute ospf 10
PE2(config-router-af)# exit-address-family
PE2(config-router)# address-family ipv4 vrf VPNB
PE2(config-router-af)# redistribute ospf 20
PE2(config-router-af)# exit-address-family
PE2(config-router)# end
```

Configure PE3.

```
PE3(config)# router ospf 10 vrf VPNA
PE3(config-router)# network 192.168.20.0 0.0.0.255 area 0
PE3(config-router)# redistribute bgp subnets
PE3(config-router)# exit
PE3(config)# router ospf 20 vrf VPNB
PE3(config-router)# network 192.168.10.0 0.0.0.255 area 0
PE3(config-router)# redistribute bgp subnets
PE3(config-router)# exit
PE3(config)# router bgp 1
PE3(config-router)# address-family ipv4 vrf VPNA
PE3(config-router-af)# redistribute ospf 10
PE3(config-router-af)# exit
```

```
PE3(config-router)# address-family ipv4 vrf VPNB
PE3(config-router-af)# redistribute ospf 20
PE3(config-router-af)# exit-address-family
PE3(config-router)# end
```

VPN SiteA configuration

```
VPN-SITEA> enable
VPN-SITEA# configure terminal
VPN-SITEA(config)# interface gigabitethernet 0/1
VPN-SITEA(config-if-GigabitEthernet 0/1)# no switchport
VPN-SITEA(config-if-GigabitEthernet 0/1)# ip address 192.168.100.2
255.255.255.0
VPN-SITEA(config-if-GigabitEthernet 0/1)# exit
VPN-SITEA(config)# router ospf 10
VPN-SITEA(config-router)# network 192.168.100.0 0.0.0.255 area 0
VPN-SITEA(config-router)# end
```

VPNA SiteB configuration

```
VPNA-SITEB> enable
VPNA-SITEB# configure terminal
VPNA-SITEB(config)# interface gigabitethernet 0/1
VPNA-SITEA(config-if-GigabitEthernet 0/1)# no switchport
VPNA-SITEB(config-if-GigabitEthernet 0/1)# ip address 192.168.20.2
255.255.255.0
VPNA-SITEB(config-if-GigabitEthernet 0/1)# exit
VPNA-SITEB(config)# router ospf 10
VPNA-SITEB(config-router)# network 192.168.20.0 0.0.0.255 area 0
VPNA-SITEB(config-router)# end
```

VPNB SiteB configuration

```
VPNB-SITEB> enable
VPNB-SITEB# configure terminal
VPNB-SITEB(config)# interface gigabitethernet 0/1
VPNB-SITEA(config-if-GigabitEthernet 0/1)# no switchport
VPNB-SITEB(config-if-GigabitEthernet 0/1)# ip address 192.168.10.2
255.255.255.0
VPNB-SITEB(config-if-GigabitEthernet 0/1)# exit
VPNB-SITEB(config)# router ospf 10
VPNB-SITEB(config-router)# network 192.168.10.0 0.0.0.255 area 0
VPNB-SITEB(config-router)# end
```

VPNA SiteC configuration

```
VPNA-SITEC> enable
VPNA-SITEC# configure terminal
VPNA-SITEC(config)# interface gigabitethernet 0/1
VPNA-SITEC(config-if-GigabitEthernet 0/1)# no switchport
VPNA-SITEC(config-if-GigabitEthernet 0/1)# ip address 192.168.40.2
255.255.255.0
```

```
VPNA-SITEC(config-if-GigabitEthernet 0/1)# exit
VPNA-SITEC(config)# router ospf 10
VPNA-SITEC(config-router)# network 192.168.40.0 0.0.0.255 area 0
VPNA-SITEC(config-router)# end
```

VPNB SiteC configuration

```
VPNB-SITEC> enable
VPNB-SITEC# configure terminal
VPNB-SITEC(config)# interface gigabitethernet 0/1
VPNB-SITEC(config-if-GigabitEthernet 0/1)# no switchport
VPNB-SITEC(config-if-GigabitEthernet 0/1)# ip address 192.168.30.2
255.255.255.0
VPNB-SITEC(config-if-GigabitEthernet 0/1)# exit
VPNB-SITEC(config)# router ospf 10
VPNB-SITEC(config-router)# network 192.168.30.0 0.0.0.255 area 0
VPNB-SITEC(config-router)# end
```

5. Configuration Files

- PE1 configuration file

```
hostname PE1
!
ip vrf VPN_EXTRA
 rd 1:100
 route-target both 1:100
 route-target both 1:200
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN_EXTRA
 ip address 192.168.100.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.10.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.1 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 neighbor 172.168.0.3 remote-as 1
```

```
 neighbor 172.168.0.3 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.2 activate
  neighbor 172.168.0.3 activate
  exit-address-family
  !
 address-family ipv4 vrf VPN_EXTRA
  redistribute ospf 10
  exit-address-family
!
router ospf 1
 network 172.168.0.1 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN_EXTRA
 redistribute bgp subnets
 network 192.168.100.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- P1 configuration file

```
hostname P1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.10.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.20.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.4 255.255.255.255
!
router ospf 1
 network 172.168.0.4 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
 network 172.168.20.0 0.0.0.255 area 0
!
```

```
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- P2 configuration file

```
hostname P2
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.30.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.20.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/3
 no switchport
 ip address 172.168.40.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.5 255.255.255.255
!
router ospf 1
 network 172.168.0.5 0.0.0.0 area 0
 network 172.168.20.0 0.0.0.255 area 0
 network 172.168.30.0 0.0.0.255 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- PE2 configuration file

```
hostname PE2
!
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
```

```
ip vrf VPNB
 rd 1:200
 route-target both 1:200
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.40.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.40.1 255.255.255.0
!
interface GigabitEthernet 0/3
 no switchport
 ip vrf forwarding VPNB
 ip address 192.168.30.1 255.255.255.0
!
interface Loopback 0
 ip address 172.168.0.2 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 neighbor 172.168.0.3 remote-as 1
 neighbor 172.168.0.3 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
  neighbor 172.168.0.3 activate
  exit-address-family
  !
 address-family ipv4 vrf VPNA
  redistribute ospf 10
  exit-address-family
  !
 address-family ipv4 vrf VPNB
  redistribute ospf 20
  exit-address-family
!
router ospf 1
 network 172.168.0.2 0.0.0.0 area 0
 network 172.168.40.0 0.0.0.255 area 0
```

```
!
router ospf 10 vrf VPNA
redistribute bgp subnets
 network 192.168.40.0 0.0.0.255 area 0
!
router ospf 20 vrf VPNB
 redistribute bgp subnets
 network 192.168.30.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- PE3 configuration file

```
hostname PE3
!
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
ip vrf VPNB
 rd 1:200
 route-target both 1:200
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPNB
 ip address 192.168.10.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.20.1 255.255.255.0
!
interface GigabitEthernet 0/3
 no switchport
 ip address 172.168.30.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.3 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
```

```
 neighbor 172.168.0.1 update-source Loopback 0
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
  neighbor 172.168.0.2 activate
  exit-address-family
  !
 address-family ipv4 vrf VPNA
  redistribute ospf 10
  exit-address-family
  !
 address-family ipv4 vrf VPNB
  redistribute ospf 20
  exit-address-family
!
router ospf 1
 network 172.168.0.3 0.0.0.0 area 0
 network 172.168.30.0 0.0.0.255 area 0
!
router ospf 10 vrf VPNA
 redistribute bgp subnets
 network 192.168.20.0 0.0.0.255 area 0
!
router ospf 20 vrf VPNB
 redistribute bgp subnets
 network 192.168.10.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- VPN SiteA configuration file

```
hostname VPN-SITEA
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.100.2 255.255.255.0
!
router ospf 10
 network 192.168.100.0 0.0.0.255 area 0
!
```

- VPNA SiteB configuration file

```
hostname VPNA-SITEB
!
interface GigabitEthernet 0/1
```

```
 no switchport
 ip address 192.168.20.2 255.255.255.0
!
router ospf 10
 network 192.168.20.0 0.0.0.255 area 0
!
```

- VPNB SiteB configuration file

```
hostname VPNB-SITEB
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.10.2 255.255.255.0
!
router ospf 10
 network 192.168.10.0 0.0.0.255 area 0
!
```

- VPNA SiteC configuration file

```
hostname VPNA-SITEC
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.40.2 255.255.255.0
!
router ospf 10
 network 192.168.40.0 0.0.0.255 area 0
!
```

- VPNB SiteC configuration file

```
hostname VPNB-SITEC
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.30.2 255.255.255.0
!
router ospf 10
 network 192.168.30.0 0.0.0.255 area 0
!
```

6. Verification

After the configuration is completed, run the **ping** command to detect the connectivity between sites.

- Verify that PE1 can ping PE2, PE3, P1, P2, and VPN SiteA.
- Verify that PE2 can ping PE1, PE3, P1, P2, VPNA SiteC, and VPNB SiteC.
- Verify that PE3 can ping PE1, PE2, P1, P2, VPNA SiteB, and VPNB SiteB.
- Verify that P1 can ping PE1, PE2, PE3, and P2.
- Verify that P2 can ping PE1, PE2, PE3, and P1.

- Verify that VPN SiteA can ping VPNA SiteB, VPNA SiteC, VPNB SiteB, and VPNB SiteC.
- Verify that VPNA SiteB can ping VPN SiteA and VPNA SiteC and cannot ping VPNB SiteB or VPNB SiteC.
- Verify that VPNA SiteC can ping VPN SiteA and VPNA SiteB and cannot ping VPNB SiteB or VPNB SiteC.
- Verify that VPNB SiteB can ping VPN SiteA and VPNB SiteC and cannot ping VPNA SiteB or VPNA SiteC.
- Verify that VPNB SiteC can ping VPN SiteA and VPNB SiteB and cannot ping VPNA SiteB or VPNA SiteC.

7. Common Errors

The router ID is not 32 bits. As a result, the LDP session or BGP neighbor relationship fails to be established.

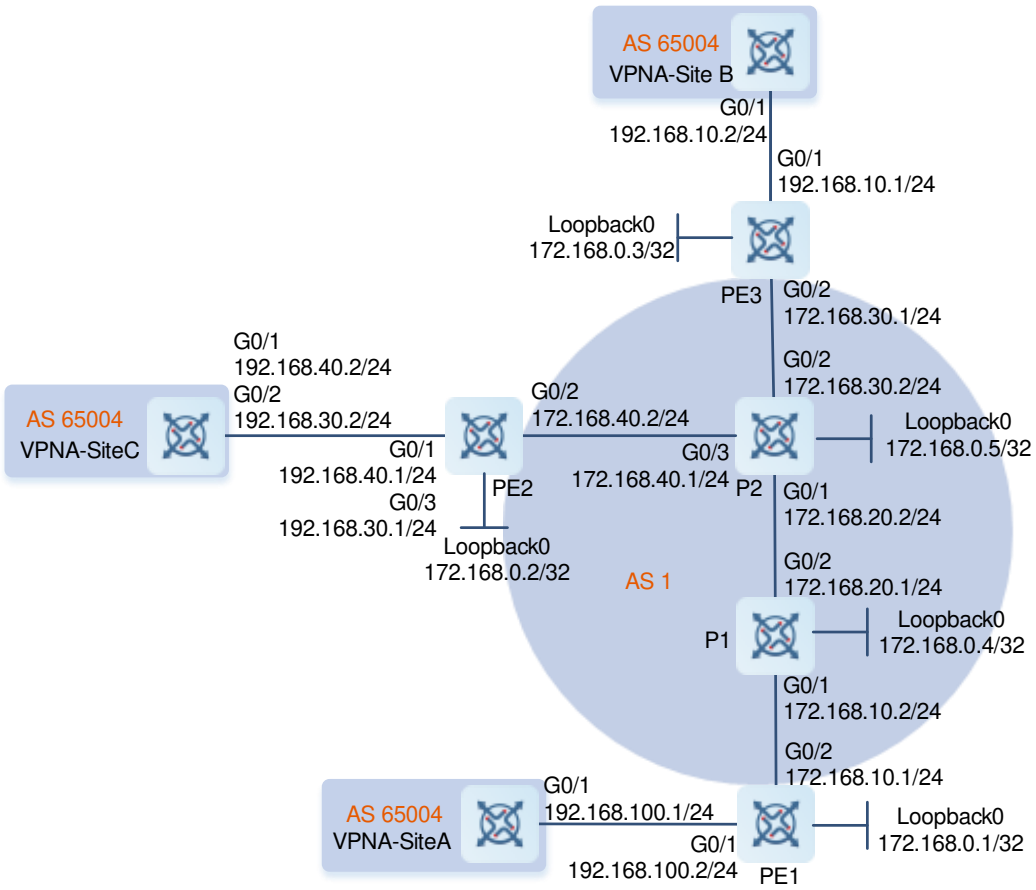
1.16.3 Configuring Basic IPv4 MPLS L3VPN Functions (Hub-and-Spoke)

1. Requirements

Data in a VPN cannot be exchanged directly and needs to be forwarded through a unified control center, and only the control center knows all resources in a VPN. Users in a VPN can obtain resources in the VPN only through the control center. As shown in Figure 1-21, VPNA SiteA can access resources in VPNA SiteB only through VPNA SiteC.

2. Topology

Figure 1-21 Configuring Basic IPv4 MPLS L3VPN Functions (Hub-and-Spoke)



3. Notes

- Configure interface IP addresses and OSPF on MPLS backbone network nodes to ensure communication between them.
- Enable MPLS forwarding and LDP and establish LDP LSPs on MPLS backbone network nodes.
- Configure VPN routing instance spoke1 on PE1, instances from-spoke and from-hub on PE2, and instance spoke2 on PE3, define RD and RT values, and associate VRF instances with interfaces.
- Configure MP-IBGP neighbors on PEs to exchange VPN routing information.
- Establish EBGP sessions between VPN sites and PEs to exchange VPN routing information.

4. Procedure

(1) Configure interface IP addresses and OSPF on MPLS backbone network nodes to ensure communication between them.

Configure PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# interface loopback 0
PE1(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
PE1(config-if-Loopback 0)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# no switchport
PE1(config-if-GigabitEthernet 0/2)# ip address 172.168.10.1 255.255.255.0
PE1(config)# router ospf 10
PE1(config-router)# network 172.168.0.1 0.0.0.0 area 0
PE1(config-router)# network 172.168.10.0 0.0.0.255 area 0
PE1(config-router)# exit
```

Configure P1.

```
P1> enable
P1# configure terminal
P1(config)# interface loopback 0
P1(config-if-Loopback 0)# ip address 172.168.0.4 255.255.255.255
P1(config-if-Loopback 0)# exit
P1(config)# interface gigabitethernet 0/1
P1(config-if-GigabitEthernet 0/1)# no switchport
P1(config-if-GigabitEthernet 0/1)# ip address 172.168.10.2 255.255.255.0
P1(config-if-GigabitEthernet 0/1)# exit
P1(config)# interface gigabitethernet 0/2
P1(config-if-GigabitEthernet 0/2)# no switchport
P1(config-if-GigabitEthernet 0/2)# ip address 172.168.20.1 255.255.255.0
P1(config-if-GigabitEthernet 0/2)# exit
P1(config)# router ospf 1
P1(config-router)# network 172.168.0.4 0.0.0.0 area 0
P1(config-router)# network 172.168.10.0 0.0.0.255 area 0
P1(config-router)# network 172.168.20.0 0.0.0.255 area 0
```

```
P1(config-router)# exit
```

Configure P2.

```
P2> enable
P2# configure terminal
P2(config)# interface loopback 0
P2(config-if-Loopback 0)# ip address 172.168.0.5 255.255.255.255
P2(config-if-Loopback 0)# exit
P2(config)# interface gigabitethernet 0/1
P2(config-if-GigabitEthernet 0/1)# no switchport
P2(config-if-GigabitEthernet 0/1)# ip address 172.168.20.2 255.255.255.0
P2(config-if-GigabitEthernet 0/1)# exit
P2(config)# interface gigabitethernet 0/2
P2(config-if-GigabitEthernet 0/2)# no switchport
P2(config-if-GigabitEthernet 0/2)# ip address 172.168.30.1 255.255.255.0
P2(config-if-GigabitEthernet 0/2)# exit
P2(config)# interface gigabitethernet 0/3
P2(config-if-GigabitEthernet 0/3)# no switchport
P2(config-if-GigabitEthernet 0/3)# ip address 172.168.40.1 255.255.255.0
P2(config-if-GigabitEthernet 0/3)# exit
P2(config)# router ospf 1
P2(config-router)# network 172.168.0.5 0.0.0.0 area 0
P2(config-router)# network 172.168.20.0 0.0.0.255 area 0
P2(config-router)# network 172.168.30.0 0.0.0.255 area 0
P2(config-router)# network 172.168.40.0 0.0.0.255 area 0
P2(config-router)# exit
```

Configure PE2.

```
PE2> enable
PE2# configure terminal
PE2(config)# interface loopback 0
PE2(config-if-Loopback 0)# ip address 172.168.0.2 255.255.255.255
PE2(config-if-Loopback 0)# exit
PE2(config)# interface gigabitethernet 0/2
PE2(config-if-GigabitEthernet 0/2)# no switchport
PE2(config-if-GigabitEthernet 0/2)# ip address 172.168.40.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/2)# exit
PE2(config)# router ospf 10
PE2(config-router)# network 172.168.0.2 0.0.0.0 area 0
PE2(config-router)# network 172.168.40.0 0.0.0.255 area 0
PE2(config-router)# exit
```

Configure PE3.

```
PE3> enable
PE3# configure terminal
PE3(config)# interface loopback 0
PE3(config-if-Loopback 0)# ip address 172.168.0.3 255.255.255.255
PE3(config-if-Loopback 0)# exit
```

```
PE3(config)# interface gigabitethernet 0/2
PE3(config-if-GigabitEthernet 0/2)# no switchport
PE3(config-if-GigabitEthernet 0/2)# ip address 172.168.30.2 255.255.255.0
PE3(config-if-GigabitEthernet 0/2)# exit
PE3(config)# router ospf 10
PE3(config-router)# network 172.168.0.3 0.0.0.0 area 0
PE3(config-router)# network 172.168.30.0 0.0.0.255 area 0
PE3(config-router)# exit
```

(2) Enable MPLS forwarding and LDP and establish LDP LSPs on MPLS backbone network nodes.

Configure PE1.

```
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# label-switching
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/2)# exit
```

Configure P1.

```
P1(config)# mpls enable
P1(config)# mpls router ldp
P1(config-mpls-router)# ldp router-id interface loopback 0 force
P1(config-mpls-router)# exit
P1(config)# interface gigabitethernet 0/1
P1(config-if-GigabitEthernet 0/1)# label-switching
P1(config-if-GigabitEthernet 0/1)# mpls ldp enable
P1(config-if-GigabitEthernet 0/1)# exit
P1(config)# interface gigabitethernet 0/2
P1(config-if-GigabitEthernet 0/2)# label-switching
P1(config-if-GigabitEthernet 0/2)# mpls ldp enable
P1(config-if-GigabitEthernet 0/2)# exit
```

Configure P2.

```
P2(config)# mpls enable
P2(config)# mpls router ldp
P2(config-mpls-router)# ldp router-id interface loopback 0 force
P2(config-mpls-router)# exit
P2(config)# interface gigabitethernet 0/1
P2(config-if-GigabitEthernet 0/1)# label-switching
P2(config-if-GigabitEthernet 0/1)# mpls ldp enable
P2(config-if-GigabitEthernet 0/1)# exit
P2(config)# interface gigabitethernet 0/2
P2(config-if-GigabitEthernet 0/2)# label-switching
P2(config-if-GigabitEthernet 0/2)# mpls ldp enable
P2(config-if-GigabitEthernet 0/2)# exit
```

```
P2(config)# interface gigabitethernet 0/3
P2(config-if-GigabitEthernet 0/3)# label-switching
P2(config-if-GigabitEthernet 0/3)# mpls ldp enable
P2(config-if-GigabitEthernet 0/3)# exit
```

Configure PE2.

```
PE2(config)# mpls enable
PE2(config)# mpls router ldp
PE2(config-mpls-router)# ldp router-id interface loopback 0 force
PE2(config-mpls-router)# exit
PE2(config)# interface gigabitethernet 0/2
PE2(config-if-GigabitEthernet 0/2)# label-switching
PE2(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE2(config-if-GigabitEthernet 0/2)# exit
```

Configure PE3.

```
PE3(config)# mpls enable
PE3(config)# mpls router ldp
PE3(config-mpls-router)# ldp router-id interface loopback 0 force
PE3(config-mpls-router)# exit
PE3(config)# interface gigabitethernet 0/2
PE3(config-if-GigabitEthernet 0/2)# label-switching
PE3(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE3(config-if-GigabitEthernet 0/2)# exit
```

(3) Configure VPN routing instances, define RD and RT values, and associate VRF instances with interfaces on PEs.

Configure PE1.

```
PE1(config)# ip vrf spoke1
PE1(config-vrf)# rd 1:100
PE1(config-vrf)# route-target export 1:200
PE1(config-vrf)# route-target import 1:100
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# no switchport
PE1(config-if-GigabitEthernet 0/1)# ip vrf forwarding spoke1
PE1(config-if-GigabitEthernet 0/1)# ip address 192.168.100.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# exit
```

Configure PE2.

```
PE2(config)# ip vrf from-spoke
PE2(config-vrf)# rd 1:100
PE2(config-vrf)# route-target import 1:300
PE2(config-vrf)# route-target import 1:200
PE2(config-vrf)# exit
PE2(config)# ip vrf from-hub
PE2(config-vrf)# rd 1:200
PE2(config-vrf)# route-target export 1:100
```

```
PE2(config-vrf)# exit
PE2(config)# interface gigabitethernet 0/1
PE2(config-if-GigabitEthernet 0/1)# no switchport
PE2(config-if-GigabitEthernet 0/1)# ip vrf forwarding from-hub
PE2(config-if-GigabitEthernet 0/1)# ip address 192.168.40.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/1)# exit
PE2(config)# interface gigabitethernet 0/3
PE2(config-if-GigabitEthernet 0/3)# no switchport
PE2(config-if-GigabitEthernet 0/3)# ip vrf forwarding from-spoke
PE2(config-if-GigabitEthernet 0/3)# ip address 192.168.30.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/3)# exit
```

Configure PE3.

```
PE3(config)# ip vrf spoke2
PE3(config-vrf)# rd 1:100
PE3(config-vrf)# route-target export 1:300
PE3(config-vrf)# route-target import 1:100
PE3(config-vrf)# exit
PE3# configure terminal
PE3(config)# interface gigabitethernet 0/1
PE3(config-if-GigabitEthernet 0/1)# no switchport
PE3(config-if-GigabitEthernet 0/1)# ip vrf forwarding spoke2
PE3(config-if-GigabitEthernet 0/1)# ip address 192.168.10.1 255.255.255.0
PE3(config-if-GigabitEthernet 0/1)# exit
```

(4) Configure MP-IBGP neighbors on PEs to exchange VPN routing information.

Configure PE1.

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 172.168.0.2 remote-as 1
PE1(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE1(config-router)# neighbor 172.168.0.3 remote-as 1
PE1(config-router)# neighbor 172.168.0.3 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 172.168.0.2 activate
PE1(config-router-af)# neighbor 172.168.0.2 allowas-in
PE1(config-router-af)# neighbor 172.168.0.3 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

Configure PE2.

```
PE2(config)# router bgp 1
PE2(config-router)# neighbor 172.168.0.1 remote-as 1
PE2(config-router)# neighbor 172.168.0.1 update-source loopback 0
PE2(config-router)# neighbor 172.168.0.3 remote-as 1
PE2(config-router)# neighbor 172.168.0.3 update-source loopback 0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 172.168.0.1 activate
```

```
PE2(config-router-af)# neighbor 172.168.0.3 activate
PE2(config-router-af)# exit-address-family
PE2(config-router)# exit
```

Configure PE3.

```
PE3(config)# router bgp 1
PE3(config-router)# neighbor 172.168.0.1 remote-as 1
PE3(config-router)# neighbor 172.168.0.1 update-source loopback 0
PE3(config-router)# neighbor 172.168.0.2 remote-as 1
PE3(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE3(config-router)# address-family vpnv4
PE3(config-router-af)# neighbor 172.168.0.1 activate
PE3(config-router-af)# neighbor 172.168.0.2 activate
PE3(config-router-af)# neighbor 172.168.0.2 allowas-in
PE3(config-router-af)# exit-address-family
PE3(config-router)# exit
```

(5) Establish EBGP sessions between VPN sites and PEs to exchange VPN routing information.

Configure PE1.

```
PE1(config)# router bgp 1
PE1(config-router)# address-family ipv4 vrf spoke1
PE1(config-router-af)# neighbor 192.168.100.2 remote-as 65004
PE1(config-router-af)# neighbor 192.168.100.2 activate
PE1(config-router-af)# neighbor 192.168.100.2 as-override
PE1(config-router-af)# exit-address-family
PE1(config-router)# end
```

Configure PE2.

```
PE2(config)# router bgp 1
PE2(config-router)# address-family ipv4 vrf from-spoke
PE2(config-router-af)# neighbor 192.168.30.2 remote-as 65004
PE2(config-router-af)# neighbor 192.168.30.2 activate
PE2(config-router-af)# neighbor 192.168.30.2 as-override
PE2(config-router-af)# exit-address-family
PE2(config-router)# address-family ipv4 vrf from-hub
PE2(config-router-af)# neighbor 192.168.40.2 remote-as 65004
PE2(config-router-af)# neighbor 192.168.40.2 activate
PE2(config-router-af)# neighbor 192.168.40.2 allows-in
PE2(config-router-af)# exit-address-family
PE2(config-router)# end
```

Configure PE3.

```
PE3(config)# router bgp 1
PE3(config-router)# address-family ipv4 vrf spoke2
PE3(config-router-af)# neighbor 192.168.10.2 remote-as 65004
PE3(config-router-af)# neighbor 192.168.10.2 activate
PE3(config-router-af)# neighbor 192.168.10.2 as-override
PE3(config-router-af)# exit-address-family
```

```
PE3(config-router)# end
```

VPNA SiteA configuration

```
VPNA-SITEA> enable
VPNA-SITEA# configure terminal
VPNA-SITEA(config)# interface gigabitethernet 0/1
VPNA-SITEA(config-if-GigabitEthernet 0/1)# ip address 192.168.100.2
255.255.255.0
VPNA-SITEA(config-if-GigabitEthernet 0/1)# exit
VPNA-SITEA(config)# router bgp 65004
VPNA-SITEA(config-router)# neighbor 192.168.100.1 remote-as 1
VPNA-SITEA(config-router)# neighbor 192.168.100.1 activate
VPNA-SITEA(config-router)# redistribute connected
VPNA-SITEA(config-router)# end
```

VPNA SiteB configuration

```
VPNA-SITEB> enable
VPNA-SITEB# configure terminal
VPNA-SITEB(config)# interface gigabitethernet 0/1
VPNA-SITEB(config-if-GigabitEthernet 0/1)# ip address 192.168.10.2
255.255.255.0
VPNA-SITEB(config-if-GigabitEthernet 0/1)# exit
VPNA-SITEB(config)# router bgp 65004
VPNA-SITEB(config-router)# neighbor 192.168.10.1 remote-as 1
VPNA-SITEB(config-router)# neighbor 192.168.10.1 activate
VPNA-SITEB(config-router)# redistribute connected
VPNA-SITEB(config-router)# end
```

VPNA SiteC configuration

```
VPNA-SITEC> enable
VPNA-SITEC# configure terminal
VPNA-SITEC(config)# interface gigabitethernet 0/1
VPNA-SITEC(config-if-GigabitEthernet 0/1)# ip address 192.168.40.2
255.255.255.0
VPNA-SITEC(config-if-GigabitEthernet 0/1)# exit
VPNA-SITEC(config)# interface gigabitethernet 0/2
VPNA-SITEC(config-if-GigabitEthernet 0/2)# ip address 192.168.30.2
255.255.255.0
VPNA-SITEC(config-if-GigabitEthernet 0/2)# exit
VPNA-SITEC(config)# router bgp 65004
VPNA-SITEC(config-router)# neighbor 192.168.30.1 remote-as 1
VPNA-SITEC(config-router)# neighbor 192.168.30.1 activate
VPNA-SITEC(config-router)# neighbor 192.168.40.1 remote-as 1
VPNA-SITEC(config-router)# neighbor 192.168.40.1 activate
VPNA-SITEC(config-router)# redistribute connected
VPNA-SITEC(config-router)# end
```

5. Configuration Files

- PE1 configuration file

```
hostname PE1
!
ip vrf spoke1
 rd 1:100
 route-target export 1:200
 route-target import 1:100
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding spoke1
 ip address 192.168.100.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.10.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.1 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 neighbor 172.168.0.3 remote-as 1
 neighbor 172.168.0.3 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.2 activate
  neighbor 172.168.0.2 allowas-in
  neighbor 172.168.0.3 activate
  exit-address-family
  !
 address-family ipv4 vrf spoke1
  neighbor 192.168.100.2 remote-as 65004
  neighbor 192.168.100.2 activate
  neighbor 192.168.100.2 as-override
  exit-address-family
!
router ospf 10
 network 172.168.0.1 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
```

```
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- P1 configuration file

```
hostname P1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.10.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.20.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.4 255.255.255.255
!
router ospf 1
 network 172.168.0.4 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
 network 172.168.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- P2 configuration file

```
hostname P2
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.20.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
```

```
 ip address 172.168.30.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/3
 no switchport
 ip address 172.168.40.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.5 255.255.255.255
!
router ospf 1
 network 172.168.0.5 0.0.0.0 area 0
 network 172.168.20.0 0.0.0.255 area 0
 network 172.168.30.0 0.0.0.255 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- PE2 configuration file

```
hostname PE2
!
ip vrf from-hub
 rd 1:200
 route-target export 1:100
!
ip vrf from-spoke
 rd 1:100
 route-target import 1:300
 route-target import 1:200
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding from-hub
 ip address 192.168.40.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.40.2 255.255.255.0
 mpls ldp enable
 label-switching
```

```
!
interface GigabitEthernet 0/3
 no switchport
 ip vrf forwarding from-spoke
 ip address 192.168.30.1 255.255.255.0
!
interface Loopback 0
 ip address 172.168.0.2 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 neighbor 172.168.0.3 remote-as 1
 neighbor 172.168.0.3 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
  neighbor 172.168.0.3 activate
  exit-address-family
  !
 address-family ipv4 vrf from-spoke
  neighbor 192.168.30.2 remote-as 65004
  neighbor 192.168.30.2 activate
  neighbor 192.168.30.2 as-override
  exit-address-family
  !
 address-family ipv4 vrf from-hub
  neighbor 192.168.40.2 remote-as 65004
  neighbor 192.168.40.2 activate
  neighbor 192.168.40.2 allowas-in
  exit-address-family
!
router ospf 10
 network 172.168.0.2 0.0.0.0 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- PE3 configuration file

```
hostname PE3
!
ip vrf spoke2
 rd 1:100
 route-target export 1:300
 route-target import 1:100
```

```
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding spoke2
 ip address 192.168.10.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.30.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 172.168.0.3 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
  neighbor 172.168.0.2 activate
  neighbor 172.168.0.2 allowas-in
  exit-address-family
  !
 address-family ipv4 vrf spoke2
  neighbor 192.168.10.2 remote-as 65004
  neighbor 192.168.10.2 activate
  neighbor 192.168.10.2 as-override
  exit-address-family
  !
router ospf 10
 network 172.168.0.3 0.0.0.0 area 0
 network 172.168.30.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- VPN SiteA configuration file

```
hostname VPN-SITEA
!
interface GigabitEthernet 0/1
 no switchport
```

```
 ip address 192.168.100.2 255.255.255.0
!
router bgp 65004
 neighbor 192.168.100.1 remote-as 1
 address-family ipv4
  redistribute connected
  neighbor 192.168.100.1 activate
  exit-address-family
!
```

- VPNA SiteB configuration file

```
hostname VPNA-SITEB
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.10.2 255.255.255.0
!
router bgp 65004
 neighbor 192.168.10.1 remote-as 1
 address-family ipv4
  redistribute connected
  neighbor 192.168.10.1 activate
  exit-address-family
!
```

- VPNA SiteC configuration file

```
hostname VPNA-SITEC
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.40.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 192.168.30.2 255.255.255.0
!
router bgp 65004
 neighbor 192.168.30.1 remote-as 1
 neighbor 192.168.40.1 remote-as 1
 address-family ipv4
  redistribute connected
  neighbor 192.168.30.1 activate
  neighbor 192.168.40.1 activate
  exit-address-family
!
```

6. Verification

After the configuration is completed, run the **ping** command to detect the connectivity between sites and run the **traceroute** command to trace bypassing devices.

- Verify that PE1 can ping PE2 and PE3.
- Verify that PE2 can ping PE1 and PE3.
- Verify that PE3 can ping PE1 and PE2.
- Verify that VPNA SiteA can ping VPNA SiteB, and traffic from VPNA SiteA to VPNA SiteB passes through VPNA SiteC.
- Verify that VPNA SiteB can ping VPNA SiteA, and traffic from VPNA SiteB to VPNA SiteA passes through VPNA SiteC.
- Verify that VPNA SiteC can ping VPNA SiteA and VPNA SiteB.

7. Common Errors

The router ID is not 32 bits. As a result, the LDP session or BGP neighbor relationship fails to be established.

1.16.4 Configuring Basic IPv4 MPLS L3VPN Functions (Unified Egress for Centralized Internet Access Control)

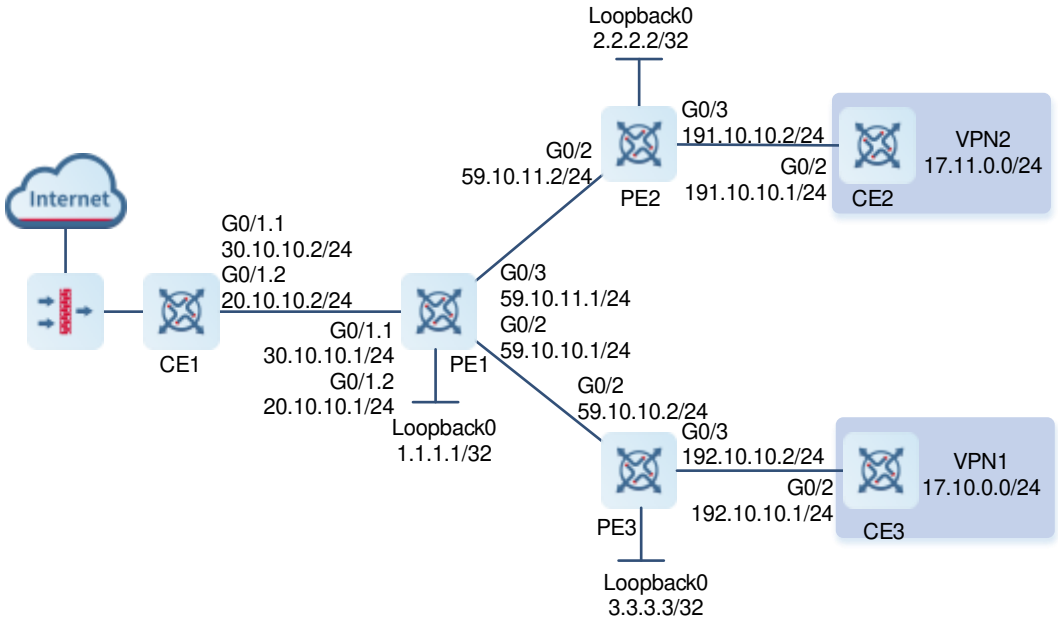
1. Requirements

VPNs cannot access each other, and these VPNs access the Internet through a unified device. VPN1 and VPN2 access the Internet through PE1. However, VPN1 and VPN2 cannot access each other.

When the centralized control mode is utilized and newly added VPN sites need to access the Internet through the unified egress, you only need to add filtering rules on the egress CE without changing configurations of other VPN sites. This ensures good extensibility. The disadvantage is that traffic to be isolated can be discarded only after reaching the egress CE, wasting network bandwidth.

2. Topology

Figure 1-22 Configuring Basic IPv4 MPLS L3VPN Functions (Unified Internet Access Egress and Centralized Control)



3. Notes

- On PE2, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Use OSPF to exchange routes with CE2, and establish IBGP neighbor relationship with PE1 to distribute IP routes. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On CE2, configure an IP address for the interface used to connect to PE2, and use OSPF to exchange routes with PE2.
- On PE3, configure a loopback interface, create VRF instance VPN2, set the export and import RT values of VPN2 to 1 and 200, define RD and RT values, and associate the VRF instance with the corresponding interface. Use OSPF to exchange routes with CE3, and establish IBGP neighbor relationship with PE1 to distribute IP routes. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On PE1, configure a loopback interface, create a trunk interface, create VRF instance vrf_out, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure a default route to the Internet, create VRF instance vrf_in, define RD and RT values, and associate the VRF instance with the corresponding interface. Use EBGP to exchange routes with CE1, establish IBGP neighbor relationship with PE2 and PE3, and configure the route exchange function for VRF instance vrf_out. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On CE1, create a Layer 3 sub-interface, configure the EBGP neighbor relationship with PE1, and create an ACL rule on Layer 3 sub-interface GigabitEthernet0/1.1.

Caution

For connection between PE1 and CE1, this example uses switch virtual interface (SVI) and 802.1Q sub-interface configurations, which are not supported by some devices. PE1 and CE1 can be connected through any two links (physical or logical links) only if two route adjacencies are formed between them. Users can select a suitable connection method based on actual requirements.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PEs and CEs are similar. The following shows how to configure OSPF neighbors on PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# router ospf 1
PE1(config-router)# network 59.10.11.0 0.0.0.255 area 0
PE1(config-router)# network 59.10.10.0 0.0.0.255 area 0
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1(config-router)# exit
```

(3) Configure basic MPLS functions.

Configurations on PEs are similar. The following shows how to configure basic MPLS functions on PE1.

```
PE1(config-router)# exit
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/3
PE1(config-if-GigabitEthernet 0/3)# ip address 59.10.11.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/3)# label-switching
PE1(config-if-GigabitEthernet 0/3)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/3)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# ip address 59.10.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# label-switching
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/2)# exit
```

(4) Create a VRF instance and an Ethernet sub-interface and associate them.

Configurations on PEs are similar. The following uses PE1 as an example.

```
PE1(config)# ip vrf vrf_in
PE1(config-vrf)# rd 1:400
PE1(config-vrf)# route-target import 1:100
PE1(config-vrf)# route-target import 1:200
PE1(config-vrf)# exit
PE1(config)# ip vrf vrf_out
```

```
PE1(config-vrf)# rd 1:300
PE1(config-vrf)# route-target export 1:100
PE1(config-vrf)# route-target export 1:200
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1.1
PE1(config-if-GigabitEthernet 0/1.1)# ip vrf forwarding vrf_in
PE1(config-if-GigabitEthernet 0/1.1)# ip address 30.10.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1.1)# exit
PE1(config)# interface gigabitethernet 0/1.2
PE1(config-if-GigabitEthernet 0/1.2)# ip vrf forwarding vrf_out
PE1(config-if-GigabitEthernet 0/1.2)# ip address 20.10.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1.1)# exit
```

(5) Configure BGP neighbors to advertise VPN routes.

PE1 establishes EBGP neighbor relationship with CE1.

```
PE1(config)# router bgp 1
PE1(config-router)# address-family ipv4 vrf vrf_in
PE1(config-router-af)# neighbor 30.10.10.2 remote-as 100
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

CE1 establishes EBGP neighbor relationship with PE1.

```
CE1(config)# router bgp 100
CE1(config-router)# neighbor 30.10.10.1 remote-as 1
CE1(config-router)# exit
```

PE1 establishes IBGP neighbor relationship with PE2 and PE3 to advertise VPN routes.

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 2.2.2.2 remote-as 1
PE1(config-router)# neighbor 2.2.2.2 update-source loopback 0
PE1(config-router)# neighbor 3.3.3.3 remote-as 1
PE1(config-router)# neighbor 3.3.3.3 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 2.2.2.2 activate
PE1(config-router-af)# neighbor 3.3.3.3 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf vrf_out
PE1(config-router-af)# default-information originate
PE1(config-router-af)# redistribute static
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

5. Verification

(1) After the configuration is completed, run the **show ip route** command to display existing routes.

PE1 verification result

```
PE1# show ip route vrf vrf_out
Routing Table: vrf_out
```

```
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is 20.10.10.2 to network 0.0.0.0
S*   0.0.0.0/0 [1/0] via 20.10.10.2, GigabitEthernet 0/1.2, 03:12:19
C    20.10.10.0/24 is directly connected, GigabitEthernet 0/1.1, 03:12:16
L    20.10.10.1/32 is directly connected, GigabitEthernet 0/1.2, 03:12:16

PE1# show ip route vrf vrf_in
Routing Table: vrf_in

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is 20.10.10.2 to network 0.0.0.0
B*   0.0.0.0/0 [20/0] via 20.10.10.2, 01:12:21
B    17.10.0.1/32 [200/1] via 3.3.3.3, 00:04:26
B    17.11.0.1/32 [200/1] via 2.2.2.2, 00:26:00
C    30.10.10.0/24 is directly connected, GigabitEthernet 0/1.1, 03:12:16
L    30.10.10.1/32 is directly connected, GigabitEthernet 0/1.1, 03:12:16
B    191.10.10.0/24 [200/1] via 2.2.2.2, 00:26:00
B    192.10.10.0/24 [200/1] via 3.3.3.3, 00:36:05
```

CE1 verification result

```
CE1# show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is 30.10.10.1 to network 0.0.0.0
B*   0.0.0.0/0 [20/0] via 30.10.10.1, 01:07:22
B    17.10.0.1/32 [20/0] via 30.10.10.1, 00:01:33
B    17.11.0.1/32 [20/0] via 30.10.10.1, 00:23:14
C    20.10.10.0/24 is directly connected, GigabitEthernet 0/1.2
```

```
C    20.10.10.2/32 is local host.
C    30.10.10.0/24 is directly connected, GigabitEthernet 0/1.1
C    30.10.10.2/32 is local host.
B    191.10.10.0/24 [20/0] via 30.10.10.1, 00:23:14
B    192.10.10.0/24 [20/0] via 30.10.10.1, 00:33:19
```

CE2 verification result

```
CE2# show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is 191.10.10.2 to network 0.0.0.0
O*E2 0.0.0.0/0 [110/1] via 191.10.10.2, 00:08:11, GigabitEthernet 0/2
C    17.11.0.0/24 is directly connected, Loopback 0
C    17.11.0.1/32 is local host.
C    191.10.10.0/24 is directly connected, GigabitEthernet 0/2
C    191.10.10.1/32 is local host.
```

CE3 verification result

```
CE3# show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is 192.10.10.2 to network 0.0.0.0
O*E2 0.0.0.0/0 [110/1] via 192.10.10.2, 00:31:48, GigabitEthernet 0/2
C    17.10.0.0/24 is directly connected, Loopback 0
C    17.10.0.1/32 is local host.
C    192.10.10.0/24 is directly connected, GigabitEthernet 0/2
C    192.10.10.1/32 is local host.
```

(2) After the configuration is completed, run the **ping** command to detect the connectivity between sites.

Verify that PE2 can ping PE1 and PE3.

Verify that CE2 can ping CE1 but cannot ping CE3.

Verify that PE3 can ping PE1 and PE2.

Verify that CE3 can ping CE1 but cannot ping CE2.

Verify that PE1 can ping PE2 and PE3.

Verify that CE1 can ping CE2 and CE3.

6. Configuration Files

CE1 configuration file

```
hostname CE1
!
ip access-list standard 1
 10 deny 17.0.0.0 0.255.255.255
 20 permit any
!
interface GigabitEthernet 0/1.1
 encapsulation dot1Q 10
 ip access-group 1 out
 ip address 30.10.10.2 255.255.255.0
!
interface GigabitEthernet 0/1.2
 encapsulation dot1Q 20
 ip address 20.10.10.2 255.255.255.0
!
router bgp 100
 neighbor 30.10.10.1 remote-as 1
 !
 address-family ipv4
  neighbor 30.10.10.1 activate
 exit-address-family
!
```

PE1 configuration file

```
hostname PE1
!
mpls enable
!
ip vrf vrf_in
 rd 1:400
 route-target import 1:200
 route-target import 1:100
!
ip vrf vrf_out
 rd 1:300
 route-target export 1:200
 route-target export 1:100
!
interface GigabitEthernet 0/1.1
 ip vrf forwarding vrf_in
 ip address 30.10.10.1 255.255.255.0
!
interface GigabitEthernet 0/1.2
 ip vrf forwarding vrf_out
```

```
 ip address 20.10.10.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 59.10.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 no switchport
 ip address 59.10.11.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 1
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 update-source Loopback 0
 neighbor 3.3.3.3 remote-as 1
 neighbor 3.3.3.3 update-source Loopback 0
 !
 address-family ipv4
  neighbor 2.2.2.2 activate
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 2.2.2.2 activate
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family ipv4 vrf vrf_out
  default-information originate
  redistribute static
 exit-address-family
 !
 address-family ipv4 vrf vrf_in
  neighbor 30.10.10.2 remote-as 100
  neighbor 30.10.10.2 activate
 exit-address-family
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 59.10.10.0 0.0.0.255 area 0
```

```
 network 59.10.11.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
ip route vrf vrf_out 0.0.0.0 0.0.0.0 GigabitEthernet 0/1.2 20.10.10.2
!
```

PE2 configuration file

```
hostname PE2
!
mpls enable
!
ip vrf VPN2
 rd 1:200
 route-target both 1:200
!
interface GigabitEthernet 0/2
 no switchport
 ip address 59.10.11.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 ip vrf forwarding VPN2
 ip address 191.10.10.2 255.255.255.0
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN2
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
```

```
 network 2.2.2.2 0.0.0.0 area 0
 network 59.10.11.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN2
 redistribute bgp subnets
 network 191.10.10.0 0.0.0.255 area 0
 default-information originate
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE3 configuration file

```
hostname PE3
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/2
 no switchport
 ip address 59.10.10.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 ip vrf forwarding VPN1
 ip address 192.10.10.2 255.255.255.0
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
```

```
 redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 59.10.10.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN2
 redistribute bgp subnets
 network 192.10.10.0 0.0.0.255 area 0
 default-information originate
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/2
 no switchport
 ip address 191.10.10.1 255.255.255.0
!
interface Loopback 0
 ip address 17.11.0.1 255.255.255.0
!
router ospf 1
 network 17.11.0.0 0.0.0.255 area 0
 network 191.10.10.0 0.0.0.255 area 0
!
```

CE3 configuration file

```
hostname CE3
!
interface GigabitEthernet 0/2
 no switchport
 ip address 192.10.10.1 255.255.255.0
!
interface Loopback 0
 ip address 17.10.0.1 255.255.255.0
!
router ospf 1
 network 17.10.0.0 0.0.0.255 area 0
 network 192.10.10.0 0.0.0.255 area 0
!
```

7. Common Errors

The router ID is not 32 bits. As a result, the LDP session or BGP neighbor relationship fails to be established.

1.16.5 Configuring Basic IPv4 MPLS L3VPN Functions (Unified Internet Access Egress and Distributed Control)

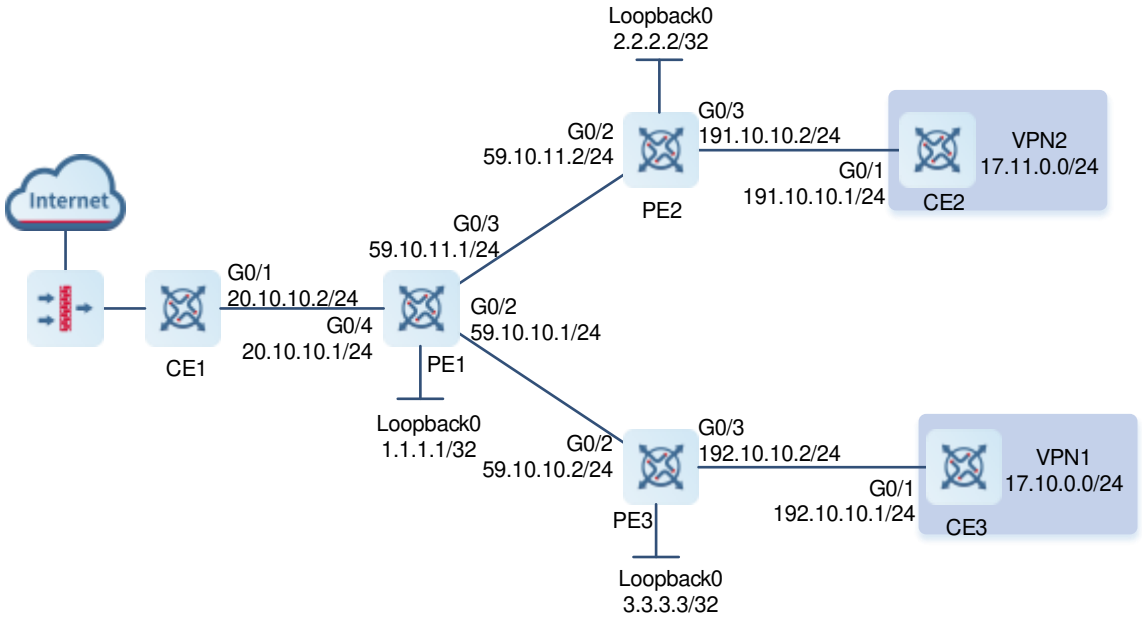
1. Requirements

VPNs cannot access each other, and these VPNs access the Internet through a unified device. VPN1 and VPN2 access the Internet through PE1. However, VPN1 and VPN2 cannot access each other.

When the distributed control mode is utilized and newly added VPN sites need to access the Internet through the unified egress, you need to add filtering rules for CEs in each VPN site that accesses the Internet through the unified egress. This results in poor extensibility. The advantage is that traffic to be isolated can be discarded at CEs of VPN sites, saving network bandwidth.

2. Topology

Figure 1-23 Configuring Basic IPv4 MPLS L3VPN Functions (Unified Internet Access Egress and Distributed Control)



3. Notes

- On PE1, configure a loopback interface, create VRF instance vrf_net, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure a default route to the Internet, use EBGP to exchange routes with CE1, establish IBGP neighbor relationship with PE2 and PE3, and configure the route exchange function of VRF instance vrf_net. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On CE1, establish EBGP neighbor relationship with PE1.
- On PE2, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Use OSPF to exchange routes with CE2, and establish IBGP neighbor relationship with PE1 to distribute IP routes. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.

- On CE2, create ACL rules and apply the ACL rules to Layer 3 sub-interfaces. Configure route exchange with PE2 and configure a default static route.
- On PE3, configure a loopback interface, create VRF instance VPN2, set the export and import RT values of VPN2 to 1 and 200, define RD and RT values, and associate the VRF instance with the corresponding interface. Use OSPF to exchange routes with CE3, and establish IBGP neighbor relationship with PE1 to distribute IP routes. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On CE3, create ACL rules and apply the ACL rules to Layer 3 sub-interfaces. Configure route exchange with PE3 and configure a default static route.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PEs and CEs are similar. The following shows how to configure OSPF neighbors on PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# router ospf 1
PE1(config-router)# network 59.10.11.0 0.0.0.255 area 0
PE1(config-router)# network 59.10.10.0 0.0.0.255 area 0
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
```

(3) Configure basic MPLS functions.

Configurations on PEs are similar. The following shows how to configure basic MPLS functions on PE1.

```
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/3
PE1(config-if-GigabitEthernet 0/3)# ip address 59.10.11.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/3)# label-switching
PE1(config-if-GigabitEthernet 0/3)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/3)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# ip address 59.10.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# label-switching
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/2)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following uses PE1 as an example.

```
PE1(config)# ip vrf vrf_net
PE1(config-vrf)# rd 1:300
PE1(config-vrf)# route-target import 1:100
PE1(config-vrf)# route-target import 1:200
PE1(config-vrf)# route-target export 1:100
```

```
PE1(config-vrf)# route-target export 1:200
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/4
PE1(config-if-GigabitEthernet 0/4)# no switchport
PE1(config-if-GigabitEthernet 0/4)# ip vrf forwarding vrf_net
PE1(config-if-GigabitEthernet 0/4)# ip address 20.10.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/4)# exit
```

(5) Configure BGP neighbors to advertise VPN routes.

PE1 establishes EBGP neighbor relationship with CE1 to advertise VPN routes.

```
PE1(config)# router bgp 1
PE1(config-router)# address-family ipv4 vrf vrf_net
PE1(config-router-af)# neighbor 20.10.10.2 remote-as 100
PE1(config-router-af)# default-information originate
PE1(config-router-af)# redistribute static
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

CE1 establishes EBGP neighbor relationship with PE1.

```
CE1(config)# router bgp 100
CE1(config-router)# neighbor 20.10.10.1 remote-as 1
CE1(config-router)# exit
```

PE1 establishes IBGP neighbor relationship with PE2 and PE3.

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 2.2.2.2 remote-as 1
PE1(config-router)# neighbor 2.2.2.2 update-source loopback 0
PE1(config-router)# neighbor 3.3.3.3 remote-as 1
PE1(config-router)# neighbor 3.3.3.3 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 2.2.2.2 activate
PE1(config-router-af)# neighbor 3.3.3.3 activate
```

Configure CE1.

```
CE1> enable
CE1(config)# interface gigabitethernet 0/1
CE1(config-if-GigabitEthernet 0/1)# ip address 20.10.10.2 255.255.255.0
CE1(config-if-GigabitEthernet 0/1)# exit
CE1(config)# router bgp 100
CE1(config-router)# neighbor 20.10.10.1 remote-as 1
```

Configure PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# interface loopback 0
PE1(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
PE1(config-if-Loopback 0)# exit
PE1(config)# ip vrf vrf_net
```

```
PE1(config-vrf)# rd 1:300
PE1(config-vrf)# route-target import 1:100
PE1(config-vrf)# route-target import 1:200
PE1(config-vrf)# route-target export 1:100
PE1(config-vrf)# route-target export 1:200
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/4
PE1(config-if-GigabitEthernet 0/4)# no switchport
PE1(config-if-GigabitEthernet 0/4)# ip vrf forwarding vrf_net
PE1(config-if-GigabitEthernet 0/4)# ip address 20.10.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/4)# exit
PE1(config)#ip route vrf vrf_net 0.0.0.0 0.0.0.0 GigabitEthernet 0/4
20.10.10.2
PE1(config)# router bgp 1
PE1(config-router)# address-family ipv4 vrf vrf_net
PE1(config-router-af)# neighbor 20.10.10.2 remote-as 100
PE1(config-router-af)# exit-address-family
PE1(config-router)# neighbor 2.2.2.2 remote-as 1
PE1(config-router)# neighbor 2.2.2.2 update-source loopback 0
PE1(config-router)# neighbor 3.3.3.3 remote-as 1
PE1(config-router)# neighbor 3.3.3.3 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 2.2.2.2 activate
PE1(config-router-af)# neighbor 3.3.3.3 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf vrf_net
PE1(config-router-af)# default-information originate
PE1(config-router-af)# redistribute static
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/3
PE1(config-if-GigabitEthernet 0/3)# ip address 59.10.11.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/3)# label-switching
PE1(config-if-GigabitEthernet 0/3)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/3)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# ip address 59.10.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# label-switching
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/2)# exit
PE1(config)# router ospf 1
PE1(config-router)# network 59.10.11.0 0.0.0.255 area 0
```

```
PE1(config-router)# network 59.10.10.0 0.0.0.255 area 0
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
```

Configure PE2.

```
PE2> enable
PE2# configure terminal
PE2(config)# interface loopback 0
PE2(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
PE2(config-if-Loopback 0)# exit
PE2(config)# ip vrf VPN1
PE2(config-vrf)# rd 1:100
PE2(config-vrf)# route-target both 1:100
PE2(config-vrf)# exit
PE2(config)# interface gigabitethernet 0/3
PE2(config-if-GigabitEthernet 0/3)# ip vrf forwarding VPN1
PE2(config-if-GigabitEthernet 0/3)# ip address 191.10.10.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/3)# exit
PE2(config)# router ospf 10 vrf VPN1
PE2(config-router)# network 191.10.10.0 0.0.0.255 area 0
PE2(config-router)# default-information originate
PE2(config-router)# redistribute bgp subnets
PE2(config-router)# exit
PE2(config)# router bgp 1
PE2(config-router)# neighbor 1.1.1.1 remote-as 1
PE2(config-router)# neighbor 1.1.1.1 update-source loopback 0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 1.1.1.1 activate
PE2(config-router-af)# exit-address-family
PE2(config-router)# address-family ipv4 vrf VPN1
PE2(config-router-af)# redistribute ospf 10
PE2(config-router-af)# exit-address-family
PE2(config-router)# exit
PE2(config)# mpls enable
PE2(config)# mpls router ldp
PE2(config-mpls-router)# ldp router-id interface loopback 0 force
PE2(config-mpls-router)# exit
PE2(config)# interface gigabitethernet 0/2
PE2(config-if-GigabitEthernet 0/2)# ip address 59.10.11.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/2)# label-switching
PE2(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE2(config-if-GigabitEthernet 0/2)# exit
PE2(config)# router ospf 1
PE2(config-router)# network 59.10.11.0 0.0.0.255 area 0
PE2(config-router)# network 2.2.2.2 0.0.0.0 area 0
```

Configure PE3.

```
PE3> enable
```

```
PE3# configure terminal
PE3(config)# interface loopback 0
PE3(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
PE3(config-if-Loopback 0)# exit
PE3(config)# ip vrf VPN1
PE3(config-vrf)# rd 1:200
PE3(config-vrf)# route-target both 1:200
PE3(config-vrf)# exit
PE3(config)# interface gigabitethernet 0/3
PE3(config-if-GigabitEthernet 0/3)# ip vrf forwarding VPN1
PE3(config-if-GigabitEthernet 0/3)# ip address 192.10.10.2 255.255.255.0
PE3(config-if-GigabitEthernet 0/3)# exit
PE3(config)# router ospf 10 vrf VPN1
PE3(config-router)# network 192.10.10.0 0.0.0.255 area 0
PE3(config-router)# default-information originate
PE3(config-router)# redistribute bgp subnets
PE3(config-router)# exit
PE3(config)# router bgp 1
PE3(config-router)# neighbor 1.1.1.1 remote-as 1
PE3(config-router)# neighbor 1.1.1.1 update-source loopback 0
PE3(config-router)# address-family vpnv4
PE3(config-router-af)# neighbor 1.1.1.1 activate
PE3(config-router-af)# exit-address-family
PE3(config-router)# address-family ipv4 vrf VPN1
PE3(config-router-af)# redistribute ospf 10
PE3(config-router-af)# exit-address-family
PE3(config-router)# exit
PE3(config)# mpls enable
PE3(config)# mpls router ldp
PE3(config-mpls-router)# ldp router-id interface loopback 0 force
PE3(config-mpls-router)# exit
PE3(config)# interface gigabitethernet 0/2
PE3(config-if-GigabitEthernet 0/2)# ip address 59.10.10.2 255.255.255.0
PE3(config-if-GigabitEthernet 0/2)# label-switching
PE3(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE3(config-if-GigabitEthernet 0/2)# exit
PE3(config)# router ospf 1
PE3(config-router)# network 59.10.10.0 0.0.0.255 area 0
PE3(config-router)# network 3.3.3.3 0.0.0.0 area 0
```

Configure CE2.

```
CE2> enable
CE2# configure terminal
CE2(config)# access-list 2000 deny ip any 17.11.0.0 0.0.255.255
CE2(config)# access-list 2000 permit ip any any
CE2(config)# interface gigabitethernet 0/1
CE2(config-if-GigabitEthernet 0/1)# ip access-group 2000 out
```

```
CE2(config-if-GigabitEthernet 0/1)# exit
CE2(config)# router ospf 1
CE2(config-router)# network 191.10.10.0 0.0.0.255 area 0
CE2(config-router)# network 17.11.0.0 0.0.0.255 area 0
```

CE3 configuration

```
CE3> enable
CE3# configure terminal
CE3(config)# access-list 2000 deny ip any 17.10.0.0 0.0.255.255
CE3(config)# access-list 2000 permit ip any any
CE3(config)# interface gigabitethernet 0/1
CE3(config-if-GigabitEthernet 0/1)# ip access-group 2000 out
CE3(config-if-GigabitEthernet 0/1)# exit
CE3(config)# router ospf 1
CE3(config-router)# network 192.10.10.0 0.0.0.255 area 0
CE3(config-router)# network 17.10.0.0 0.0.0.255 area 0
```

5. Verification

(1) After the configuration is completed, run the **show ip route** command to display existing routes.

CE1 verification result

```
CE1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is 20.10.10.1 to network 0.0.0.0
B*   0.0.0.0/0 [20/0] via 20.10.10.1, 00:01:14
B    17.10.0.1/32 [20/0] via 20.10.10.1, 00:01:46
B    17.11.0.1/32 [20/0] via 20.10.10.1, 00:02:10
C    20.10.10.0/24 is directly connected, GigabitEthernet 0/1, 00:01:40
L    20.10.10.2/32 is directly connected, GigabitEthernet 0/1, 00:01:40
B    191.10.10.0/24 [20/0] via 20.10.10.1, 00:02:10
B    192.10.10.0/24 [20/0] via 20.10.10.1, 00:01:46
```

PE1 verification result

```
PE1# show ip route vrf vrf_net
Routing Table: vrf_net

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is 20.10.10.2 to network 0.0.0.0
S*   0.0.0.0/0 [1/0] via 20.10.10.2, GigabitEthernet 0/4, 00:01:42
B    17.10.0.1/32 [200/1] via 3.3.3.3, 00:01:42
B    17.11.0.1/32 [200/1] via 2.2.2.2, 00:01:47
C    20.10.10.0/24 is directly connected, GigabitEthernet 0/4, 00:01:42
L    20.10.10.1/32 is directly connected, GigabitEthernet 0/4, 00:01:42
B    191.10.10.0/24 [200/1] via 2.2.2.2, 00:01:47
B    192.10.10.0/24 [200/1] via 3.3.3.3, 00:01:42

PE1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    1.1.1.1/32 is directly connected, Loopback 0, 00:01:42
O    2.2.2.2/32 [110/1] via 59.10.11.2, GigabitEthernet 0/3, 03:59:18
O    3.3.3.3/32 [110/1] via 59.10.10.2, GigabitEthernet 0/2, 03:46:02
C    59.10.10.0/24 is directly connected, GigabitEthernet 0/3, 00:01:42
L    59.10.10.1/32 is directly connected, GigabitEthernet 0/3, 00:01:42
C    59.10.11.0/24 is directly connected, GigabitEthernet 0/2, 00:01:42
L    59.10.11.1/32 is directly connected, GigabitEthernet 0/2, 00:01:42
```

(2) After the configuration is completed, run the **ping** command to detect the connectivity between sites.

Verify that PE1 can ping PE2 and PE3.

Verify that CE1 can ping CE2 and CE3.

Verify that PE2 can ping PE1 and PE3.

Verify that CE2 can ping CE1 but cannot ping CE3.

Verify that PE3 can ping PE1 and PE2.

Verify that CE3 can ping CE1 but cannot ping CE2.

6. Configuration Files

CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 20.10.10.2 255.255.255.0
!
router bgp 100
 neighbor 20.10.10.1 remote-as 1
 !
 address-family ipv4
  neighbor 20.10.10.1 activate
 exit-address-family
!
```

PE1 configuration file

```
hostname PE1
!
mpls enable
!
ip vrf vrf_net
 rd 1:300
 route-target both 1:100
 route-target both 1:200
!
interface GigabitEthernet 0/2
 no switchport
 ip address 59.10.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 no switchport
 ip address 59.10.11.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/4
 no switchport
 ip vrf forwarding vrf_net
 ip address 20.10.10.1 255.255.255.0
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 1
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 update-source Loopback 0
```

```
 neighbor 3.3.3.3 remote-as 1
 neighbor 3.3.3.3 update-source Loopback 0
 !
 address-family ipv4
  neighbor 2.2.2.2 activate
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 2.2.2.2 activate
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family ipv4 vrf vrf_net
  default-information originate
  redistribute static
  neighbor 20.10.10.2 remote-as 100
  neighbor 20.10.10.2 activate
 exit-address-family
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 59.10.10.0 0.0.0.255 area 0
 network 59.10.11.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
ip route vrf vrf_net 0.0.0.0 0.0.0.0 GigabitEthernet 0/4 20.10.10.2
!
```

PE2 configuration file

```
hostname PE2
!
mpls enable
!
ip vrf VPN2
 rd 1:200
 route-target both 1:200
!
interface GigabitEthernet 0/2
 no switchport
 ip address 59.10.11.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
```

```
 ip vrf forwarding VPN2
 ip address 191.10.10.2 255.255.255.0
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN2
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 59.10.11.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN2
 redistribute bgp subnets
 network 191.10.10.0 0.0.0.255 area 0
 default-information originate
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE3 configuration file

```
hostname PE3
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/2
 no switchport
 ip address 59.10.10.2 255.255.255.0
```

```
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 ip vrf forwarding VPN1
 ip address 192.10.10.2 255.255.255.0
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 59.10.10.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN2
 redistribute bgp subnets
 network 192.10.10.0 0.0.0.255 area 0
 default-information originate
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE2 configuration file

```
hostname CE2
!
access-list 2000 deny ip any 17.11.0.0 0.0.255.255
access-list 2000 permit ip any any
!
interface GigabitEthernet 0/1
 no switchport
```

```
 ip access-group 2000 out
 ip address 191.10.10.1 255.255.255.0
!
interface Loopback 0
 ip address 17.11.0.1 255.255.255.0
!
router ospf 1
 network 17.11.0.0 0.0.0.255 area 0
 network 191.10.10.0 0.0.0.255 area 0
!
```

CE3 configuration file

```
hostname CE3
!
access-list 2000 deny ip any 17.10.0.0 0.0.255.255
access-list 2000 permit ip any any
!
interface GigabitEthernet 0/1
 no switchport
 ip access-group 2000 out
 ip address 192.10.10.1 255.255.255.0
!
interface Loopback 0
 ip address 17.10.0.1 255.255.255.0
!
router ospf 1
 network 17.10.0.0 0.0.0.255 area 0
 network 192.10.10.0 0.0.0.255 area 0
!
```

7. Common Errors

The router ID is not 32 bits. As a result, the LDP session or BGP neighbor relationship fails to be established.

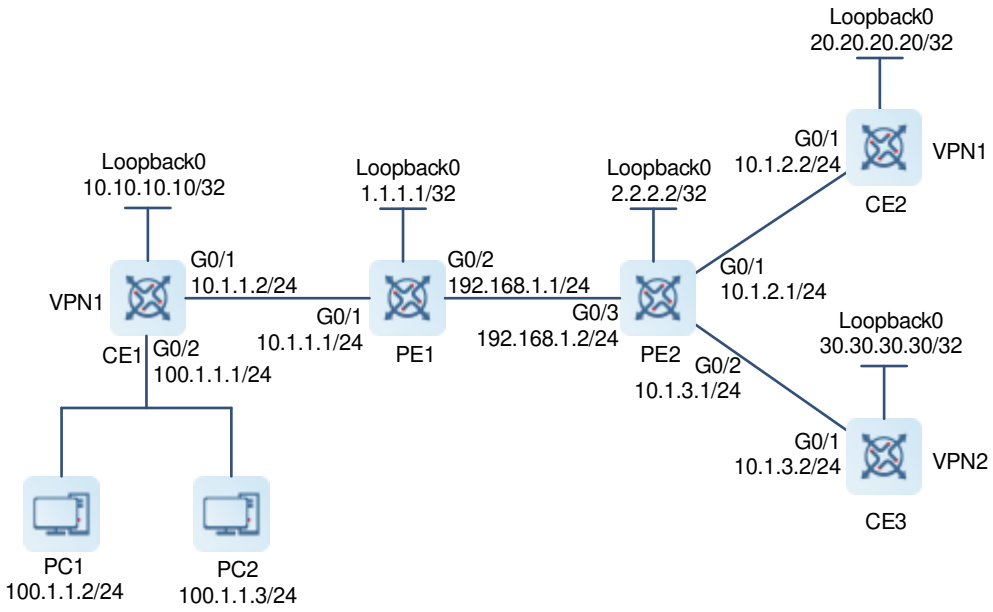
1.16.6 Configuring Basic IPv4 MPLS L3VPN Functions (Multi-Role Host)

1. Requirements

Generally, hosts in a VPN site can access other hosts in the same VPN site only. However, some hosts may need to access multiple VPNs, and these hosts are called multi-role hosts. As shown in Figure 1-24, PC1 and PC2 are hosts at a site of VPN1, and PC2 is a multi-role host and needs to access hosts in VPN1 and VPN2 sites. PC1 can access hosts only in VPN1 sites.

2. Topology

Figure 1-24 Configuring Basic IPv4 MPLS L3VPN Functions (Multi-Role Host)



3. Notes

- Configure an MPLS backbone network.
- Configure an MPLS L3VPN.
- Configure VRF instance VPN_MR for a multi-role host to access VPNs (VPN1 and VPN2) and advertise routes of the multi-role host to other VPNs.
- Configure a policy-based routing (PBR) route and redirect access packets of the multi-role host to VRF instance VPN_MR.
- Configure a default route on CE1.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on all devices are similar. The following shows how to configure OSPF neighbors on PE1.

```
PE1(config)# router ospf 1
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1(config-router)# network 192.168.1.0 0.0.0.255 area 0
PE1(config-router)# exit
```

(3) Configure basic MPLS functions.

Configurations on PEs are similar. The following shows how to configure basic MPLS functions on PE1.

```
PE1(config)# mpls enable
PE1(config)# interface gigabitethernet 0/2
```

```
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/2)# label-switching
PE1(config-if-GigabitEthernet 0/2)# exit
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following uses PE1 as an example.

```
PE1(config)# ip vrf VPN1
PE1(config-vrf)# rd 100:1
PE1(config-vrf)# route-target both 100:1
PE1(config-vrf)# exit
PE1(config)# ip vrf VPN_MR
PE1(config-vrf)# rd 200:1
PE1(config-vrf)# route-target export 100:2
PE1(config-vrf)# route-target import 100:1
PE1(config-vrf)# route-target import 100:2
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPN1
PE1(config-if-GigabitEthernet 0/1)# ip address 10.1.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# exit
```

(5) Configure VPN routes.

Configurations on PE1 and PE2 are similar. The following shows how to configure VPN routes on PE1.

```
PE1(config)# router ospf 10 vrf VPN1
PE1(config-router)# network 10.1.1.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# exit
PE1(config)# ip route vrf VPN_MR 100.1.1.3 255.255.255.255 gigabitethernet 0/1
```

(6) Establish IBGP neighbor relationship and advertise VPN routes.

Configurations on PE1 and PE2 are similar. The following shows how to configure IBGP neighbors on PE1.

```
PE1(config)# router bgp 100
PE1(config-router)# neighbor 2.2.2.2 remote-as 100
PE1(config-router)# neighbor 2.2.2.2 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 2.2.2.2 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPN1
PE1(config-router-af)# redistribute ospf 10
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPN_MR
PE1(config-router-af)# redistribute static
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

(7) Configure PRB routes.

```
Configure PBR routes on PE1. PE1(config)# ip access-list standard VRF_ACL
PE1(config-std-nacl)# permit host 100.1.1.3
PE1(config-std-nacl)# deny any
PE1(config-std-nacl)# exit
PE1(config)# route-map VRF_MAP permit 10
PE1(config-route-map)# match ip address VRF_ACL
PE1(config-route-map)# set vrf VPN_MR
PE1(config-route-map)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip policy route-map VRF_MAP
```

5. Verification

(1) After the configuration is completed, run the **show ip route** command to display VPN routes.

CE1 verification result

```
CE1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is 10.1.1.1 to network 0.0.0.0
S*   0.0.0.0/0 [1/0] via 10.1.1.1, GigabitEthernet 0/1, 00:08:59
C    10.1.1.0/24 is directly connected, GigabitEthernet 0/1, 00:08:59
L    10.1.1.2/32 is directly connected, GigabitEthernet 0/1, 00:08:59
O IA 10.1.2.0/24 [110/2] via 10.1.1.1, GigabitEthernet 0/1, 00:08:59
L    10.10.10.10/32 is directly connected, Loopback 0, 00:08:59
O IA 20.20.20.20/32 [110/2] via 10.1.1.1, GigabitEthernet 0/1, 00:08:59
C    100.1.1.0/24 is directly connected, GigabitEthernet 0/2, 00:08:59
L    100.1.1.1/32 is directly connected, GigabitEthernet 0/2, 00:08:59
```

PE1 verification result

```
PE1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
```

```
       * - candidate default

Gateway of last resort is no set
C    1.1.1.1/32 is directly connected, Loopback 0, 00:59:23
O    2.2.2.2/32 [110/1] via 192.168.1.2, GigabitEthernet 0/2, 00:59:23
C    192.168.1.0/24 is directly connected, GigabitEthernet 0/2, 00:59:23
L    192.168.1.1/32 is directly connected, GigabitEthernet 0/2, 00:59:23

PE1# show ip route vrf VPN1
Routing Table: VPN1

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    10.1.1.0/24 is directly connected, GigabitEthernet 0/1, 00:03:04
L    10.1.1.1/32 is directly connected, GigabitEthernet 0/1, 00:03:04
B    10.1.2.0/24 [200/1] via 2.2.2.2, 00:43:04
O    10.10.10.10/32 [110/1] via 10.1.1.2, GigabitEthernet 0/1, 00:07:53
B    20.20.20.20/32 [200/1] via 2.2.2.2, 00:43:04
O    100.1.1.0/24 [110/2] via 10.1.1.2, GigabitEthernet 0/1, 00:06:04

PE1# show ip route vrf VPN_MR
Routing Table: VPN_MR

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
B    10.1.1.0/24 [20/1] via 0.0.0.0, GigabitEthernet 0/1, 00:09:02
B    10.1.2.0/24 [200/1] via 2.2.2.2, 00:38:34
B    10.1.3.0/24 [200/1] via 2.2.2.2, 00:38:34
B    10.10.10.10/32 [20/1] via 10.1.1.2, 00:08:05
B    20.20.20.20/32 [200/1] via 2.2.2.2, 00:38:34
```

```
B    30.30.30.30/32 [200/1] via 2.2.2.2, 00:38:34
B    100.1.1.0/24 [20/2] via 10.1.1.2, 00:06:16
S    100.1.1.3/32 [1/0] via 10.1.1.2, GigabitEthernet 0/1, 00:06:16
```

PE2 verification result

```
PE2# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
O    1.1.1.1/32 [110/1] via 192.168.1.1, GigabitEthernet 0/3, 01:05:36
C    2.2.2.2/32 is directly connected, Loopback 0, 00:06:16
C    192.168.1.0/24 is directly connected, GigabitEthernet 0/3, 00:06:16
L    192.168.1.2/32 is directly connected, GigabitEthernet 0/3, 00:06:16

PE2# show ip route vrf VPN1
Routing Table: VPN1

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
B    10.1.1.0/24 [200/1] via 1.1.1.1, 00:09:25
C    10.1.2.0/24 is directly connected, GigabitEthernet 0/1, 00:08:28
L    10.1.2.1/32 is directly connected, GigabitEthernet 0/1, 00:08:28
B    10.10.10.10/32 [200/1] via 1.1.1.1, 00:08:28
O    20.20.20.20/32 [110/1] via 10.1.2.2, GigabitEthernet 0/1, 00:51:18
B    100.1.1.0/24 [200/2] via 1.1.1.1, 00:06:39

PE2# show ip route vrf VPN2
Routing Table: VPN2

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
```

```
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    10.1.3.0/24 is directly connected, GigabitEthernet 0/2, 00:50:57
L    10.1.3.1/32 is directly connected, GigabitEthernet 0/2, 00:50:57
O    30.30.30.30/32 [110/1] via 10.1.3.2, GigabitEthernet 0/2, 00:50:57
B    100.1.1.3/32 [200/0] via 1.1.1.1, 00:09:28
```

(2) After the configuration is completed, run the **ping** command on PC1 and PC2 to detect the connectivity with other sites.

P1 can ping 20.20.20.20 but cannot ping 30.30.30.30.

```
PC1# ping 20.20.20.20
Sending 5, 100-byte ICMP Echoes to 20.20.20.20, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/8/10 ms
PC1# ping 30.30.30.30
Sending 5, 100-byte ICMP Echoes to 30.30.30.30, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)
```

PC2 can ping both 20.20.20.20 and 30.30.30.30.

```
PC2# ping 20.20.20.20
Sending 5, 100-byte ICMP Echoes to 20.20.20.20, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/8/10 ms
PC2#ping 30.30.30.30
Sending 5, 100-byte ICMP Echoes to 30.30.30.30, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/10/10 ms
```

6. Configuration Files

CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.1.2 255.255.255.0
!
```

```
interface GigabitEthernet 0/2
 no switchport
 ip address 100.1.1.1 255.255.255.0
!
interface Loopback 0
 ip address 10.10.10.10 255.255.255.255
!
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
 network 10.10.10.10 0.0.0.0 area 0
 network 100.1.1.0 0.0.0.255 area 0
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet 0/1 10.1.1.1
!
```

PE1 configuration file

```
hostname PE1
!
mpls enable
!
route-map VRF_MAP permit 10
 match ip address VRF_ACL
 set vrf VPN_MR
!
ip vrf VPN1
 rd 100:1
 route-target both 100:1
!
ip vrf VPN_MR
 rd 200:1
 route-target both 100:2
 route-target import 100:1
!
ip access-list standard VRF_ACL
 10 permit host 100.1.1.3
 20 deny any
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip policy route-map VRF_MAP
 ip address 10.1.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 192.168.1.1 255.255.255.0
 label-switching
```

```
 mpls ldp enable
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 100
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback 0
 !
 address-family ipv4
  neighbor 2.2.2.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 2.2.2.2 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute ospf 10
 exit-address-family
 !
 address-family ipv4 vrf VPN_MR
  redistribute static
 exit-address-family
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN1
 redistribute bgp subnets
 network 10.1.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
ip route vrf VPN_MR 100.1.1.3 255.255.255.255 GigabitEthernet 0/1 10.1.1.2
!
```

PE2 configuration file

```
hostname PE2
!
mpls enable
!
ip vrf VPN1
 rd 101:1
 route-target both 100:1
```

```
!
ip vrf VPN2
 rd 101:2
 route-target both 100:2
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 10.1.2.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPN2
 ip address 10.1.3.1 255.255.255.0
!
interface GigabitEthernet 0/3
 no switchport
 ip address 192.168.1.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router bgp 100
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute ospf 10 match internal
 exit-address-family
 !
 address-family ipv4 vrf VPN2
  redistribute ospf 20 match internal
 exit-address-family
!
router ospf 1
 router-id 2.2.2.2
```

```
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN1
 redistribute bgp subnets
 network 10.1.2.0 0.0.0.255 area 0
!
router ospf 20 vrf VPN2
 redistribute bgp subnets
 network 10.1.3.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.2.2 255.255.255.0
!
interface Loopback 0
 ip address 20.20.20.20 255.255.255.255
!
router ospf 1
 network 10.1.2.0 0.0.0.255 area 0
 network 20.20.20.20 0.0.0.0 area 0
!
```

CE3 configuration file

```
hostname CE3
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.3.2 255.255.255.0
!
interface Loopback 0
 ip address 30.30.30.30 255.255.255.255
!
router ospf 1
 network 10.1.3.0 0.0.0.255 area 0
 network 30.30.30.30 0.0.0.0 area 0
!
```

7. Common Errors

The router ID is not 32 bits. As a result, the LDP session or BGP neighbor relationship fails to be established.

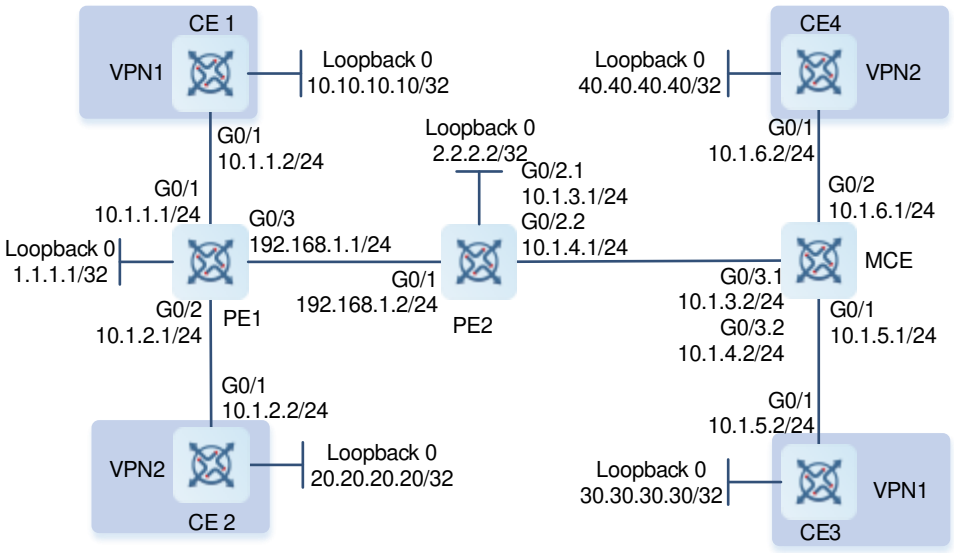
1.16.7 Configuring Basic IPv4 MPLS L3VPN Functions (MCE-based Hierarchical VPNs)

1. Requirements

Use the MCE networking method to change the original MPLS L3VPN network to a hierarchal network. The MCE saves only routes of the connected VPN site and default routes distributed by PE2 instead of routes of all VPN sites. Therefore, the capacity and performance requirements on the MCE are low. However, PE1 and PE2 need to save all VPN routes. Therefore, the capacity and performance requirements on PE1 and PE2 are high.

2. Topology

Figure 1-25 Configuring Basic IPv4 MPLS L3VPN Functions (MCE-based Hierarchal VPNs)



3. Notes

- Configure the MPLS network: configure interface addresses and OSPF on PE1 and PE2 and configure the MPLS function.
- Configure the MPLS L3VPN: configure access from CEs to PEs on PE1, PE2, CE1, and CE2 and configure MP-IBGP on PE1 and PE2.
- Configure the MCE and connected CEs.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on all devices are similar. The following shows how to configure OSPF neighbors on PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# router ospf 1
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1(config-router)# network 192.168.1.0 0.0.0.255 area 0
PE1(config-router)# exit
```

(3) Configure basic MPLS functions.

Configurations on PEs are similar. The following shows how to configure basic MPLS functions on PE1.

```
PE1(config)# mpls enable
PE1(config)# interface gigabitethernet 0/3
PE1(config-if-GigabitEthernet 0/3)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/3)# label-switching
PE1(config-if-GigabitEthernet 0/3)# exit
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following uses PE1 as an example.

```
PE1(config)# ip vrf VPN1
PE1(config-vrf)# rd 100:1
PE1(config-vrf)# route-target both 100:1
PE1(config-vrf)# exit
PE1(config)# ip vrf VPN2
PE1(config-vrf)# rd 100:2
PE1(config-vrf)# route-target both 100:2
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPN1
PE1(config-if-GigabitEthernet 0/1)# ip address 10.1.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# ip vrf forwarding VPN2
PE1(config-if-GigabitEthernet 0/2)# ip address 10.1.2.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# exit
```

(5) Configure VPN routes.

Configurations on PEs and MCEs are similar. The following shows how to configure VPN routes on PE1.

```
PE1(config)# router ospf 10 vrf VPN1
PE1(config-router)# network 10.1.1.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# exit
PE1(config)# router ospf 20 vrf VPN2
PE1(config-router)# network 10.1.2.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# exit
```

(6) Configure BGP neighbors to advertise VPN routes.

Configurations on PE1 and PE2 are similar. The following shows how to configure IBGP neighbors on PE1.

```
PE1(config)# router bgp 100
PE1(config-router)# neighbor 2.2.2.2 remote-as 100
PE1(config-router)# neighbor 2.2.2.2 update-source loopback 0
```

```
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 2.2.2.2 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPN1
PE1(config-router-af)# redistribute ospf 10
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPN2
PE1(config-router-af)# redistribute ospf 20
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

(7) Configure a default route.

Configure a default route on PE2.

```
PE2(config)# ip route vrf VPN1 0.0.0.0 0.0.0.0 null 0
PE2(config)# ip route vrf VPN2 0.0.0.0 0.0.0.0 null 0
```

5. Verification

(1) After the configuration is completed, run the **show ip route** command to display VPN routes.

MCE verification result

```
MCE# show ip route vrf VPN1
Routing Table: VPN1

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is 10.1.3.1 to network 0.0.0.0
O*E2 0.0.0.0/0 [110/1] via 10.1.3.1, GigabitEthernet 0/3.1, 00:38:35
C    10.1.3.0/24 is directly connected, GigabitEthernet 0/3.1, 00:38:35
L    10.1.3.2/32 is directly connected, GigabitEthernet 0/3.1, 00:38:35
C    10.1.5.0/24 is directly connected, GigabitEthernet 0/1, 00:38:35
L    10.1.5.0/42 is directly connected, GigabitEthernet 0/1, 00:38:35
O    30.30.30.30/32 [110/1] via 10.1.5.2, GigabitEthernet 0/1, 00:40:05

MCE# show ip route vrf VPN2
Routing Table: VPN2

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is 10.1.4.1 to network 0.0.0.0
O*E2 0.0.0.0/0 [110/1] via 10.1.4.1, 00:38:35, GigabitEthernet 0/3.2
C    10.1.4.0/24 is directly connected, GigabitEthernet 0/3.2, 00:38:35
L    10.1.4.2/32 is directly connected, GigabitEthernet 0/3.2, 00:38:35
C    10.1.6.0/24 is directly connected, GigabitEthernet 0/2, 00:38:35
L    10.1.6.0/42 is directly connected, GigabitEthernet 0/2, 00:38:35
O    40.40.40.40/32 [110/1] via 10.1.6.2, GigabitEthernet 0/2, 00:40:07
```

(2) Devices in the same VPN site can ping each other, and devices in different VPN sites cannot ping each other.

CE3 verification result

```
CE3# ping 10.10.10.10 source 30.30.30.30
Sending 5, 100-byte ICMP Echoes to 10.10.10.10, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/20/20 ms

CE3#ping 20.20.20.20 source 30.30.30.30
Sending 5, 100-byte ICMP Echoes to 20.20.20.20, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)
```

CE4 verification result

```
CE4# ping 20.20.20.20 source 40.40.40.40
Sending 5, 100-byte ICMP Echoes to 20.20.20.20, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/20/20 ms

CE3#ping 10.10.10.10 source 40.40.40.40
Sending 5, 100-byte ICMP Echoes to 10.10.10.10, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)
```

6. Configuration Files

CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.1.2 255.255.255.0
```

```
!
interface Loopback 0
 ip address 10.10.10.10 255.255.255.255
!
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
 network 10.10.10.10 0.0.0.0 area 0
!
```

CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.2.2 255.255.255.0
!
interface Loopback 0
 ip address 20.20.20.20 255.255.255.255
!
router ospf 1
 network 10.1.2.0 0.0.0.255 area 0
 network 20.20.20.20 0.0.0.0 area 0
!
```

PE1 configuration file

```
hostname PE1
!
mpls enable
!
ip vrf VPN1
 rd 100:1
 route-target both 100:1
!
ip vrf VPN2
 rd 100:2
 route-target both 100:2
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 10.1.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPN2
 ip address 10.1.2.1 255.255.255.0
!
```

```
interface GigabitEthernet 0/3
 no switchport
 ip address 192.168.1.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 100
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback 0
 !
 address-family ipv4
  neighbor 2.2.2.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 2.2.2.2 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute ospf 10
 exit-address-family
 !
 address-family ipv4 vrf VPN2
  redistribute ospf 20
 exit-address-family
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN1
 redistribute bgp subnets
 network 10.1.1.0 0.0.0.255 area 0
!
router ospf 20 vrf VPN2
 redistribute bgp subnets
 network 10.1.2.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
 exit
!
```

PE2 configuration file

```
hostname PE2
!
mpls enable
!
ip vrf VPN1
 rd 101:1
 route-target both 100:1
!
ip vrf VPN2
 rd 101:2
 route-target both 100:2
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.1.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2.1
 encapsulation dot1Q 1
 ip vrf forwarding VPN1
 ip address 10.1.3.1 255.255.255.0
!
interface GigabitEthernet 0/2.2
 encapsulation dot1Q 2
 ip vrf forwarding VPN2
 ip address 10.1.4.1 255.255.255.0
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router bgp 100
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute ospf 10
 exit-address-family
```

```
 !
 address-family ipv4 vrf VPN2
  redistribute ospf 20
 exit-address-family
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN1
 network 10.1.3.0 0.0.0.255 area 0
 default-information originate
!
router ospf 20 vrf VPN2
 network 10.1.4.0 0.0.0.255 area 0
 default-information originate
!
mpls router ldp
 ldp router-id interface Loopback 0
!
ip route vrf VPN1 0.0.0.0 0.0.0.0 Null 0
ip route vrf VPN2 0.0.0.0 0.0.0.0 Null 0
!
```

MCE configuration file

```
hostname MCE
!
ip vrf VPN1
 rd 102:1
!
ip vrf VPN2
 rd 102:2
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN2
 ip address 10.1.5.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPN2
 ip address 10.1.6.1 255.255.255.0
!
interface GigabitEthernet 0/3.1
 encapsulation dot1Q 1
 ip vrf forwarding VPN1
 ip address 10.1.3.2 255.255.255.0
```

```
!
interface GigabitEthernet 0/3.2
 encapsulation dot1Q 2
 ip vrf forwarding VPN2
 ip address 10.1.4.2 255.255.255.0
!
router ospf 10 vrf VPN1
 network 10.1.3.0 0.0.0.255 area 0
 network 10.1.5.0 0.0.0.255 area 0
!
router ospf 20 vrf VPN2
 network 10.1.4.0 0.0.0.255 area 0
 network 10.1.6.0 0.0.0.255 area 0
!
```

CE3 configuration file

```
hostname CE3
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.5.2 255.255.255.0
!
interface Loopback 0
 ip address 30.30.30.30 255.255.255.255
!
router ospf 1
 network 10.1.5.0 0.0.0.255 area 0
 network 30.30.30.30 0.0.0.0 area 0
!
```

CE4 configuration file

```
hostname CE4
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.6.2 255.255.255.0
!
interface Loopback 0
 ip address 40.40.40.40 255.255.255.255
!
router ospf 1
 network 10.1.6.0 0.0.0.255 area 0
 network 40.40.40.40 0.0.0.0 area 0
!
```

7. Common Errors

The router ID is not 32 bits. As a result, the LDP session or BGP neighbor relationship fails to be established.

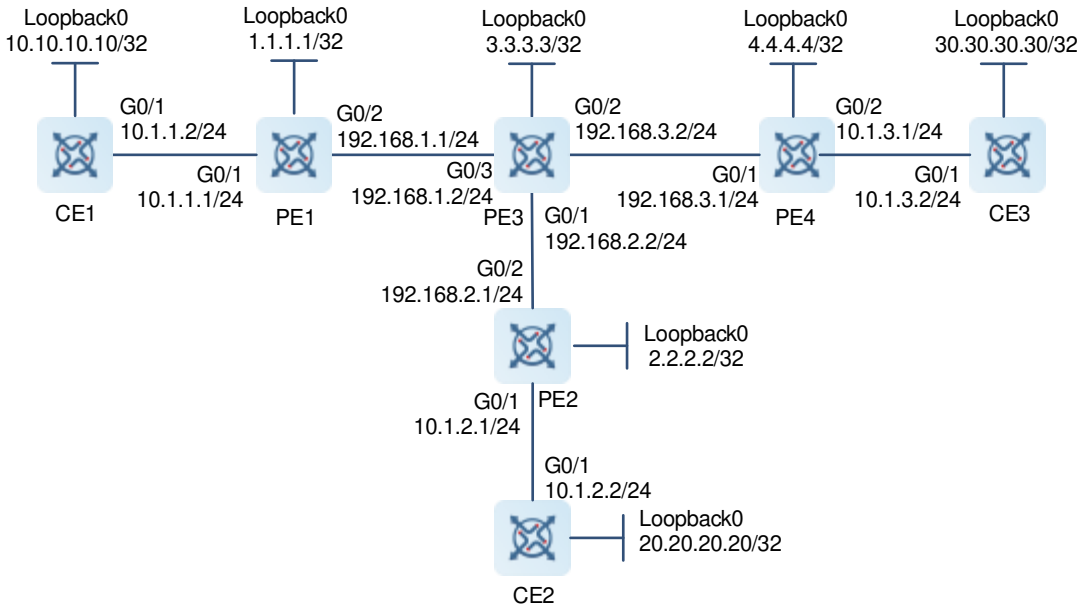
1.16.8 Configuring Basic IPv4 MPLS L3VPN Functions (Hierarchal VPNs Based on BGP Routing Policies)

1. Requirements

Use BGP routing policies to change the original MPLS L3VPN network to a hierarchal network. Figure 1-26 shows the topology. PE1 and PE2 are lower-layer PEs and do not save all VPN routes. They save only routes of associated VPN sites and default routes advertised by the upper-layer PE. PE3 and PE4 are upper-layer PEs and need to save all VPN routes, and PE3 is the upper-layer PE of PE1 and PE2.

2. Topology

Figure 1-26 Configuring Basic IPv4 MPLS L3VPN Functions (Hierarchal VPNs Based on BGP Routing Policies)



3. Notes

- Configure the MPLS network: configure interface addresses, OSPF, and the MPLS function on PE1, PE2, PE3, and PE4.
- Configure the MPLS L3VPN: configure a VRF instance on RR PE3, configure MP-IBGP on PE1, PE2, PE3, and PE4, and configure access from CEs to PEs.
- Configure routing policies on PE3 to advertise VPN routes only to PE1 and PE2.
- Configure default VPN routes on PE3 and advertise them to PE1 and PE2 but not to PE4.
- Configure the VRF-associated OSPF instance to advertise default routes on PE1 and PE2.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on all devices are similar. The following shows how to configure OSPF neighbors on PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# router ospf 1
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1(config-router)# network 192.168.1.0 0.0.0.255 area 0
PE1(config-router)# exit
```

(3) Configure basic MPLS functions.

Configurations on PEs are similar. The following shows how to configure basic MPLS functions on PE1.

```
PE1(config)# mpls enable
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/2)# label-switching
PE1(config-if-GigabitEthernet 0/2)# exit
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following uses PE1 as an example.

```
PE1(config)# ip vrf VPN1
PE1(config-vrf)# rd 100:1
PE1(config-vrf)# route-target both 100:1
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPN1
PE1(config-if-GigabitEthernet 0/1)# ip address 10.1.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# exit
```

(5) Configure VPN routes.

Configurations on PEs are similar. The following shows how to configure VPN routes on PE1.

```
PE1(config)# router ospf 10 vrf VPN1
PE1(config-router)# network 10.1.1.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# default-information originate
PE1(config-router)# exit
```

(6) Configure BGP neighbors to advertise VPN routes.

Configurations on PE1 and PE2 are similar. The following shows how to configure IBGP neighbors on PE1.

```
PE1(config)# router bgp 100
PE1(config-router)# neighbor 3.3.3.3 remote-as 100
PE1(config-router)# neighbor 3.3.3.3 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 3.3.3.3 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPN1
PE1(config-router-af)# redistribute ospf 10
```

```
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

5. Verification

(1) After the configuration is completed, run the **show ip route** command to display VPN routes.

PE1 verification result

```
PE1# show ip route vrf VPN1
Routing Table: VPN1

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is 3.3.3.3 to network 0.0.0.0
B*   0.0.0.0/0 [200/0] via 3.3.3.3, 00:00:09
C    10.1.1.0/24 is directly connected, GigabitEthernet 0/1, 00:00:09
L    10.1.1.1/32 is directly connected, GigabitEthernet 0/1, 00:00:09
O    10.10.10.10/32 [110/2] via 10.1.1.2, GigabitEthernet 0/1, 01:36:19
```

PE3 verification result

```
PE3# show ip route vrf VPN1
Routing Table: VPN1

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is 0.0.0.0 to network 0.0.0.0
S*   0.0.0.0/0 is directly connected, Null 0, 01:54:23
B    10.1.1.0/24 [200/1] via 1.1.1.1, 01:54:23
B    10.1.2.0/24 [200/1] via 2.2.2.2, 01:54:27
B    10.1.3.0/24 [200/1] via 4.4.4.4, 01:54:29
B    10.10.10.10/32 [200/1] via 1.1.1.1, 01:54:23
B    20.20.20.20/32 [200/2] via 2.2.2.2, 01:54:27
B    30.30.30.30/32 [200/1] via 4.4.4.4, 01:54:29
```

(2) VPN sites can ping each other.

CE1 verification result

```
CE1# ping 20.20.20.20 source 10.10.10.10
Sending 5, 100-byte ICMP Echoes to 20.20.20.20, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/10/20 ms
CE1# ping 30.30.30.30 source 10.10.10.10
Sending 5, 100-byte ICMP Echoes to 30.30.30.30, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/10/20 ms
```

CE2 verification result

```
CE2# ping 10.10.10.10 source 20.20.20.20
Sending 5, 100-byte ICMP Echoes to 10.10.10.10, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/10/20 ms
CE2# ping 30.30.30.30 source 20.20.20.20
Sending 5, 100-byte ICMP Echoes to 30.30.30.30, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/10/20 ms
```

CE3 verification result

```
CE3# ping 10.10.10.10 source 30.30.30.30
Sending 5, 100-byte ICMP Echoes to 10.10.10.10, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/10/20 ms
CE3# ping 20.20.20.20 source 30.30.30.30.
Sending 5, 100-byte ICMP Echoes to 20.20.20.20, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/10/20 ms
```

6. Configuration Files

CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.1.2 255.255.255.0
!
interface Loopback 0
 ip address 10.10.10.10 255.255.255.255
!
```

```
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
 network 10.10.10.10 0.0.0.0 area 0
!
```

PE1 configuration file

```
hostname PE1
!
mpls enable
!
ip vrf VPN1
 rd 100:1
 route-target both 100:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 10.1.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 192.168.1.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 100
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback 0
 !
 address-family ipv4
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute ospf 10
 exit-address-family
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
```

```
!
router ospf 10 vrf VPN1
 redistribute bgp subnets
 network 10.1.1.0 0.0.0.255 area 0
 default-information originate
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE3 configuration file

```
hostname PE3
!
mpls enable
!
route-map UPE_FILT_RMP permit 10
 match ip address UPE_FILT_ACL
!
route-map PE_FILT_RMP permit 10
 match ip address PE_FILT_ACL
!
ip vrf VPN1
 rd 100:3
 route-target both 100:1
!
ip access-list standard PE_FILT_ACL
 10 deny host 0.0.0.0
 20 permit any
!
ip access-list standard UPE_FITL_ACL
 10 permit host 0.0.0.0
 20 deny any
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.2.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 192.168.3.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 no switchport
```

```
 ip address 192.168.1.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 100
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback 0
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback 0
 neighbor 4.4.4.4 remote-as 100
 neighbor 4.4.4.4 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
  neighbor 2.2.2.2 activate
  neighbor 4.4.4.4 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 route-reflector-client
  neighbor 1.1.1.1 route-map UPE_FILT_RMP out
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 route-reflector-client
  neighbor 2.2.2.2 route-map UPE_FILT_RMP out
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 route-reflector-client
  neighbor 4.4.4.4 route-map PE_FILT_RMP out
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  network 0.0.0.0
 exit-address-family
!
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.3.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

```
ip route vrf VPN1 0.0.0.0 0.0.0.0 Null 0
!
```

PE2 configuration file

```
hostname PE2
!
mpls enable
!
ip vrf VPN1
 rd 100:2
 route-target both 100:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 10.1.2.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 192.168.2.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router bgp 100
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback 0
 !
 address-family ipv4
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.2.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN1
```

```
 redistribute bgp subnets
 network 10.1.2.0 0.0.0.255 area 0
 default-information originate
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.2.2 255.255.255.0
!
interface Loopback 0
 ip address 20.20.20.20 255.255.255.255
!
router ospf 1
 network 10.1.2.0 0.0.0.255 area 0
 network 20.20.20.20 0.0.0.0 area 0
!
```

PE4 configuration file

```
hostname PE4
!
mpls enable
!
ip vrf VPN1
 rd 100:4
 route-target both 100:1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.3.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPN1
 ip address 10.1.3.1 255.255.255.0
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router bgp 100
```

```
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback 0
 !
 address-family ipv4
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 4.4.4.4 0.0.0.0 area 0
 network 192.168.3.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN1
 redistribute bgp subnets
 network 10.1.3.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE4 configuration file

```
hostname CE4
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.3.2 255.255.255.0
!
interface Loopback 0
 ip address 30.30.30.30 255.255.255.255
!
router ospf 1
 network 10.1.3.0 0.0.0.255 area 0
 network 30.30.30.30 0.0.0.0 area 0
!
```

7. Common Errors

The router ID is not 32 bits. As a result, the LDP session or BGP neighbor relationship fails to be established.

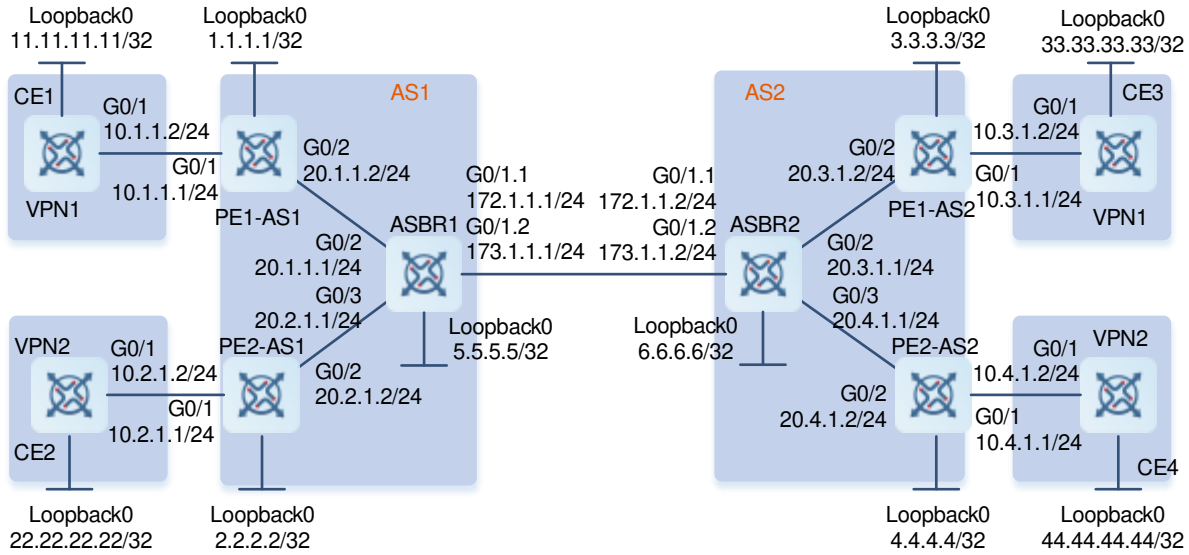
1.16.9 Configuring Inter-AS VPN Service Model – Option A

1. Requirements

In networks with few inter-AS VPNs, implement the Option A solution to provide inter-AS BGP/MPLS VPN services. An ASBR establishes a VRF instance for each VPN that needs to traverse domains and binds interfaces for these VRF instances. VRF instances between ASBRs exchange VPN routes by using these interfaces.

2. Topology

Figure 1-27 Inter-AS VPN in VRF-to-VRF Mode



3. Notes

The Option A solution requires an ASBR to configure an interface (usually logical sub-interface) for each inter-AS VPN and bind the interface to the inter-AS VPN. The number of bound interfaces should be at least equivalent to the number of inter-AS VPNs and the VPNs need to be configured one after another on the ASBR.

- The LDP router ID must be 32 bits.
- The BGP router ID must be 32 bits.
- An ASBR needs to configure an interface for each VPN and bind the interface to the VPN.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PE and ASBRs are similar. The following shows how to configure OSPF neighbors on PE1-AS1.

```
PE1-AS1> enable
PE1-AS1# configure terminal
PE1-AS1(config)# router ospf 1
PE1-AS1(config-router)# router-id 1.1.1.1
```

```
PE1-AS1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1-AS1(config-router)# network 20.1.1.0 0.0.0.255 area 0
```

(3) Configure basic MPLS functions.

Configurations on PE and ASBRs are similar. The following shows how to configure basic MPLS functions on PE1-AS1.

```
PE1-AS1(config)# mpls enable
PE1-AS1(config)# mpls router ldp
PE1-AS1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1-AS1(config-mpls-router)# exit
PE1-AS1(config)# interface gigabitethernet 0/2
PE1-AS1(config-if-GigabitEthernet 0/2)# label-switching
PE1-AS1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1-AS1(config-if-GigabitEthernet 0/2)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs and ASBRs are similar. The following shows how to create a VPN on PE-AS1.

```
PE1-AS1(config)# ip vrf VPN1
PE1-AS1(config-vrf)# rd 101:1
PE1-AS1(config-vrf)# route-target both 100:1
PE1-AS1(config-vrf)# exit
PE1-AS1(config)# interface gigabitethernet 0/1
PE1-AS1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPN1
PE1-AS1(config-if-GigabitEthernet 0/1)# ip address 10.1.1.1 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/1)# exit
```

Create a VPN on the ASBRs. The following shows how to create a VPN on ASBR1.

```
ASBR1(config)# ip vrf VPN1
ASBR1(config-vrf)# rd 301:1
ASBR1(config-vrf)# route-target both 100:1
ASBR1(config-vrf)# exit
ASBR1(config)# ip vrf VPN2
ASBR1(config-vrf)# rd 401:1
ASBR1(config-vrf)# route-target both 200:1
ASBR1(config-vrf)# exit
ASBR1(config)# interface gigabitethernet 0/1.1
ASBR1(config-if-GigabitEthernet 0/1)# encapsulation dot1Q 1
ASBR1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPN1
ASBR1(config-if-GigabitEthernet 0/1)# ip address 172.1.1.1 255.255.255.0
ASBR1(config-if-GigabitEthernet 0/1)# exit
ASBR1(config)# interface gigabitethernet 0/1.2
ASBR1(config-if-GigabitEthernet 0/1)# encapsulation dot1Q 2
ASBR1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPN2
ASBR1(config-if-GigabitEthernet 0/1)# ip address 173.1.1.1 255.255.255.0
ASBR1(config-if-GigabitEthernet 0/1)# exit
```

(5) Configure BGP neighbors to advertise VPN routes.

Configurations on PEs and CEs are similar. The following shows how to establish EBGP neighbor relationship on PE-AS1 and CE1.

PEs establish EBGP neighbor relationship with CEs to advertise VPN routes.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# address-family ipv4 vrf VPN1
PE1-AS1(config-router-af)# neighbor 10.1.1.2 remote-as 65001
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

CEs establish EBGP neighbor relationship with PEs to advertise VPN routes.

```
CE1> enable
CE1# configure terminal
CE1(config)# router bgp 65001
CE1(config-router)# neighbor 10.1.1.1 remote-as 1
CE1(config-router)# address-family ipv4
CE1(config-router-af)# neighbor 10.1.1.1 activate
CE1(config-router-af)# network 11.11.11.11 mask 255.255.255.255
CE1(config-router-af)# exit-address-family
CE1(config-router)# exit
```

Configurations on PEs and ASBRs are similar. The following shows how to establish EBGP neighbor relationship on PE-AS1 and ASBR1.

PEs establish IBGP neighbor relationship with ASBRs.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# neighbor 5.5.5.5 remote-as 1
PE1-AS1(config-router)# neighbor 5.5.5.5 update-source Loopback 0
PE1-AS1(config-router)# address-family ipv4 unicast
PE1-AS1(config-router-af)# neighbor 5.5.5.5 activate
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

ASBRs establish IBGP neighbor relationship with PEs.

```
ASBR1> enable
ASBR1# configure terminal
ASBR1(config)# router bgp 1
ASBR1(config-router)# neighbor 1.1.1.1 remote-as 1
ASBR1(config-router)# neighbor 1.1.1.1 update-source Loopback 0
ASBR1(config-router)# neighbor 2.2.2.2 remote-as 1
ASBR1(config-router)# neighbor 2.2.2.2 update-source Loopback 0
ASBR1(config-router)# address-family ipv4 unicast
ASBR1(config-router-af)# neighbor 1.1.1.1 activate
ASBR1(config-router-af)# neighbor 2.2.2.2 activate
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
```

ASBRs establish EBGP neighbor relationship with each other. The following shows how to establish EBGP neighbor relationship on ASBR1.

```
ASBR1> enable
ASBR1# configure terminal
ASBR1(config)# router bgp 1
ASBR1(config-router)# address-family ipv4 vrf VPN1
ASBR1(config-router-af)# neighbor 172.1.1.2 remote-as 2
ASBR1(config-router-af)# neighbor 172.1.1.2 activate
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)#address-family ipv4 vrf VPN2
ASBR1(config-router-af)#  neighbor 173.1.1.2 remote-as 2
ASBR1(config-router-af)#  neighbor 173.1.1.2 activate
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
```

5. Verification

Verify that CE1 can ping CE3 but cannot ping CE2 or CE4.

```
CE1# ping 33.33.33.33 source 11.11.11.11
Sending 5, 100-byte ICMP Echoes to 33.33.33.33, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/10/10 ms

CE1#ping 22.22.22.22 source 11.11.11.11
Sending 5, 100-byte ICMP Echoes to 22.22.22.22, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)

CE1#ping 44.44.44.44 source 11.11.11.11
Sending 5, 100-byte ICMP Echoes to 44.44.44.44, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)
```

6. Configuration Files

CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.1.1.2 255.255.255.0
!
interface Loopback 0
 ip address 11.11.11.11 255.255.255.255
!
router bgp 65001
 neighbor 10.1.1.1 remote-as 1
```

```
 !
 address-family ipv4
  network 11.11.11.11 mask 255.255.255.255
  neighbor 10.1.1.1 activate
 exit-address-family
!
```

CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.2.1.2 255.255.255.0
!
interface Loopback 0
 ip address 22.22.22.22 255.255.255.255
!
router bgp 65002
 neighbor 10.2.1.1 remote-as 1
 !
 address-family ipv4
  network 22.22.22.22 mask 255.255.255.255
  neighbor 10.2.1.1 activate
 exit-address-family
!
```

CE3 configuration file

```
hostname CE3
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.3.1.2 255.255.255.0
!
interface Loopback 0
 ip address 33.33.33.33 255.255.255.255
!
router bgp 65003
 neighbor 10.3.1.1 remote-as 1
 !
 address-family ipv4
  network 33.33.33.33 mask 255.255.255.255
  neighbor 10.3.1.1 activate
 exit-address-family
!
```

CE4 configuration file

```
hostname CE4
!
```

```
interface GigabitEthernet 0/1
 no switchport
 ip address 10.4.1.2 255.255.255.0
!
interface Loopback 0
 ip address 44.44.44.44 255.255.255.255
!
router bgp 65004
 neighbor 10.4.1.1 remote-as 1
 !
 address-family ipv4
  network 44.44.44.44 mask 255.255.255.255
  neighbor 10.4.1.1 activate
 exit-address-family
!
```

PE1-AS1 configuration file

```
hostname PE1-AS1
!
mpls enable
!
ip vrf VPN1
 rd 101:1
 route-target both 100:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 10.1.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.1.1.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 1
 neighbor 5.5.5.5 remote-as 1
 neighbor 5.5.5.5 update-source Loopback 0
 !
 address-family ipv4
  neighbor 5.5.5.5 activate
 exit-address-family
 !
```

```
 address-family vpnv4 unicast
  neighbor 5.5.5.5 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute connected
  neighbor 10.1.1.2 remote-as 65001
  neighbor 10.1.1.2 activate
 exit-address-family
!
router ospf 1
 router-id 1.1.1.1
 network 1.1.1.1 0.0.0.0 area 0
 network 20.1.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE1-AS2 configuration file

```
hostname PE1-AS2
!
mpls enable
!
ip vrf VPN1
 rd 201:1
 route-target both 100:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 10.3.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.3.1.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 2
 neighbor 6.6.6.6 remote-as 2
 neighbor 6.6.6.6 update-source Loopback 0
 !
 address-family ipv4
```

```
 no neighbor 6.6.6.6 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 6.6.6.6 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute connected
  neighbor 10.3.1.2 remote-as 65003
  neighbor 10.3.1.2 activate
 exit-address-family
!
router ospf 1
 router-id 3.3.3.3
 network 3.3.3.3 0.0.0.0 area 0
 network 20.3.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE2-AS1 configuration file

```
hostname PE2-AS1
!
mpls enable
!
ip vrf VPN2
 rd 102:1
 route-target both 200:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN2
 ip address 10.2.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.2.1.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router bgp 1
 neighbor 5.5.5.5 remote-as 1
```

```
 neighbor 5.5.5.5 update-source Loopback 0
 !
 address-family ipv4
 neighbor 5.5.5.5 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 5.5.5.5 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN2
  redistribute connected
  neighbor 10.2.1.2 remote-as 65002
  neighbor 10.2.1.2 activate
 exit-address-family
!
router ospf 1
 router-id 2.2.2.2
 network 2.2.2.2 0.0.0.0 area 0
 network 20.2.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE2-AS2 configuration file

```
hostname PE2-AS2
!
mpls enable
!
ip vrf VPN2
 rd 202:1
 route-target both 200:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN2
 ip address 10.4.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.4.1.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
```

```
!
router bgp 2
 neighbor 6.6.6.6 remote-as 2
 neighbor 6.6.6.6 update-source Loopback 0
 !
 address-family ipv4
  no neighbor 6.6.6.6 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 6.6.6.6 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN2
  redistribute connected
  neighbor 10.4.1.2 remote-as 65004
  neighbor 10.4.1.2 activate
 exit-address-family
!
router ospf 1
 router-id 4.4.4.4
 network 4.4.4.4 0.0.0.0 area 0
 network 20.4.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

ASBR1 configuration file

```
hostname ASBR1
!
mpls enable
!
ip vrf VPN1
 rd 301:1
 route-target both 100:1
!
ip vrf VPN2
 rd 401:1
 route-target both 200:1
!
interface GigabitEthernet 0/1
!
interface GigabitEthernet 0/1.1
 encapsulation dot1Q 1
 ip vrf forwarding VPN1
 ip address 172.1.1.1 255.255.255.0
```

```
!
interface GigabitEthernet 0/1.2
 encapsulation dot1Q 2
 ip vrf forwarding VPN2
 ip address 173.1.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.1.1.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 no switchport
 ip address 20.2.1.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
  neighbor 2.2.2.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
  neighbor 2.2.2.2 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 172.1.1.2 remote-as 2
  neighbor 172.1.1.2 activate
 exit-address-family
!
 address-family ipv4 vrf VPN2
  neighbor 173.1.1.2 remote-as 2
  neighbor 173.1.1.2 activate
 exit-address-family
```

```
!
router ospf 1
 router-id 5.5.5.5
 network 5.5.5.5 0.0.0.0 area 0
 network 20.1.1.0 0.0.0.255 area 0
 network 20.2.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

ASBR2 configuration file

```
hostname ASBR2
!
mpls enable
!
ip vrf VPN1
 rd 301:1
 route-target both 100:1
!
ip vrf VPN2
 rd 401:1
 route-target both 200:1
!
interface GigabitEthernet 0/1
!
interface GigabitEthernet 0/1.1
 encapsulation dot1Q 1
 ip vrf forwarding VPN1
 ip address 172.1.1.2 255.255.255.0
!
interface GigabitEthernet 0/1.2
 encapsulation dot1Q 2
 ip vrf forwarding VPN2
 ip address 173.1.1.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.3.1.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 no switchport
 ip address 20.4.1.1 255.255.255.0
 label-switching
 mpls ldp enable
```

```
!
interface Loopback 0
 ip address 6.6.6.6 255.255.255.255
!
router bgp 2
 neighbor 3.3.3.3 remote-as 2
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 4.4.4.4 remote-as 2
 neighbor 4.4.4.4 update-source Loopback 0
!
 address-family ipv4
  neighbor 3.3.3.3 activate
  neighbor 4.4.4.4 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 3.3.3.3 activate
  neighbor 4.4.4.4 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 172.1.1.1 remote-as 1
  neighbor 172.1.1.1 activate
 exit-address-family
!
 address-family ipv4 vrf VPN2
  neighbor 173.1.1.1 remote-as 1
  neighbor 173.1.1.1 activate
 exit-address-family
!
router ospf 1
 router-id 6.6.6.6
 network 6.6.6.6 0.0.0.0 area 0
 network 20.3.1.0 0.0.0.255 area 0
 network 20.4.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

7. Common Errors

The **ip vrf forwarding** command is not used to bind a VRF instance for an interface. As a result, no routing protocol is run between a PE and a CE. When the **show ip route vrf** command is run on the PE, no CE route is found.

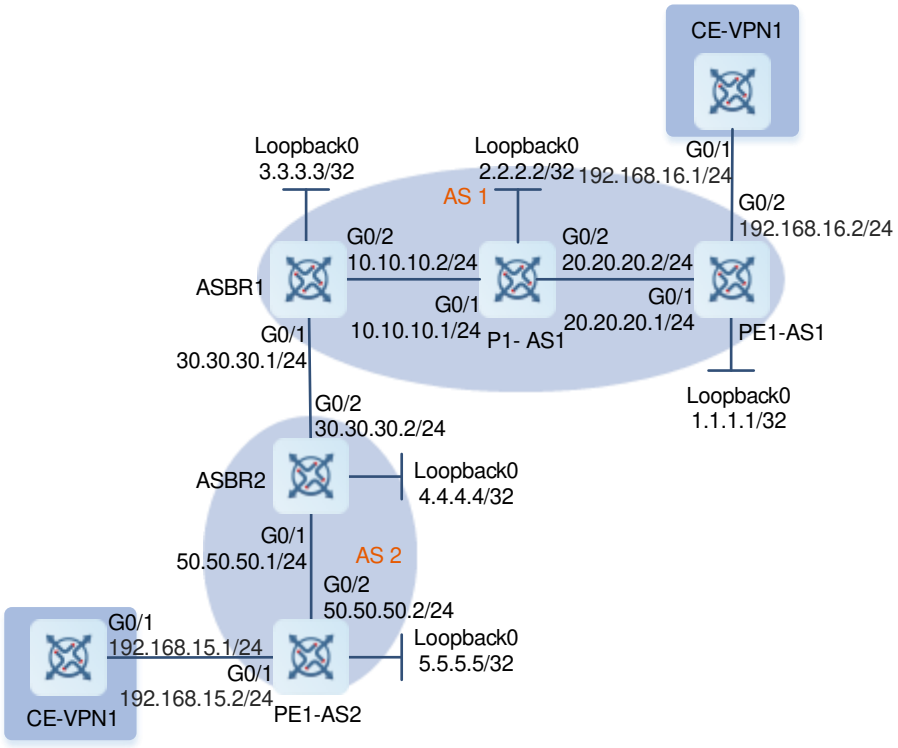
1.16.10 Configuring Inter-AS VPN Service Model – Option B (Next Hop Unchanged)

1. Requirements

A VPN has sites in two different ASs and requires the VPN sites in these two ASs to access each other.

2. Topology

Figure 1-28 Configuring Inter-AS VPN Service Model – Option B (Next Hop Unchanged)



3. Notes

- On PE1-AS1, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure the BGP, establish an MP-IBGP session with ASBR1, and configure a CE neighbor using EBGP. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On PE1-AS2, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure the BGP, establish an MP-IBGP session with ASBR2, and configure a CE neighbor using EBGP. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On P1-AS1, configure a loopback interface, configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On ASBR1, configure a loopback interface, configure BGP, disable the RT filtering function of BGP, and establish neighbor relationship with P1-AS1 and ASBR2. Configure MPLS signaling and enable MPLS on the public network interface, run OSPF on the backbone network to transmit routing information, and redistribute routes of directly-connected network segments. Configure an IP address for the interface used to connect to ASBR2 and enable labeled MPLS packet forwarding on the interface.

- On ASBR2, configure a loopback interface, configure the BGP, disable the RT filtering function of BGP, and establish neighbor relationship with PE1-AS2 and ASBR1. Configure MPLS signaling and enable MPLS on the public network interface, run OSPF on the backbone network to transmit routing information, and redistribute routes of directly-connected network segments. Configure an IP address for the interface used to connect to ASBR1 and enable labeled MPLS packet forwarding on the interface.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PEs, Ps, and ASBRs are similar. The following shows how to configure OSPF neighbors on PE1-AS1.

```
PE1-AS1> enable
PE1-AS1# configure terminal
PE1-AS1(config)# router ospf 10
PE1-AS1(config-router)# network 20.20.20.0 0.0.0.255 area 0
PE1-AS1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1-AS1(config-router)# end
```

(3) Configure basic MPLS functions.

Configurations on PEs, Ps, and ASBRs are similar. The following shows how to configure basic MPLS functions on PE1-AS1.

```
PE1-AS1(config)# mpls enable
PE1-AS1(config)# mpls router ldp
PE1-AS1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1-AS1(config-mpls-router)# exit
PE1-AS1(config)# interface gigabitethernet 0/1
PE1-AS1(config-if-GigabitEthernet 0/1)# label-switching
PE1-AS1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1-AS1(config-if-GigabitEthernet 0/1)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following shows how to create a VPN on PE1-AS1.

```
PE1-AS1(config)# ip vrf VPN1
PE1-AS1(config-vrf)# rd 1:100
PE1-AS1(config-vrf)# route-target both 1:100
PE1-AS1(config-vrf)# exit
PE1-AS1(config)# interface gigabitethernet 0/2
PE1-AS1(config-if-GigabitEthernet 0/2)# ip vrf forwarding VPN1
PE1-AS1(config-if-GigabitEthernet 0/2)# ip address 192.168.16.2 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/2)# exit
```

(5) Configure BGP neighbors to advertise VPN routes.

PE1-AS1 establishes EBGP neighbor relationship with CE1-VPN1.

```
PE1-AS1> enable
PE1-AS1# configure terminal
PE1-AS1(config)# router bgp 1
```

```
PE1-AS1(config-router)# address-family ipv4 vrf VPN1
PE1-AS1(config-router-af)# neighbor 192.168.16.1 remote-as 65001
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

CE1-VPN1 establishes EBGP neighbor relationship with PE1-AS1 to advertise VPN routes.

```
CE1-VPN1> enable
CE1-VPN1# configure terminal
CE1-VPN1(config)# router bgp 65001
CE1-VPN1(config-router)# neighbor 192.168.16.2 remote-as 1
CE1-VPN1(config-router)# address-family ipv4
CE1-VPN1(config-router-af)# neighbor 192.168.16.2 activate
CE1-VPN1(config-router-af)# network 10.10.10.10 mask 255.255.255.255
CE1-VPN1(config-router-af)# exit-address-family
CE1-VPN1(config-router)# exit
```

PE-ASs establish IBGP neighbor relationship with ASBRs.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# neighbor 3.3.3.3 remote-as 1
PE1-AS1(config-router)# neighbor 3.3.3.3 update-source loopback 0
PE1-AS1(config-router)# address-family vpnv4
PE1-AS1(config-router-af)# neighbor 3.3.3.3 activate
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)#
```

ASBRs establish IBGP neighbor relationship with PE-ASs.

```
ASBR1> enable
ASBR1# configure terminal
ASBR1(config)# router bgp 1
ASBR1(config-router)# neighbor 1.1.1.1 remote-as 1
ASBR1(config-router)# neighbor 1.1.1.1 update-source loopback 0
ASBR1(config-router)# no bgp default route-target filter
ASBR1(config-router)# address-family vpnv4
ASBR1(config-router-af)# neighbor 1.1.1.1 activate
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
```

ASBRs establish EBGP neighbor relationship with PE-ASs. The following example configures EBGP neighbor relationship for ASBR1.

```
ASBR1(config)# router bgp 1
ASBR1(config-router)# neighbor 30.30.30.2 remote-as 2
ASBR1(config-router)# address-family vpnv4 unicast
ASBR1(config-router-af)# neighbor 30.30.30.2 activate
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
```

Configure egress proxy for host routes on ASBR. The following example configures egress proxy for host routes on ASBR 1.

```
ASBR1(config)# mpls router ldp
```

```
ASBR1(config-mpls-router)# egress-proxy for host
```

5. Verification

(1) After the configuration is complete, verify that devices at the same VPN site can communicate with each other, and devices in different VPNs cannot communicate with each other.

PE1-AS1 verification result

```
PE1-AS1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    1.1.1.1/32 is directly connected, Loopback 0, 02:33:49
O    2.2.2.2/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 02:33:49
O    3.3.3.3/32 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 02:31:30
O    10.10.10.0/24 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 02:32:15
C    20.20.20.0/24 is directly connected, GigabitEthernet 0/0, 02:33:49
L    20.20.20.1/32 is directly connected, GigabitEthernet 0/0, 02:33:49
O E2 30.30.30.0/24 [110/20] via 20.20.20.2, GigabitEthernet 0/1, 00:39:24
O E2 30.30.30.2/32 [110/20] via 20.20.20.2, GigabitEthernet 0/1, 00:39:21

PE1-AS1# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms

PE1-AS1# ping 4.4.4.4
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)

PE1-AS1# ping 5.5.5.5
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)
```

PE1-AS2 verification result

```
PE1-AS2# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
O    4.4.4.4/32 [110/1] via 50.50.50.1, GigabitEthernet 0/1, 00:40:56
C    5.5.5.5/32 is directly connected, Loopback 0, 00:40:56
O E2 30.30.30.0/24 [110/20] via 50.50.50.1, GigabitEthernet 0/1, 00:40:56
O E2 30.30.30.1/32 [110/20] via 50.50.50.1, GigabitEthernet 0/1, 00:40:56
C    50.50.50.0/24 is directly connected, GigabitEthernet 0/1, 00:40:56
L    50.50.50.2/32 is directly connected, GigabitEthernet 0/1, 00:40:56

PE1-AS2# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)

PE1-AS2# ping 4.4.4.4
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/10 ms
```

CE1-VPN1 verification result

```
CE1-VPN1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    10.10.10.10/32 is directly connected, Loopback 0, 00:01:27
B    20.20.20.20/32 [20/0] via 192.168.16.2, 00:02:27
C    192.168.16.0/24 is directly connected, GigabitEthernet 0/1, 00:01:27
```

```
L    192.168.16.1/32 is directly connected, GigabitEthernet 0/1, 00:01:27

CE1-VPN1# ping 20.20.20.20 source 10.10.10.10
Sending 5, 100-byte ICMP Echoes to 20.20.20.20, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

CE2-VPN1 verification result

```
CE2-VPN1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
B    10.10.10.10/32 [20/0] via 192.168.15.2, 00:04:07
C    20.20.20.20/32 is directly connected, Loopback 0, 00:01:07
C    192.168.15.0/24 is directly connected, GigabitEthernet 0/0, 00:01:07
L    192.168.15.1/32 is directly connected, GigabitEthernet 0/0, 00:01:07

CE2-VPN1# ping 10.10.10.10 source 20.20.20.20
Sending 5, 100-byte ICMP Echoes to 10.10.10.10, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

6. Configuration Files

CE1-VPN1 configuration file

```
hostname CE1-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.16.1 255.255.255.0
!
interface Loopback 0
 ip address 10.10.10.10 255.255.255.255
!
router bgp 65001
 neighbor 192.168.16.2 remote-as 1
 !
 address-family ipv4
  network 10.10.10.10 mask 255.255.255.255
  neighbor 192.168.16.2 activate
```

```
 exit-address-family
!
```

PE1-AS1 configuration file

```
hostname PE1-AS1
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip address 20.20.20.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.16.2 255.255.255.0
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 1
 neighbor 3.3.3.3 remote-as 1
 neighbor 3.3.3.3 update-source Loopback 0
 !
 address-family ipv4
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.16.1 remote-as 65001
  neighbor 192.168.16.1 activate
 exit-address-family
!
router ospf 10
 network 1.1.1.1 0.0.0.0 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
```

```
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

P1-AS1 configuration file

```
hostname P1-AS1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.10.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.20.20.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router ospf 10
 network 2.2.2.2 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

ASBR1 configuration file

```
hostname ASBR1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 30.30.30.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 10.10.10.2 255.255.255.0
```

```
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 1
 no bgp default route-target filter
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 neighbor 30.30.30.2 remote-as 2
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
  neighbor 30.30.30.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
  neighbor 30.30.30.2 activate
 exit-address-family
!
router ospf 10
 redistribute connected subnets
 network 3.3.3.3 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
egress-proxy for host
!
```

ASBR2 configuration file

```
hostname ASBR2
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 50.50.50.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 30.30.30.2 255.255.255.0
 label-switching
```

```
 mpls ldp enable
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router bgp 2
 no bgp default route-target filter
 neighbor 5.5.5.5 remote-as 2
 neighbor 5.5.5.5 update-source Loopback 0
 neighbor 30.30.30.1 remote-as 1
 !
 address-family ipv4
  neighbor 5.5.5.5 activate
  neighbor 30.30.30.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 5.5.5.5 activate
  neighbor 30.30.30.1 activate
 exit-address-family
!
router ospf 10
 redistribute connected subnets
 network 4.4.4.4 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
egress-proxy for host
!
```

PE1-AS2 configuration file

```
hostname PE1-AS2
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.15.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
```

```
 ip address 50.50.50.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router bgp 2
 neighbor 4.4.4.4 remote-as 2
 neighbor 4.4.4.4 update-source Loopback 0
 !
 address-family ipv4
  neighbor 4.4.4.4 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 4.4.4.4 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.15.1 remote-as 65002
  neighbor 192.168.15.1 activate
 exit-address-family
!
router ospf 10
 network 5.5.5.5 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE2-VPN1 configuration file

```
hostname CE2-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.15.1 255.255.255.0
!
interface Loopback 0
 ip address 20.20.20.20 255.255.255.255
!
router bgp 65002
 neighbor 192.168.15.2 remote-as 2
 !
 address-family ipv4
  network 20.20.20.20 mask 255.255.255.255
```

```
 neighbor 192.168.15.2 activate
exit-address-family
!
```

7. Common Errors

- An LDP session fails to be established.
- The RT filtering function of BGP is enabled.
- The **redistribute connecter subnets** command is not used to redistribute routes of the directly-connected subnet to OSPF. As a result, the CE and VPN1 site cannot ping each other.

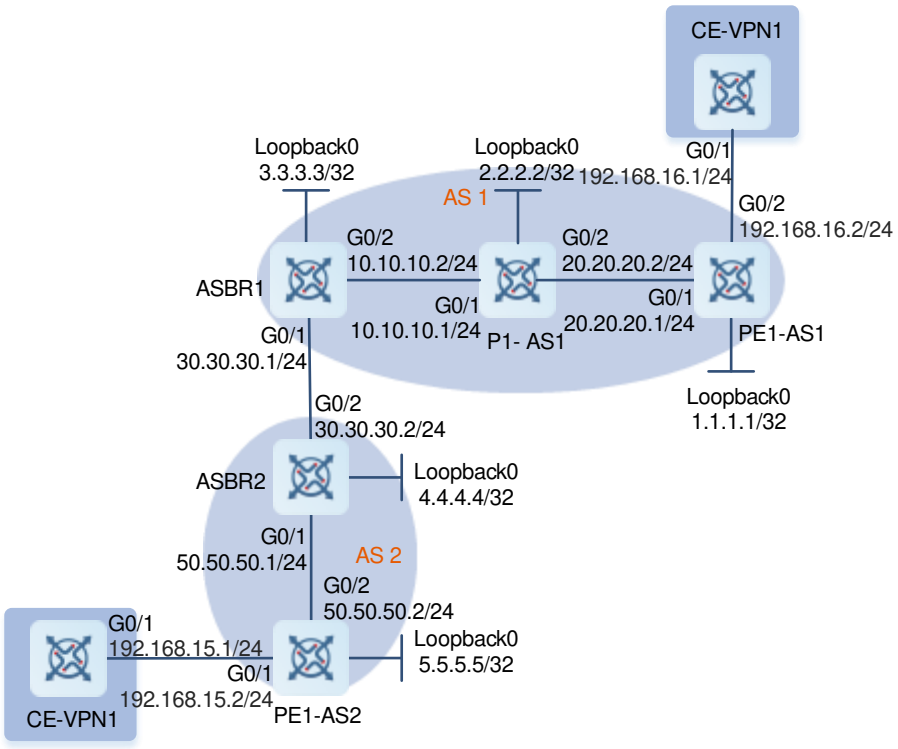
1.16.11 Configuring Inter-AS VPN Service Model – Option B (Next Hop Changed)

1. Requirements

A VPN has sites in two different ASs and requires the VPN sites in these two ASs to access each other.

2. Topology

Figure 1-29 Configuring Inter-AS VPN Service Model – Option B (Next Hop Changed)



3. Notes

- On PE1-AS1, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure the BGP, establish an MP-IBGP session with ASBR1, and configure a CE neighbor using EBGP. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On PE1-AS2, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure the BGP, establish an MP-IBGP

session with ASBR2, and configure a CE neighbor using EBGP. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.

- On P1-AS1, configure a loopback interface, configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On ASBR1, configure a loopback interface, configure the BGP, disable the RT filtering function of BGP, and establish neighbor relationship with PE1-AS1 and ASBR2. Configure MPLS signaling and enable MPLS on the public network interface, run OSPF on the backbone network to transmit routing information, and redistribute routes of directly-connected network segments. Configure an IP address for the interface used to connect to ASBR2 and enable labeled MPLS packet forwarding on the interface. Configure the ASBR to modify the next hop as its own address when sending VPN routes to an IBGP neighbor.
- On ASBR2, configure a loopback interface, configure the BGP, disable the RT filtering function of BGP, and establish neighbor relationship with PE1-AS2 and ASBR1. Configure MPLS signaling and enable MPLS on the public network interface, run OSPF on the backbone network to transmit routing information, and redistribute routes of directly-connected network segments. Configure an IP address for the interface used to connect to ASBR2 and enable labeled MPLS packet forwarding on the interface. Configure the ASBR to modify the next hop as its own address when sending VPN routes to an IBGP neighbor.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PEs, Ps, and ASBRs are similar. The following shows how to configure OSPF neighbors on PE1-AS1.

```
PE1-AS1> enable
PE1-AS1# configure terminal
PE1-AS1(config)# router ospf 10
PE1-AS1(config-router)# network 20.20.20.0 0.0.0.255 area 0
PE1-AS1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1-AS1(config-router)# end
```

(3) Configure basic MPLS functions.

Configurations on PEs, Ps, and ASBRs are similar. The following shows how to configure basic MPLS functions on PE1-AS1.

```
PE1-AS1(config)# mpls enable
PE1-AS1(config)# mpls router ldp
PE1-AS1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1-AS1(config-mpls-router)# exit
PE1-AS1(config)# interface gigabitethernet 0/1
PE1-AS1(config-if-GigabitEthernet 0/1)# ip address 20.20.20.1 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/1)# label-switching
PE1-AS1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1-AS1(config-if-GigabitEthernet 0/1)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following shows how to create a VPN on PE1-AS1.

```
PE1-AS1(config)# ip vrf VPN1
```

```
PE1-AS1(config-vrf)# rd 1:100
PE1-AS1(config-vrf)# route-target both 1:100
PE1-AS1(config-vrf)# exit
PE1-AS1(config)# interface gigabitethernet 0/2
PE1-AS1(config-if-GigabitEthernet 0/2)# ip vrf forwarding VPN1
PE1-AS1(config-if-GigabitEthernet 0/2)# ip address 192.168.16.2 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/2)# exit
```

(5) Configure BGP neighbors to advertise VPN routes.

PE1-AS1 establishes EBGP neighbor relationship with CE1-VPN1.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router-af)# address-family ipv4 vrf VPN1
PE1-AS1(config-router-af)# neighbor 192.168.16.1 remote-as 65001
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

CE1-VPN1 establishes EBGP neighbor relationship with PE1-AS1 to advertise VPN routes.

```
CE1-VPN1> enable
CE1-VPN1# configure terminal
CE1-VPN1(config)# router bgp 65001
CE1-VPN1(config-router)# neighbor 192.168.16.2 remote-as 1
CE1-VPN1(config-router)# address-family ipv4
CE1-VPN1(config-router-af)# neighbor 192.168.16.2 activate
CE1-VPN1(config-router-af)# network 10.10.10.10 mask 255.255.255.255
CE1-VPN1(config-router-af)# exit-address-family
CE1-VPN1(config-router)# exit
```

PE_ASs establish IBGP neighbor relationship with ASBRs.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# neighbor 3.3.3.3 remote-as 1
PE1-AS1(config-router)# neighbor 3.3.3.3 update-source loopback 0
PE1-AS1(config-router)# address-family vpnv4
PE1-AS1(config-router-af)# neighbor 3.3.3.3 activate
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

ASBRs establish IBGP neighbor relationship with PE-ASs.

```
ASBR1> enable
ASBR1# configure terminal
ASBR1(config)# router bgp 1
ASBR1(config-router)# neighbor 1.1.1.1 remote-as 1
ASBR1(config-router)# neighbor 1.1.1.1 update-source loopback 0
ASBR1(config-router)# no bgp default route-target filter
ASBR1(config-router)# address-family vpnv4
ASBR1(config-router-af)# neighbor 1.1.1.1 activate
ASBR1(config-router-af)# neighbor 1.1.1.1 next-hop-self
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
```

ASBRs establish EBGP neighbor relationship with each other. The following shows how to establish EBGP neighbor relationship on ASBR1.

```
ASBR1(config)# router bgp 1
ASBR1(config-router)# neighbor 30.30.30.2 remote-as 2
ASBR1(config-router)# address-family vpnv4 unicast
ASBR1(config-router-af)# neighbor 30.30.30.2 activate
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
```

Configure egress proxy for host routes on ASBR. The following example configures egress proxy for host routes on ASBR 1.

```
ASBR1(config)# mpls router ldp
ASBR1(config-mpls-router)# egress-proxy for host
```

5. Verification

(1) After the configuration is completed, verify that devices at the same VPN site can communicate with each other and devices at different VPN sites cannot communicate with each other.

PE1-AS1 verification result

```
PE1-AS1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    1.1.1.1/32 is directly connected, Loopback 0, 00:33:49
O    2.2.2.2/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 02:33:49
O    3.3.3.3/32 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 02:31:30
O    10.10.10.0/24 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 02:32:15
C    20.20.20.0/24 is directly connected, GigabitEthernet 0/0, 00:33:49
L    20.20.20.1/32 is directly connected, GigabitEthernet 0/0, 00:33:49
O E2 30.30.30.0/24 [110/20] via 20.20.20.2, GigabitEthernet 0/1, 00:39:24
O E2 30.30.30.2/32 [110/20] via 20.20.20.2, GigabitEthernet 0/1, 00:39:21

PE1-AS1# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms

PE1-AS1# ping 4.4.4.4
```

```
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)

PE1-AS1# ping 5.5.5.5
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)
```

PE1-AS2 verification result

```
PE1-AS2# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
O    4.4.4.4/32 [110/1] via 50.50.50.1, GigabitEthernet 0/1, 00:40:56
C    5.5.5.5/32 is directly connected, Loopback 0, 00:00:56
O E2 30.30.30.0/24 [110/20] via 50.50.50.1, GigabitEthernet 0/1, 00:40:56
O E2 30.30.30.1/32 [110/20] via 50.50.50.1, GigabitEthernet 0/1, 00:40:56
C    50.50.50.0/24 is directly connected, GigabitEthernet 0/1, 00:00:56
L    50.50.50.2/32 is directly connected, GigabitEthernet 0/1, 00:00:56

PE1-AS2# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5)

PE1-AS2# ping 4.4.4.4
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/10 ms
```

CE1-VPN1 verification result

```
CE1-VPN1# show ip route

Codes:  C - Connected, L - Local, S - Static
```

```
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    10.10.10.10/32 is directly connected, Loopback 0, 00:00:27
B    20.20.20.20/32 [20/0] via 192.168.16.2, 00:02:27
C    192.168.16.0/24 is directly connected, GigabitEthernet 0/1, 00:00:27
L    192.168.16.1/32 is directly connected, GigabitEthernet 0/1, 00:00:27

CE1-VPN1# ping 20.20.20.20 source 10.10.10.10
Sending 5, 100-byte ICMP Echoes to 20.20.20.20, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

CE2-VPN1 verification result

```
CE2-VPN1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
B    10.10.10.10/32 [20/0] via 192.168.15.2, 00:04:07
C    20.20.20.20/32 is directly connected, Loopback 0, 00:00:07
C    192.168.15.0/24 is directly connected, GigabitEthernet 0/0, 00:00:07
L    192.168.15.1/32 is directly connected, GigabitEthernet 0/0, 00:00:07

CE2-VPN1# ping 10.10.10.10 source 20.20.20.20
Sending 5, 100-byte ICMP Echoes to 10.10.10.10, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

6. Configuration Files

CE1-VPN1 configuration file

```
hostname CE1-VPN1
!
interface GigabitEthernet 0/1
```

```
 no switchport
 ip address 192.168.16.1 255.255.255.0
!
interface Loopback 0
 ip address 10.10.10.10 255.255.255.255
!
router bgp 65001
 neighbor 192.168.16.2 remote-as 1
 !
 address-family ipv4
  network 10.10.10.10 mask 255.255.255.255
  neighbor 192.168.16.2 activate
 exit-address-family
!
```

PE1-AS1 configuration file

```
hostname PE1-AS1
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip address 20.20.20.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.16.2 255.255.255.0
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 1
 neighbor 3.3.3.3 remote-as 1
 neighbor 3.3.3.3 update-source Loopback 0
 !
 address-family ipv4
  neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family vpnv4 unicast
```

```
 neighbor 3.3.3.3 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.16.1 remote-as 65001
  neighbor 192.168.16.1 activate
 exit-address-family
!
router ospf 10
 network 1.1.1.1 0.0.0.0 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

P1-AS1 configuration file

```
hostname P1-AS1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.10.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.20.20.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router ospf 10
 network 2.2.2.2 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

ASBR1 configuration file

```
hostname ASBR1
```

```
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 30.30.30.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 10.10.10.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 1
 no bgp default route-target filter
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 neighbor 30.30.30.2 remote-as 2
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
  neighbor 30.30.30.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 next-hop-self
  neighbor 30.30.30.2 activate
 exit-address-family
!
router ospf 10
 network 3.3.3.3 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
egress-proxy for host
!
```

ASBR2 configuration file

```
hostname ASBR2
```

```
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 50.50.50.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 30.30.30.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router bgp 2
 no bgp default route-target filter
 neighbor 5.5.5.5 remote-as 2
 neighbor 5.5.5.5 update-source Loopback 0
 neighbor 30.30.30.1 remote-as 1
 !
 address-family ipv4
  neighbor 5.5.5.5 activate
  neighbor 30.30.30.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 5.5.5.5 activate
  neighbor 5.5.5.5 next-hop-self
  neighbor 30.30.30.1 activate
 exit-address-family
!
router ospf 10
 network 4.4.4.4 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
egress-proxy for host
!
```

PE1-AS2 configuration file

```
hostname PE1-AS2
```

```
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.15.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 50.50.50.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router bgp 2
 neighbor 4.4.4.4 remote-as 2
 neighbor 4.4.4.4 update-source Loopback 0
 !
 address-family ipv4
  neighbor 4.4.4.4 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 4.4.4.4 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.15.1 remote-as 65002
  neighbor 192.168.15.1 activate
 exit-address-family
!
router ospf 10
 network 5.5.5.5 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE2-VPN1 configuration file

```
hostname CE2-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.15.1 255.255.255.0
!
interface Loopback 0
 ip address 20.20.20.20 255.255.255.255
!
router bgp 65002
 neighbor 192.168.15.2 remote-as 2
 !
 address-family ipv4
  network 20.20.20.20 mask 255.255.255.255
  neighbor 192.168.15.2 activate
 exit-address-family
!
```

7. Common Errors

- An LDP session fails to be established.
- The RT filtering function of BGP is enabled.
- When distributing VPN routes to IBGP neighbors, an ASBR does not configure itself as the next hop. As a result, the CE and VPN1 cannot ping each other.

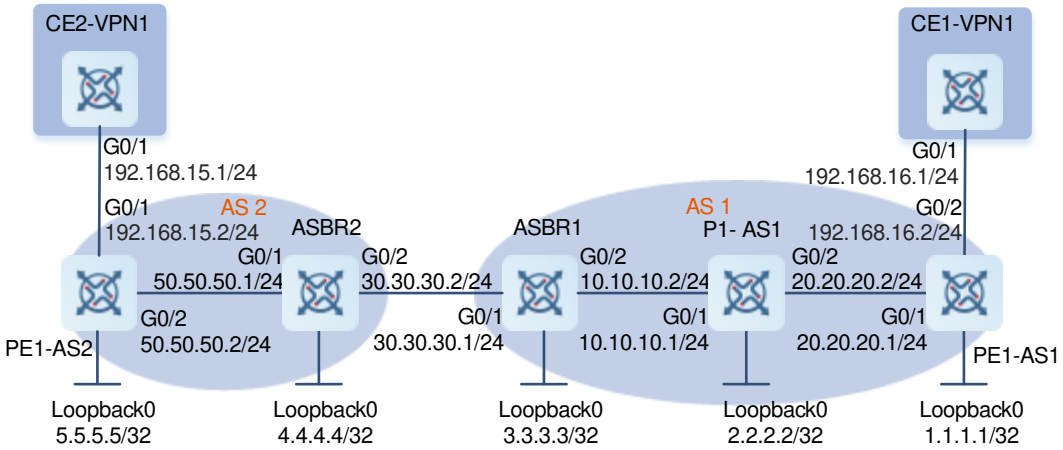
1.16.12 Configuring Inter-AS VPN Service Model – Option C (Enabling Label Switching for IPv4 Routes with EBGP Neighbors)

1. Requirements

A VPN has sites in two different ASs and requires the VPN sites in these two ASs to access each other.

2. Topology

Figure 1-30 Configuring Inter-AS VPN Service Model – Option C (Enabling Label Switching for IPv4 Routes with EBGP Neighbors)



3. Notes

- On PE1-AS1, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure a multi-hop MP-EBGP session, disable IPv4 route exchange in the multi-hop EBGP session, and configure CE neighbors using EBGP. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On PE1-AS2, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure a multi-hop MP-EBGP session, disable IPv4 route exchange in the multi-hop EBGP session, and configure CE neighbors using EBGP. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On P1-AS1, configure a loopback interface, configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On ASBR1, configure a loopback interface and configure an ACL rule to distribute or set labels only for routes that meet specific conditions. Establish an EBGP session with ASBR2 and configure a route-map rule to configure labels for PE routes that meet conditions, and statically configure PE routes in the local AS. The route-map rule is optional, and application of a route-map rule enables the BGP to distribute labels only to specific routes. Configure MPLS and use an ACL rule to distribute labels only to specific BGP routes. The ACL rule is optional, and application of an ACL rule can reduce unnecessary route entries. Configure a backbone network routing protocol and redistribute only BGP routes that meet the route-map rule. The route-map rule is optional, and application of a route-map rule can reduce unnecessary route entries. Configure an IP address for the interface used to connect to ASBR2 and enable labeled MPLS packet forwarding on the interface.
- On ASBR2, configure a loopback interface and configure an ACL rule to distribute or set labels only for routes that meet specific conditions. Establish an EBGP session with ASBR1 and configure a route-map rule to configure labels for PE routes that meet conditions, and statically configure PE routes in the local AS. The route-map rule is optional, and application of a route-map rule enables the BGP to distribute labels only to specific routes. Configure MPLS and use an ACL rule to distribute labels only to specific BGP routes. The ACL rule is optional, and application of an ACL rule can reduce unnecessary route entries. Configure a backbone network routing protocol and redistribute only BGP routes that meet the route-map rule. The route-map rule is optional, and application of a route-map rule can reduce unnecessary route entries. Configure an IP address for the interface used to connect to ASBR1 and enable labeled MPLS packet forwarding on the interface.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PEs, Ps, and ASBRs are similar. The following shows how to configure OSPF neighbors on PE1-AS1.

```
PE1-AS1(config)# router ospf 10
PE1-AS1(config-router)# network 20.20.20.0 0.0.0.255 area 0
```

```
PE1-AS1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1-AS1(config-router)# exit
```

(3) Configure basic MPLS functions.

Configurations on PEs, Ps, and ASBRs are similar. The following shows how to configure basic MPLS functions on PE1-AS1.

```
PE1-AS1(config)# mpls enable
PE1-AS1(config)# mpls router ldp
PE1-AS1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1-AS1(config-mpls-router)# exit
PE1-AS1(config)# interface gigabitethernet 0/1
PE1-AS1(config-if-GigabitEthernet 0/1)# ip address 20.20.20.1 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/1)# label-switching
PE1-AS1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1-AS1(config-if-GigabitEthernet 0/1)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following shows how to create a VPN on PE1-AS1.

```
PE1-AS1(config)# ip vrf VPN1
PE1-AS1(config-vrf)# rd 1:100
PE1-AS1(config-vrf)# route-target both 1:100
PE1-AS1(config-vrf)# exit
PE1-AS1(config)# interface gigabitethernet 0/2
PE1-AS1(config-if-GigabitEthernet 0/2)# ip vrf forwarding VPN1
PE1-AS1(config-if-GigabitEthernet 0/2)# ip address 192.168.16.2 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/2)# exit
```

(5) Configure BGP neighbors to advertise VPN routes.

PE1-AS1 establishes EBGP neighbor relationship with CE1-VPN1.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# address-family ipv4 vrf VPN1
PE1-AS1(config-router-af)# neighbor 192.168.16.1 remote-as 65001
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

CE1-VPN1 establishes EBGP neighbor relationship with PE1-AS1 to advertise VPN routes.

```
CE1-VPN1(config)# router bgp 65001
CE1-VPN1(config-router)# neighbor 192.168.16.2 remote-as 1
CE1-VPN1(config-router)# address-family ipv4
CE1-VPN1(config-router-af)# neighbor 192.168.16.2 activate
CE1-VPN1(config-router-af)# network 10.10.10.10 mask 255.255.255.255
CE1-VPN1(config-router-af)# exit-address-family
CE1-VPN1(config-router)# exit
```

PE-ASs establish EBGP neighbor relationship with each other. The following shows how to establish EBGP neighbor relationship on PE1-AS1.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# neighbor 5.5.5.5 remote-as 2
```

```
PE1-AS1(config-router)# neighbor 5.5.5.5 update-source loopback 0
PE1-AS1(config-router)# neighbor 5.5.5.5 ebgp-multihop
PE1-AS1(config-router)# address-family ipv4
PE1-AS1(config-router-af)# no neighbor 5.5.5.5 activate
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# address-family vpnv4 unicast
PE1-AS1(config-router-af)# neighbor 5.5.5.5 activate
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

ASBR1 establishes EBGP neighbor relationship with ASBR2 and redistributes BGP routes. The following shows how to establish EBGP neighbor relationship between ASBR1 and ASBR2.

```
ASBR1(config)# ip access-list extended 101
ASBR1(config-ext-nacl)# permit ip host 1.1.1.1 any
ASBR1(config-ext-nacl)# exit
ASBR1(config)# ip access-list extended 102
ASBR1(config-ext-nacl)# permit ip host 5.5.5.5 any
ASBR1(config-ext-nacl)# exit
ASBR1(config)# route-map set-mpls permit
ASBR1(config-route-map)# match ip address 101
ASBR1(config-route-map)# set mpls-label
ASBR1(config-route-map)# exit
ASBR1(config)# route-map external-pe-route permit
ASBR1(config-route-map)# match ip address 102
ASBR1(config-route-map)# exit
ASBR1(config)# router bgp 1
ASBR1(config-router)# neighbor 30.30.30.2 remote-as 2
ASBR1(config-router)# address-family ipv4
ASBR1(config-router-af)# neighbor 30.30.30.2 activate
ASBR1(config-router-af)# neighbor 30.30.30.2 send-label
ASBR1(config-router-af)# neighbor 30.30.30.2 route-map set-mpls out
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
ASBR1(config)# router ospf 10
ASBR1(config-router)# redistribute bgp route-map external-pe-route subnets
ASBR1(config-router)# network 3.3.3.3 0.0.0.0 area 0
ASBR1(config-router)# network 10.10.10.0 0.0.0.255 area 0
ASBR1(config-router)# exit
```

5. Verification

(1) After the configuration is completed, verify that devices at different sites of a VPN can communicate with each other.

PE1-AS1 verification result

```
PE1-AS1# show ip route

Codes:  C - Connected, L - Local, S - Static
```

```
      R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      IA - Inter area, EV - BGP EVPN, A - Arp to host
      LA - Local aggregate route
      * - candidate default

Gateway of last resort is no set
C   1.1.1.1/32 is directly connected, Loopback 0, 00:04:47
O   2.2.2.2/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 01:04:47
O   3.3.3.3/32 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 01:03:26
O E2 5.5.5.5/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 00:50:44
O   10.10.10.0/24 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 01:04:58
C   20.20.20.0/24 is directly connected, GigabitEthernet 0/1, 00:04:47
L   20.20.20.1/32 is directly connected, GigabitEthernet 0/1, 00:04:47

PE1-AS1# ping 5.5.5.5
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS1# ping 5.5.5.5 source 1.1.1.1
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1# ping 4.4.4.4
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS1# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1#ping 2.2.2.2
Sending 5, 100-byte ICMP Echoes to 2.2.2.2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

PE1-AS2 verification result

```
PE1-AS2# show ip route

Codes:  C - Connected, L - Local, S - Static
      R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      IA - Inter area, EV - BGP EVPN, A - Arp to host
```

```
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
O E2 1.1.1.1/32 [110/1] via 50.50.50.1, GigabitEthernet 0/1, 00:38:28
O    4.4.4.4/32 [110/1] via 50.50.50.1, GigabitEthernet 0/1, 01:09:52
C    5.5.5.5/32 is directly connected, Loopback 0, 00:30:28
C    50.50.50.0/24 is directly connected, GigabitEthernet 0/1, 00:30:28
L    50.50.50.2/32 is directly connected, GigabitEthernet 0/1, 00:30:28

PE1-AS2# ping 1.1.1.1
Sending 5, 100-byte ICMP Echoes to 1.1.1.1, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS2# ping 1.1.1.1 source 5.5.5.5
Sending 5, 100-byte ICMP Echoes to 1.1.1.1, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS2# ping 4.4.4.4
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS2# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
```

6. Configuration Files

CE1-VPN1 configuration file

```
hostname CE1-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.16.1 255.255.255.0
!
interface Loopback 0
 ip address 10.10.10.10 255.255.255.255
!
router bgp 65001
 neighbor 192.168.16.2 remote-as 1
 !
 address-family ipv4
  network 10.10.10.10 mask 255.255.255.255
  neighbor 192.168.16.2 activate
 exit-address-family
!
```

PE1-AS1 configuration file

```
hostname PE1-AS1
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip address 20.20.20.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.16.2 255.255.255.0
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 1
 neighbor 5.5.5.5 remote-as 2
 neighbor 5.5.5.5 ebgp-multihop 255
 neighbor 5.5.5.5 update-source Loopback 0
 !
 address-family ipv4
  no neighbor 5.5.5.5 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 5.5.5.5 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.16.1 remote-as 65001
  neighbor 192.168.16.1 activate
 exit-address-family
!
router ospf 10
 network 1.1.1.1 0.0.0.0 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
mpls router ldp
```

```
 ldp router-id interface Loopback 0 force
!
```

P1-AS1 configuration file

```
hostname P1-AS1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.10.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.20.20.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router ospf 10
 network 2.2.2.2 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

ASBR1 configuration file

```
hostname ASBR1
!
mpls enable
!
route-map set-mpls permit 10
 match ip address 101
 set mpls-label
!
route-map external-pe-route permit 10
 match ip address 102
!
ip access-list extended 101
 10 permit ip host 1.1.1.1 any
!
```

```
ip access-list extended 102
 10 permit ip host 5.5.5.5 any
!
interface GigabitEthernet 0/1
 no switchport
 ip address 30.30.30.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 10.10.10.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 1
 neighbor 30.30.30.2 remote-as 2
 !
 address-family ipv4
  network 1.1.1.1 mask 255.255.255.255
  neighbor 30.30.30.2 activate
  neighbor 30.30.30.2 send-label
  neighbor 30.30.30.2 route-map set-mpls out
 exit-address-family
!
router ospf 10
 redistribute bgp route-map external-pe-route subnets
 network 3.3.3.3 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
 advertise-labels for bgp-routes acl 102
!
```

ASBR2 configuration file

```
hostname ASBR2
!
mpls enable
!
route-map set-mpls permit 10
 match ip address 101
 set mpls-label
!
```

```
route-map external-pe-route permit 10
 match ip address 102
!
ip access-list extended 101
 10 permit ip host 1.1.1.1 any
!
ip access-list extended 102
 10 permit ip host 5.5.5.5 any
!
interface GigabitEthernet 0/1
 no switchport
 ip address 50.50.50.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 30.30.30.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router bgp 2
 neighbor 30.30.30.1 remote-as 1
 !
 address-family ipv4
  network 5.5.5.5 mask 255.255.255.255
  neighbor 30.30.30.1 activate
  neighbor 30.30.30.1 send-label
  neighbor 30.30.30.1 route-map set-mpls out
 exit-address-family
!
router ospf 10
 redistribute bgp route-map external-pe-route subnets
 network 4.4.4.4 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
 advertise-labels for bgp-routes acl 102
!
```

PE1-AS2 configuration file

```
hostname PE1-AS2
```

```
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.15.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 50.50.50.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router bgp 2
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 ebgp-multihop 255
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family ipv4
  no neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.15.1 remote-as 65002
  neighbor 192.168.15.1 activate
 exit-address-family
!
router ospf 10
 network 5.5.5.5 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE2-VPN1 configuration file

```
hostname CE2-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.15.1 255.255.255.0
!
interface Loopback 0
 ip address 20.20.20.20 255.255.255.255
!
router bgp 65002
 neighbor 192.168.15.2 remote-as 2
 !
 address-family ipv4
  network 20.20.20.20 mask 255.255.255.255
  neighbor 192.168.15.2 activate
 exit-address-family
!
```

7. Common Errors

- When a multi-hop MP-EBGP session is configured, IPv4 route exchange is not disabled for the multi-hop EBGP session.
- ASBRs are not configured to use LDP to distribute labels to BGP routes. As a result, VPN sites cannot interconnect with each other.

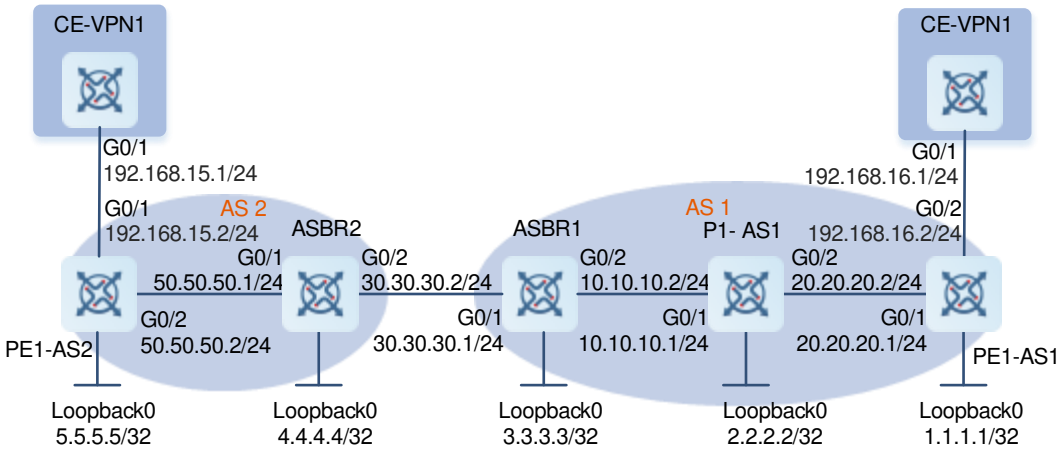
1.16.13 Configuring Inter-AS Service Model – Option C (Enabling Label Switching for IPv4 Routes with EBGP and IBGP Neighbors)

1. Requirements

A VPN has sites in two different ASs and requires the VPN sites in these two ASs to access each other.

2. Topology

Figure 1-31 Configuring Inter-AS Service Model – Option C (Enabling Label Switching for IPv4 Routes with EBGP and IBGP Neighbors)



3. Notes

- On PE1-AS1, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Establish a multi-hop MP-EBGP session and disable IPv4 route exchange for the multi-hop EBGP session. Establish an IBGP session with ASBR1, enable label switching for IPv4 routes, and configure CE neighbors using EBGP. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On PE1-AS2, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Establish a multi-hop MP-EBGP session and disable IPv4 route exchange for the multi-hop MP-EBGP session. Establish an IBGP session with ASBR2, enable label switching for IPv4 routes, and configure CE neighbors using EBGP. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On P1-AS1, configure a loopback interface, configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On ASBR1, configure a loopback interface and configure an ACL rule to distribute or set labels only for routes that meet specific conditions. Establish an EBGP session with ASBR2 and configure a route-map rule to configure labels for PE routes that meet conditions, and statically configure PE routes in the local AS. The route-map rule is optional, and application of a route-map rule enables the BGP to distribute labels only to specific routes. Configure MPLS signaling, enable MPLS on an interface, and run OSPF on the backbone network to transmit routing information. Configure an IP address for the interface used to connect to ASBR2 and enable labeled MPLS packet forwarding on the interface.
- On ASBR2, configure a loopback interface and configure an ACL rule to distribute or set labels only for routes that meet specific conditions. Establish an EBGP session with ASBR1, configure a route-map rule to configure labels for PE routes that meet conditions, and statically configure PE routes in the local AS. The route-map rule is optional, and application of a route-map rule to enable the BGP to distribute labels only to specific routes. Configure MPLS signaling, enable MPLS on an interface, and run OSPF on the backbone

network to transmit routing information. Configure an IP address for the interface used to connect to ASBR1 and enable labeled MPLS packet forwarding on the interface.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PEs, Ps, and ASBRs are similar. The following shows how to configure OSPF neighbors on PE1-AS1.

```
PE1-AS1> enable
PE1-AS1# configure terminal
PE1-AS1(config)# router ospf 10
PE1-AS1(config-router)# network 20.20.20.0 0.0.0.255 area 0
PE1-AS1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1-AS1(config-router)# end
```

(3) Configure basic MPLS functions.

Configurations on PEs, Ps, and ASBRs are similar. The following shows how to configure basic MPLS functions on PE1-AS1.

```
PE1-AS1(config)# mpls enable
PE1-AS1(config)# mpls router ldp
PE1-AS1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1-AS1(config-mpls-router)# exit
PE1-AS1(config)# interface gigabitethernet 0/1
PE1-AS1(config-if-GigabitEthernet 0/1)# ip address 20.20.20.1 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/1)# label-switching
PE1-AS1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1-AS1(config-if-GigabitEthernet 0/1)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following shows how to create a VPN on PE1-AS1.

```
PE1-AS1(config)# ip vrf VPN1
PE1-AS1(config-vrf)# rd 1:100
PE1-AS1(config-vrf)# route-target both 1:100
PE1-AS1(config-vrf)# exit
PE1-AS1(config)# interface gigabitethernet 0/2
PE1-AS1(config-if-GigabitEthernet 0/2)# ip vrf forwarding VPN1
PE1-AS1(config-if-GigabitEthernet 0/2)# ip address 192.168.16.2 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/2)# exit
```

(5) Configure BGP neighbors to advertise VPN routes.

PE1-AS1 establishes EBGP neighbor relationship with CE1-VPN1.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config)# address-family ipv4 vrf VPN1
PE1-AS1(config-router-af)# neighbor 192.168.16.1 remote-as 65001
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router-af)# exit
```

CE1-VPN1 establishes EBGP neighbor relationship with PE1-AS1 to advertise VPN routes.

```
CE1-VPN1(config)# router bgp 65001
CE1-VPN1(config-router)# neighbor 192.168.16.2 remote-as 1
CE1-VPN1(config-router)# address-family ipv4
CE1-VPN1(config-router-af)# neighbor 192.168.16.2 activate
CE1-VPN1(config-router-af)# network 10.10.10.10 mask 255.255.255.255
CE1-VPN1(config-router-af)# exit-address-family
CE1-VPN1(config-router)# exit
```

PE1-AS1 establishes IBGP neighbor relationship with ASBR1.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# neighbor 3.3.3.3 remote-as 1
PE1-AS1(config-router)# neighbor 3.3.3.3 update-source loopback 0
PE1-AS1(config-router)# address-family ipv4
PE1-AS1(config-router-af)# neighbor 3.3.3.3 activate
PE1-AS1(config-router-af)# neighbor 3.3.3.3 send-label
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router-af)# exit
```

ASBR1 establishes IBGP neighbor relationship with PE1-AS1.

```
ASBR1(config)# router bgp 1
ASBR1(config-router)# neighbor 1.1.1.1 remote-as 1
ASBR1(config-router)# neighbor 1.1.1.1 update-source loopback 0
ASBR1(config-router)# address-family ipv4
ASBR1(config-router-af)# neighbor 1.1.1.1 send-label
ASBR1(config-router-af)# neighbor 1.1.1.1 route-map external-mpls-route out
ASBR1(config-router-af)# network 1.1.1.1 mask 255.255.255.255
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
```

PE-ASs establish EBGP neighbor relationship with each other.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# neighbor 5.5.5.5 remote-as 2
PE1-AS1(config-router)# neighbor 5.5.5.5 update-source loopback 0
PE1-AS1(config-router)# neighbor 5.5.5.5 ebgp-multihop
PE1-AS1(config-router)# address-family ipv4
PE1-AS1(config-router-af)# no neighbor 5.5.5.5 activate
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# address-family vpnv4 unicast
PE1-AS1(config-router-af)# neighbor 5.5.5.5 activate
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

5. Verification

(1) After the configuration is completed, verify that devices at different VPN sites can communicate with each other.

PE1-AS1 verification result

```
PE1-AS1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C   1.1.1.1/32 is directly connected, Loopback 0, 00:00:44
O   2.2.2.2/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 01:04:47
O   3.3.3.3/32 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 01:03:26
O E2 5.5.5.5/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 00:50:44
O   10.10.10.0/24 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 01:04:58
C   20.20.20.0/24 is directly connected, GigabitEthernet 0/1, 00:00:44
L   20.20.20.1/32 is directly connected, GigabitEthernet 0/1, 00:00:44

PE1-AS1# ping 5.5.5.5
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS1# ping 5.5.5.5 source 1.1.1.1
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1# ping 4.4.4.4
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS1# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1#ping 2.2.2.2
Sending 5, 100-byte ICMP Echoes to 2.2.2.2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

PE1-AS2 verification result

```
PE1-AS1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    1.1.1.1/32 is directly connected, Loopback 0, 00:03:26
O    2.2.2.2/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 01:04:47
O    3.3.3.3/32 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 01:03:26
O E2 5.5.5.5/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 00:50:44
O    10.10.10.0/24 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 01:04:58
C    20.20.20.0/24 is directly connected, GigabitEthernet 0/1, 00:03:26
L    20.20.20.1/32 is directly connected, GigabitEthernet 0/1, 00:03:26

PE1-AS1# ping 5.5.5.5
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS1# ping 5.5.5.5 source 1.1.1.1
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1# ping 4.4.4.4
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS1# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1#ping 2.2.2.2
Sending 5, 100-byte ICMP Echoes to 2.2.2.2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1# show ip route

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default
```

```
Gateway of last resort is no set
C    1.1.1.1/32 is directly connected, Loopback 0, 00:04:47
O    2.2.2.2/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 01:04:47
O    3.3.3.3/32 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 01:03:26
O E2 5.5.5.5/32 [110/1] via 20.20.20.2, GigabitEthernet 0/1, 00:50:44
O    10.10.10.0/24 [110/2] via 20.20.20.2, GigabitEthernet 0/1, 01:04:58
C    20.20.20.0/24 is directly connected, GigabitEthernet 0/1, 00:04:47
L    20.20.20.1/32 is directly connected, GigabitEthernet 0/1, 00:04:47

Gateway of last resort is no set
C    1.1.1.1/32 is local host.
O    2.2.2.2/32 [110/1] via 20.20.20.2, 01:04:47, GigabitEthernet 0/1
O    3.3.3.3/32 [110/2] via 20.20.20.2, 01:03:26, GigabitEthernet 0/1
O E2 5.5.5.5/32 [110/1] via 20.20.20.2, 00:50:44, GigabitEthernet 0/1
O    10.10.10.0/24 [110/2] via 20.20.20.2, 01:04:58, GigabitEthernet 0/1
C    20.20.20.0/24 is directly connected, GigabitEthernet 0/1
C    20.20.20.1/32 is local host.

PE1-AS1# ping 5.5.5.5
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS1# ping 5.5.5.5 source 1.1.1.1
Sending 5, 100-byte ICMP Echoes to 5.5.5.5, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1# ping 4.4.4.4
Sending 5, 100-byte ICMP Echoes to 4.4.4.4, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
PE1-AS1# ping 3.3.3.3
Sending 5, 100-byte ICMP Echoes to 3.3.3.3, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
PE1-AS1#ping 2.2.2.2
Sending 5, 100-byte ICMP Echoes to 2.2.2.2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

6. Configuration Files

CE1-VPN1 configuration file

```
hostname CE1-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.16.1 255.255.255.0
```

```
!
interface Loopback 0
 ip address 10.10.10.10 255.255.255.255
!
router bgp 65001
 neighbor 192.168.16.2 remote-as 1
 !
 address-family ipv4
  network 10.10.10.10 mask 255.255.255.255
  neighbor 192.168.16.2 activate
 exit-address-family
!
```

PE1-AS1 configuration file

```
hostname PE1-AS1
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip address 20.20.20.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.16.2 255.255.255.0
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 1
 neighbor 3.3.3.3 remote-as 1
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 5.5.5.5 remote-as 2
 neighbor 5.5.5.5 ebgp-multihop 255
 neighbor 5.5.5.5 update-source Loopback 0
 !
 address-family ipv4
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-label
  no neighbor 5.5.5.5 activate
```

```
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 5.5.5.5 activate
  neighbor 5.5.5.5 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.16.1 remote-as 65002
  neighbor 192.168.16.1 activate
 exit-address-family
!
router ospf 10
 network 1.1.1.1 0.0.0.0 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

P1-AS1 configuration file

```
hostname P1-AS1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.10.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.20.20.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router ospf 10
 network 2.2.2.2 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
```

```
!
```

ASBR1 configuration file

```
hostname ASBR1
!
mpls enable
!
route-map internal-mpls-route permit 10
 match ip address 101
 set mpls-label
!
route-map external-mpls-route permit 10
 match ip address 102
 set mpls-label
!
ip access-list extended 101
 10 permit ip host 1.1.1.1 any
!
ip access-list extended 102
 10 permit ip host 5.5.5.5 any
!
interface GigabitEthernet 0/1
 no switchport
 ip address 30.30.30.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 10.10.10.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 neighbor 30.30.30.2 remote-as 2
 !
 address-family ipv4
  network 1.1.1.1 mask 255.255.255.255
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-label
  neighbor 1.1.1.1 route-map external-mpls-route out
  neighbor 30.30.30.2 activate
```

```
  neighbor 30.30.30.2 send-label
  neighbor 30.30.30.2 route-map internal-mpls-route out
 exit-address-family
!
router ospf 10
 network 3.3.3.3 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

ASBR2 configuration file

```
hostname ASBR2
!
mpls enable
!
route-map internal-mpls-route permit 10
 match ip address 101
 set mpls-label
!
route-map external-mpls-route permit 10
 match ip address 102
 set mpls-label
!
ip access-list extended 101
 10 permit ip host 5.5.5.5 any
!
ip access-list extended 102
 10 permit ip host 1.1.1.1 any
!
interface GigabitEthernet 0/1
 no switchport
 ip address 50.50.50.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 30.30.30.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router bgp 2
```

```
 neighbor 5.5.5.5 remote-as 2
 neighbor 5.5.5.5 update-source Loopback 0
 neighbor 30.30.30.1 remote-as 1
 !
 address-family ipv4
  network 5.5.5.5 mask 255.255.255.255
  neighbor 5.5.5.5 activate
  neighbor 5.5.5.5 send-label
  neighbor 5.5.5.5 route-map external-mpls-route out
  neighbor 30.30.30.1 activate
  neighbor 30.30.30.1 send-label
  neighbor 30.30.30.1 route-map internal-mpls-route out
 exit-address-family
!
router ospf 10
 network 4.4.4.4 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE1-AS2 configuration file

```
hostname PE1-AS2
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.15.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 50.50.50.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router bgp 2
 neighbor 1.1.1.1 remote-as 1
```

```
 neighbor 1.1.1.1 ebgp-multihop 255
 neighbor 1.1.1.1 update-source Loopback 0
 neighbor 4.4.4.4 remote-as 2
 neighbor 4.4.4.4 update-source Loopback 0
 !
 address-family ipv4
  no neighbor 1.1.1.1 activate
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-label
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.15.1 remote-as 65002
  neighbor 192.168.15.1 activate
 exit-address-family
!
router ospf 10
 network 5.5.5.5 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

CE2-VPN1 configuration file

```
hostname CE2-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.15.1 255.255.255.0
!
interface Loopback 0
 ip address 20.20.20.20 255.255.255.255
!
router bgp 65002
 neighbor 192.168.15.2 remote-as 2
 !
 address-family ipv4
  network 20.20.20.20 mask 255.255.255.255
  neighbor 192.168.15.2 activate
 exit-address-family
!
```

7. Common Errors

- When a multi-hop MP-EBGP session is configured, IPv4 route exchange is not disabled for the multi-hop EBGP session.
- Label switching for IPv4 routes is not configured between IBGP neighbors. As a result, VPN sites cannot interconnect with each other.

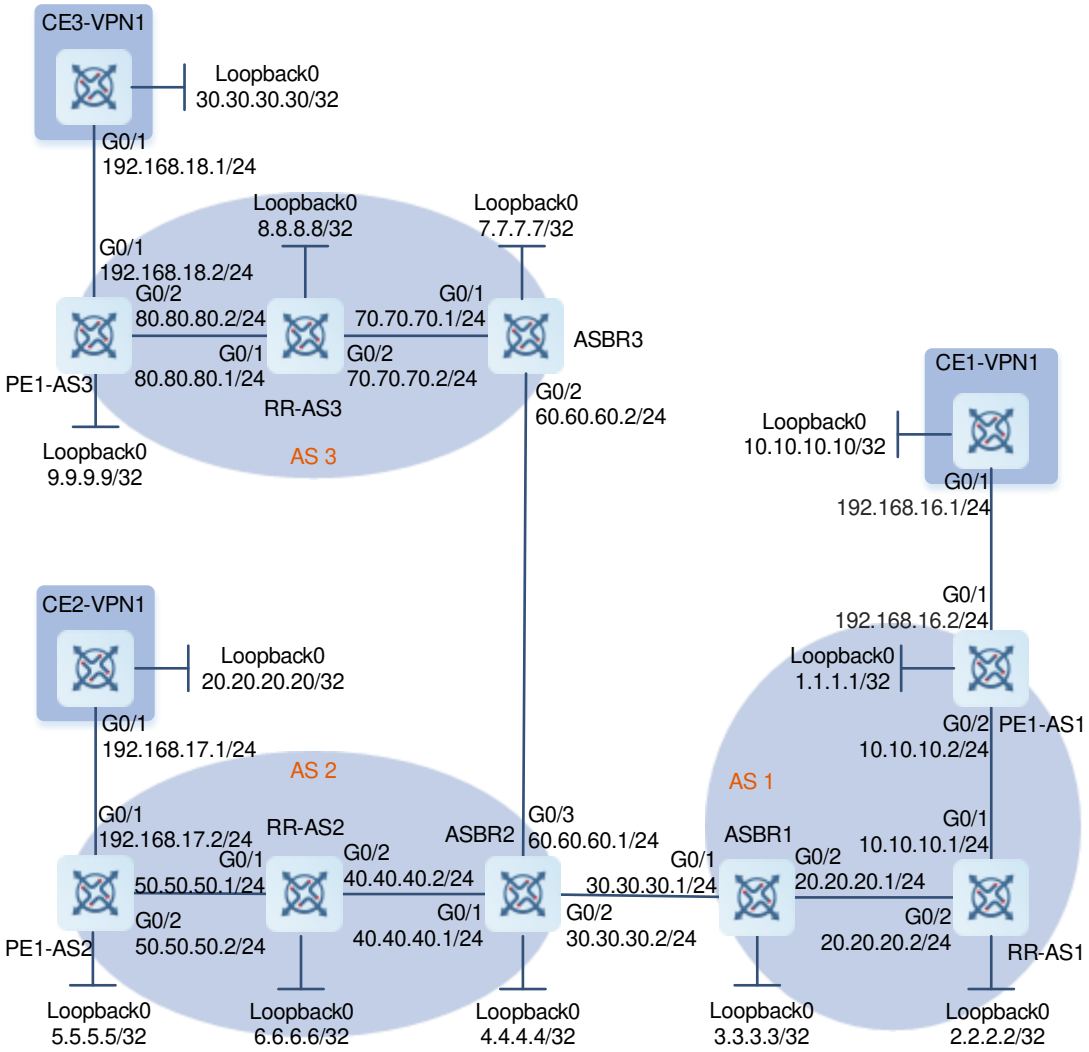
1.16.14 Configuring Inter-AS VPN Service Model – Option C (RR Deployment)

1. Requirements

Two Option C implementation solutions are provided. If sites of a VPN are deployed in different ASs, the common Option C implementation solution is adopted. As shown in Figure 1-32, sites of a VPN are deployed in three different ASs. To ensure reachability of these VPN sites, BGP connections need to be established between inter-AS PEs. Each time a new VPN site is added, the new VPN site needs to establish BGP connections with other VPN sites, which restricts the application of the common Option C solution. To solve the preceding extensibility problem, you can deploy an RR in each AS and establish multi-hop MP-EBGP connections between RRs to exchange inter-AS VPN routes. In addition, establish MP-IBGP sessions between PEs and RRs in the same AS.

2. Topology

Figure 1-32 Configuring Inter-AS VPN Service Model – Option C (RR Deployment)



3. Notes

- On PE1-AS1, configure a loopback interface, create VRF instance VPN1, define RD and RT values, and associate the VRF instance with the corresponding interface. Establish an MP-IBGP session with the RR, enable label switching for IPv4 routes, and configure CE neighbors using EBGP. Configurations on PE1-AS2 and PE1-AS3 are similar to that on PE1-AS1.
- On RR-AS1, configure a loopback interface, establish an MP-IBGP session with PE1-AS1, specify PE1-AS1 as an RR client, and enable label switching for IPv4 routes. Establish MP-EBGP sessions with other RRs and do not change the next hop of VPN routes exchanged with the RRs, disable IPv4 route exchange with the RRs, and establish an IBGP session with the ASBR1. Enable label switching for IPv4 routes, configure MPLS, and run OSPF in the backbone network to transmit routing information. Configurations on RR-AS2 and RR-AS3 are similar to that on RR-AS1.
- On ASBR1, configure a loopback interface and configure an ACL rule and a route-map rule. Establish EBGP sessions with ASBR2, enable label switching for IPv4 routes, and configure a route-map rule to configure labels for PE routes that meet conditions. The route-map rule is optional, and application of a route-map rule

enables the BGP to distribute labels only to specific routes. Establish IBGP sessions with RRs, enable label switching for IPv4 routes, configure a route-map rule to configure labels for inter-AS PE routes that meet conditions. Statically configure PE routes in the local AS. Configure MPLS signaling, enable MPLS on an interface, and run OSPF on the backbone network to transmit routing information. Configure an IP address for the interface used to connect to other ASBRs and enable labeled packet forwarding on the interface. Configurations on ASBR2 and ASBR3 are similar to that on ASBR1.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PE-ASs, RR-ASs, and ASBRs are similar. The following shows how to configure OSPF neighbors on PE1-AS1.

```
PE1-AS1> enable
PE1-AS1# configure terminal
PE1-AS1(config)# router ospf 10
PE1-AS1(config-router)# router-id 1.1.1.1
PE1-AS1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1-AS1(config-router)# network 10.10.10.0 0.0.0.255 area 0
PE1-AS1(config-router)# exit
```

(3) Configure basic MPLS functions.

Configurations on PE-ASs, RR-ASs, and ASBRs are similar. The following shows how to configure basic MPLS functions on PE1-AS1.

```
PE1-AS1(config)# mpls enable
PE1-AS1(config)# mpls router ldp
PE1-AS1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1-AS1(config-mpls-router)# exit
PE1-AS1(config)# interface gigabitethernet 0/2
PE1-AS1(config-if-GigabitEthernet 0/2)# label-switching
PE1-AS1(config-if-GigabitEthernet 0/2)# mpls ldp enable
PE1-AS1(config-if-GigabitEthernet 0/2)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following shows how to create a VPN on PE1-AS1.

```
PE1-AS1(config)# ip vrf VPN1
PE1-AS1(config-vrf)# rd 1:100
PE1-AS1(config-vrf)# route-target both 1:100
PE1-AS1(config-vrf)# exit
PE1-AS1(config)# interface gigabitethernet 0/1
PE1-AS1(config-if-GigabitEthernet 0/1)# ip vrf forwarding VPN1
PE1-AS1(config-if-GigabitEthernet 0/1)# ip address 192.168.16.2 255.255.255.0
PE1-AS1(config-if-GigabitEthernet 0/1)# exit
```

(5) Configure BGP neighbors to advertise VPN routes.

PE1-AS1 establishes EBGP neighbor relationship with CE1-VPN1.

```
PE1-AS1> enable
```

```
PE1-AS1# configure terminal
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# address-family ipv4 vrf VPN1
PE1-AS1(config-router-af)# neighbor 192.168.16.1 remote-as 65001
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

PE1-AS1 establishes IBGP neighbor relationship with RR-AS1.

```
PE1-AS1(config)# router bgp 1
PE1-AS1(config-router)# neighbor 2.2.2.2 remote-as 1
PE1-AS1(config-router)# neighbor 2.2.2.2 update-source loopback 0
PE1-AS1(config-router)# address-family vpnv4 unicast
PE1-AS1(config-router-af)# neighbor 2.2.2.2 activate
PE1-AS1(config-router-af)# exit
PE1-AS1(config-router)# address-family ipv4
PE1-AS1(config-router-af)# neighbor 2.2.2.2 activate
PE1-AS1(config-router-af)# neighbor 2.2.2.2 send-label
PE1-AS1(config-router-af)# exit-address-family
PE1-AS1(config-router)# exit
```

RR-AS1 establishes IBGP neighbor relationship with ASBR1.

```
RR-AS1> enable
RR-AS1# configure terminal
RR-AS1(config)# router bgp 1
RR-AS1(config-router)# neighbor 3.3.3.3 remote-as 1
RR-AS1(config-router)# neighbor 3.3.3.3 update-source loopback 0
RR-AS1(config-router)# address-family ipv4
RR-AS1(config-router-af)# neighbor 3.3.3.3 activate
RR-AS1(config-router-af)# neighbor 3.3.3.3 send-label
RR-AS1(config-router-af)# exit-address-family
RR-AS1(config-router)# exit
```

RR-ASs establish EBGP neighbor relationship with each other.

```
RR-AS1(config)# router bgp 1
RR-AS1(config-router)# neighbor 6.6.6.6 remote-as 2
RR-AS1(config-router)# neighbor 6.6.6.6 update-source loopback 0
RR-AS1(config-router)# neighbor 6.6.6.6 ebgp-multihop
RR-AS1(config-router)# neighbor 8.8.8.8 remote-as 3
RR-AS1(config-router)# neighbor 8.8.8.8 update-source loopback 0
RR-AS1(config-router)# neighbor 8.8.8.8 ebgp-multihop
RR-AS1(config-router)# address-family ipv4
RR-AS1(config-router-af)# no neighbor 6.6.6.6 activate
RR-AS1(config-router-af)# no neighbor 8.8.8.8 activate
RR-AS1(config-router-af)# exit-address-family
RR-AS1(config-router)# address-family vpnv4 unicast
RR-AS1(config-router-af)# neighbor 6.6.6.6 activate
RR-AS1(config-router-af)# neighbor 6.6.6.6 next-hop-unchanged
RR-AS1(config-router-af)# neighbor 8.8.8.8 activate
```

```
RR-AS1(config-router-af)# neighbor 8.8.8.8 next-hop-unchanged
RR-AS1(config-router-af)# exit-address-family
RR-AS1(config-router)# exit
```

ASBRs establish EBGP neighbor relationship with each other.

```
ASBR1(config)# router bgp 1
ASBR1(config-router)# neighbor 30.30.30.2 remote-as 2
ASBR1(config-router)# address-family ipv4
ASBR1(config-router-af)# neighbor 30.30.30.2 send-label
ASBR1(config-router-af)# network 1.1.1.1 mask 255.255.255.255
ASBR1(config-router-af)# exit-address-family
ASBR1(config-router)# exit
```

5. Verification

(1) Run the **show bgp vpnv4 unicast all** command on RR-AS1, RR-AS2, and RR-AS3 to verify that entries of 11.11.11.11 and 22.22.22.22 exist.

```
RR-AS1#show bgp vpnv4 unicast all
BGP table version is 40, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
            S Stale, b - backup entry
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network         Next Hop          Metric     LocPrf     Weight Path
Route Distinguisher: 100:1
*>i11.11.11.11/32   1.1.1.1               0        100          0 65001 i
*> 22.22.22.22/32   5.5.5.5               0                     0 2 65002 i
```

Total number of prefixes 2

(2) Different sites of the same VPN can interconnect with each other.

PE1-AS1 verification result

```
Assume that the loopback interfaces of the peer CEs are 20.20.20.20 and
30.30.30.30.
The local CE can ping 20.20.20.20.
The local CE can ping 30.30.30.30.
Ping operation to 5.5.5.5 fails.
Ping operation to 5.5.5.5 source 1.1.1.1 is successful.
Ping operation to 9.9.9.9 fails.
Ping operation to 9.9.9.9 source 1.1.1.1 is successful.
```

PE1-AS2 verification result

```
Assume that the loopback interfaces of the peer CEs are 10.10.10.10 and
30.30.30.30.
The local CE can ping 10.10.10.10
The local CE can ping 30.30.30.30.
Ping operation to 1.1.1.1 fails.
Ping operation to 1.1.1.1 source 5.5.5.5 is successful.
```

```
Ping operation to 9.9.9.9 fails.
Ping operation to 9.9.9.9 source 5.5.5.5 is successful.
```

PE1-AS3 verification result

```
Assume that the loopback interfaces of the peer CEs are 10.10.10.10 and
20.20.20.20.
The local CE can ping 10.10.10.10
The local CE can ping 20.20.20.20.
Ping operation to 1.1.1.1 fails.
Ping operation to 1.1.1.1 source 9.9.9.9 is successful.
Ping operation to 5.5.5.5 fails.
Ping operation to 5.5.5.5 source 9.9.9.9 is successful.
```

6. Configuration Files

CE1-VPN1 configuration file

```
hostname CE1-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.16.1 255.255.255.0
!
interface Loopback 0
 ip address 11.11.11.11 255.255.255.255
!
router bgp 65001
 neighbor 192.168.16.2 remote-as 1
 !
 address-family ipv4
  network 11.11.11.11 mask 255.255.255.255
  neighbor 192.168.16.2 activate
 exit-address-family
!
```

CE2-VPN1 configuration file

```
hostname CE2-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.17.1 255.255.255.0
!
interface Loopback 0
 ip address 22.22.22.22 255.255.255.255
!
router bgp 65002
 neighbor 192.168.17.2 remote-as 1
 !
 address-family ipv4
```

```
 network 22.22.22.22 mask 255.255.255.255
 neighbor 192.168.17.2 activate
exit-address-family
!
```

CE3-VPN1 configuration file

```
hostname CE3-VPN1
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.18.1 255.255.255.0
!
interface Loopback 0
 ip address 33.33.33.33 255.255.255.255
!
router bgp 65003
 neighbor 192.168.18.2 remote-as 1
 !
 address-family ipv4
  network 33.33.33.33 mask 255.255.255.255
  neighbor 192.168.18.2 activate
 exit-address-family
!
```

PE1-AS1 configuration file

```
hostname PE1-AS1
!
mpls enable
!
ip vrf VPN1
 rd 100:1
 route-target both 100:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.16.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 10.10.10.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
```

```
router bgp 1
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 update-source Loopback 0
 !
 address-family ipv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-label
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 2.2.2.2 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.16.1 remote-as 65001
  neighbor 192.168.16.1 activate
 exit-address-family
!
router ospf 10
 router-id 1.1.1.1
 network 1.1.1.1 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE1-AS2 configuration file

```
hostname PE1-AS2
!
mpls enable
!
ip vrf VPN1
 rd 100:1
 route-target both 100:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.17.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 50.50.50.2 255.255.255.0
 label-switching
 mpls ldp enable
!
```

```
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router bgp 2
 neighbor 6.6.6.6 remote-as 2
 neighbor 6.6.6.6 update-source Loopback 0
 !
 address-family ipv4
  neighbor 6.6.6.6 activate
  neighbor 6.6.6.6 send-label
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 6.6.6.6 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.17.1 remote-as 65002
  neighbor 192.168.17.1 activate
 exit-address-family
!
router ospf 10
 router-id 5.5.5.5
 network 5.5.5.5 0.0.0.0 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE1-AS3 configuration file

```
hostname PE1-AS3
!
mpls enable
!
ip vrf VPN1
 rd 100:1
 route-target both 100:1
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPN1
 ip address 192.168.18.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 80.80.80.2 255.255.255.0
```

```
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 9.9.9.9 255.255.255.255
!
router bgp 3
 neighbor 8.8.8.8 remote-as 3
 neighbor 8.8.8.8 update-source Loopback 0
 !
 address-family ipv4
  neighbor 8.8.8.8 activate
  neighbor 8.8.8.8 send-label
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 8.8.8.8 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  neighbor 192.168.18.1 remote-as 65003
  neighbor 192.168.18.1 activate
 exit-address-family
!
router ospf 10
 router-id 9.9.9.9
 network 9.9.9.9 0.0.0.0 area 0
 network 80.80.80.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

RR-AS1 configuration file

```
hostname RR-AS1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.10.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.20.20.2 255.255.255.0
```

```
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback 0
 neighbor 3.3.3.3 remote-as 1
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 6.6.6.6 remote-as 2
 neighbor 6.6.6.6 ebgp-multihop
 neighbor 6.6.6.6 update-source Loopback 0
neighbor 8.8.8.8 remote-as 3
 neighbor 8.8.8.8 ebgp-multihop
 neighbor 8.8.8.8 update-source Loopback 0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 route-reflector-client
  neighbor 1.1.1.1 send-label
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-label
  no neighbor 6.6.6.6 activate
  no neighbor 8.8.8.8 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 route-reflector-client
  neighbor 6.6.6.6 activate
  neighbor 6.6.6.6 next-hop-unchanged
  neighbor 8.8.8.8 activate
  neighbor 8.8.8.8 next-hop-unchanged
 exit-address-family
!
router ospf 10
 router-id 2.2.2.2
 network 2.2.2.2 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
 network 20.20.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

RR-AS2 configuration file

```
hostname RR-AS2
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 50.50.50.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 40.40.40.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 6.6.6.6 255.255.255.255
!
router bgp 2
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 ebgp-multihop
 neighbor 2.2.2.2 update-source Loopback 0
 neighbor 4.4.4.4 remote-as 2
 neighbor 4.4.4.4 update-source Loopback 0
 neighbor 5.5.5.5 remote-as 2
 neighbor 5.5.5.5 update-source Loopback 0
 neighbor 8.8.8.8 remote-as 3
 neighbor 8.8.8.8 ebgp-multihop
 neighbor 8.8.8.8 update-source Loopback 0
 !
 address-family ipv4
  no neighbor 2.2.2.2 activate
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-label
  neighbor 5.5.5.5 activate
  neighbor 5.5.5.5 route-reflector-client
  neighbor 5.5.5.5 send-label
  no neighbor 8.8.8.8 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 next-hop-unchanged
  neighbor 5.5.5.5 activate
```

```
 neighbor 5.5.5.5 route-reflector-client
 neighbor 8.8.8.8 activate
 neighbor 8.8.8.8 next-hop-unchanged
exit-address-family
!
router ospf 10
 router-id 6.6.6.6
 network 6.6.6.6 0.0.0.0 area 0
 network 40.40.40.0 0.0.0.255 area 0
 network 50.50.50.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

RR-AS3 configuration file

```
hostname RR-AS3
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 80.80.80.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 70.70.70.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 8.8.8.8 255.255.255.255
!
router bgp 3
neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 ebgp-multihop
 neighbor 2.2.2.2 update-source Loopback 0
neighbor 6.6.6.6 remote-as 2
 neighbor 6.6.6.6 ebgp-multihop
 neighbor 6.6.6.6 update-source Loopback 0
 neighbor 7.7.7.7 remote-as 3
 neighbor 7.7.7.7 update-source Loopback 0
 neighbor 9.9.9.9 remote-as 3
 neighbor 9.9.9.9 update-source Loopback 0
 !
```

```
 address-family ipv4
  no neighbor 2.2.2.2 activate
  no neighbor 6.6.6.6 activate
  neighbor 7.7.7.7 activate
  neighbor 7.7.7.7 send-label
  neighbor 9.9.9.9 activate
  neighbor 9.9.9.9 route-reflector-client
  neighbor 9.9.9.9 send-label
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 next-hop-unchanged
  neighbor 6.6.6.6 activate
  neighbor 6.6.6.6 next-hop-unchanged
  neighbor 9.9.9.9 activate
  neighbor 9.9.9.9 route-reflector-client
 exit-address-family
!
router ospf 10
 router-id 8.8.8.8
 network 8.8.8.8 0.0.0.0 area 0
 network 70.70.70.0 0.0.0.255 area 0
 network 80.80.80.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

ASBR1 configuration file

```
hostname ASBR1
!
mpls enable
!
route-map internal-mpls-route permit 10
 match ip address 101
 set mpls-label
!
route-map external-mpls-route permit 10
 match ip address 102
 set mpls-label
!
ip access-list extended 101
 10 permit ip host 1.1.1.1 any
!
!
ip access-list extended 102
```

```
 10 permit ip host 5.5.5.5 any
 20 permit ip host 9.9.9.9 any
!
!
interface GigabitEthernet 0/1
 no switchport
 ip address 30.30.30.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 20.20.20.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 1
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 update-source Loopback 0
 neighbor 30.30.30.2 remote-as 2
 !
 address-family ipv4
  network 1.1.1.1 mask 255.255.255.255
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-label
  neighbor 2.2.2.2 route-map external-mpls-route out
  neighbor 30.30.30.2 activate
  neighbor 30.30.30.2 send-label
  neighbor 30.30.30.2 route-map internal-mpls-route out
 exit-address-family
!
router ospf 10
 router-id 3.3.3.3
 redistribute connected
 network 3.3.3.3 0.0.0.0 area 0
 network 20.20.20.0 0.0.0.255 area 0
 network 30.30.30.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0
!
```

ASBR2 configuration file

```
hostname ASBR2
```

```
!
mpls enable
!
route-map internal-mpls-route permit 10
 match ip address 101
 set mpls-label
!
route-map external-mpls-route permit 10
 match ip address 102
 set mpls-label
!
ip access-list extended 101
 10 permit ip host 5.5.5.5 any
!
!
ip access-list extended 102
 10 permit ip host 1.1.1.1 any
 20 permit ip host 9.9.9.9 any
!
interface GigabitEthernet 0/1
 no switchport
 ip address 40.40.40.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 30.30.30.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/3
 no switchport
 ip address 60.60.60.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router bgp 2
 bgp log-neighbor-changes
 neighbor 6.6.6.6 remote-as 2
 neighbor 6.6.6.6 update-source Loopback 0
 neighbor 30.30.30.1 remote-as 1
 neighbor 60.60.60.2 remote-as 3
```

```
 !
 address-family ipv4
  network 5.5.5.5 mask 255.255.255.255
  neighbor 6.6.6.6 activate
  neighbor 6.6.6.6 send-label
  neighbor 6.6.6.6 route-map external-mpls-route out
  neighbor 30.30.30.1 activate
  neighbor 30.30.30.1 send-label
  neighbor 30.30.30.1 route-map internal-mpls-route out
  neighbor 60.60.60.2 activate
  neighbor 60.60.60.2 send-label
  neighbor 60.60.60.2 route-map internal-mpls-route out
 exit-address-family
!
router ospf 10
 router-id 4.4.4.4
 network 4.4.4.4 0.0.0.0 area 0
 network 30.30.30.0 0.0.0.255 area 0
 network 40.40.40.0 0.0.0.255 area 0
 network 60.60.60.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

ASBR3 configuration file

```
hostname ASBR3
!
mpls enable
!
route-map internal-mpls-route permit 10
 match ip address 101
 set mpls-label
!
route-map external-mpls-route permit 10
 match ip address 102
 set mpls-label
!
ip access-list extended 101
 10 permit ip host 9.9.9.9 any
!
!
ip access-list extended 102
 10 permit ip host 1.1.1.1 any
 20 permit ip host 5.5.5.5 any
!
!
```

```
interface GigabitEthernet 0/1
 no switchport
 ip address 70.70.70.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 60.60.60.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 7.7.7.7 255.255.255.255
!
router bgp 3
 neighbor 8.8.8.8 remote-as 3
 neighbor 8.8.8.8 update-source Loopback 0
 neighbor 60.60.60.1 remote-as 2
 !
 address-family ipv4
  network 9.9.9.9 mask 255.255.255.255
  neighbor 8.8.8.8 activate
  neighbor 8.8.8.8 send-label
  neighbor 8.8.8.8 route-map external-mpls-route out
  neighbor 60.60.60.1 activate
  neighbor 60.60.60.1 send-label
  neighbor 60.60.60.1 route-map internal-mpls-route out
 exit-address-family
!
router ospf 10
 router-id 7.7.7.7
 redistribute connected
 network 7.7.7.7 0.0.0.0 area 0
 network 60.60.60.0 0.0.0.255 area 0
 network 70.70.70.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0
!
```

7. Common Errors

- When a multi-hop MP-EBGP session is configured between RRs, IPv4 route exchange is not disabled for the multi-hop MP-EBGP session.
- When VPN routes are exchanged through a multi-hop MP-EBGP session, next hop unchanged is not configured. As a result, VPN sites cannot interconnect with each other.

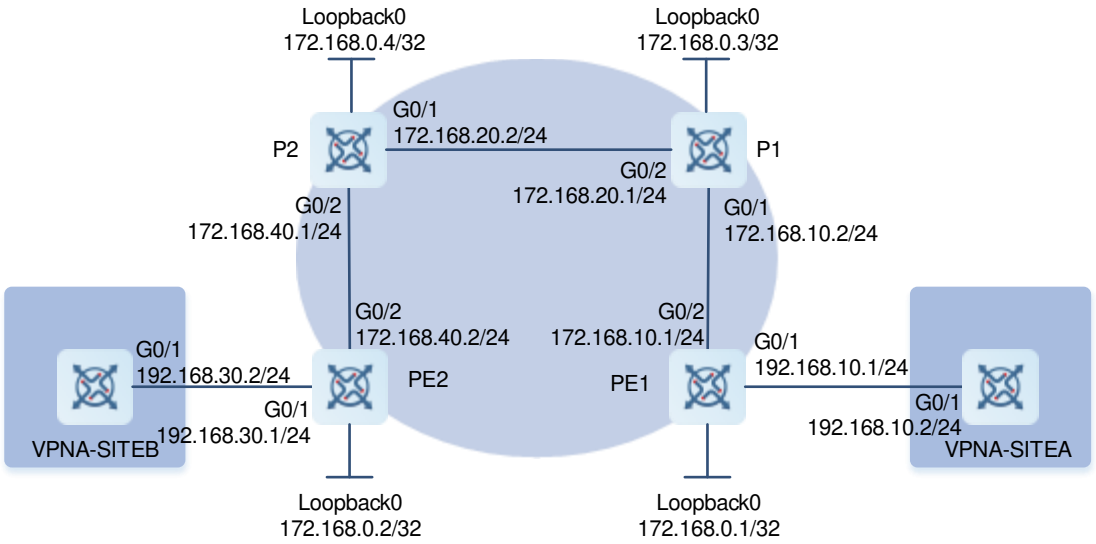
1.16.15 Configuring OSPF VPN Extended Features (Domain ID)

1. Requirements

Two different sites of a customer exchange VPN routes through the MPLS backbone network, and these sites access PEs using OSPF. It is required that after OSPF routing information is exchanged through the MPLS backbone network, OSPF routes of the original sites can be restored to the maximum.

2. Topology

Figure 1-33 Configuring OSPF VPN Extended Features (Domain ID)



3. Notes

- Configure OSPF between PEs and CEs on SiteA and SiteB.
- On PE1, configure a loopback interface, create VRF instance VPNA, define RD and RT values, and associate the VRF instance with the interface connected to CE1. Configure the BGP, establish an MP-IBGP session with PE2, exchange routes with the CE using OSPF, and set the domain ID of the OSPF process to 10. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol. Configurations on PE2 are similar to that on PE1.
- Configure the backbone network MPLS signaling on P1 and P2, enable MPLS on the interface, and configure the backbone network routing protocol.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on all devices are similar. The following shows how to configure OSPF neighbors on a PE.

```
PE1> enable
PE1# configure terminal
PE1(config)# router ospf 1
PE1(config-router)# network 172.168.10.0 0.0.0.255 area 0
```

```
PE1(config-router)# network 172.168.0.1 0.0.0.0 area 0
PE1(config-router)# exit
```

(3) Configure basic MPLS functions.

Configurations on PEs and Ps are similar. The following shows how to configure basic MPLS functions on PE1.

```
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-GigabitEthernet 0/2)# ip address 172.168.10.1 255.255.255.0
PE1(config-GigabitEthernet 0/2)# label-switching
PE1(config-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-GigabitEthernet 0/2)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following uses PE1 as an example.

```
PE1(config)# ip vrf VPNA
PE1(config-vrf)# rd 1:100
PE1(config-vrf)# route-target both 1:100
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-GigabitEthernet 0/1)# ip vrf forwarding VPNA
PE1(config-GigabitEthernet 0/1)# ip address 192.168.10.1 255.255.255.0
PE1(config-GigabitEthernet 0/1)# exit
```

(5) Configure VPN routes.

Configurations on PE1 and PE2 are similar. The following shows how to configure VPN routes on PE1.

```
PE1(config)# router ospf 10 vrf VPNA
PE1(config-router)# network 192.168.10.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# domain-id 10.10.10.10
PE1(config-router)# exit
```

(6) Configure BGP neighbors to advertise VPN routes.

A PE configures IBGP neighbors to advertise VPN routes.

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 172.168.0.2 remote-as 1
PE1(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 172.168.0.2 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPNA
PE1(config-router-af)# redistribute ospf 10
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

5. Verification

(1) After the configuration is completed, run the **show ip route** command to display routes of VPNA SiteA and VPNA SiteB.

VPNA SiteB verification result

```
VPNA-SITEB# show ip route
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
O IA    192.168.10.0/24 [110/2] via 192.168.30.1, GigabitEthernet 0/1, 00:00:36
C      192.168.30.0/24 is directly connected, GigabitEthernet 0/1, 00:00:06
```

PE2 verification result

```
PE2# show ip route vrf VPNA
Routing Table: VPNA

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
B      192.168.10.0/24 [110/2] via 172.168.0.1, 00:00:36
C      192.168.30.0/24 is directly connected, GigabitEthernet 0/3, 00:00:06
```

PE1 verification result

```
PE1# show ip route vrf VPNA
Routing Table: VPNA

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
```

```
       * - candidate default

Gateway of last resort is no set
C      192.168.10.0/24 is directly connected, GigabitEthernet 0/1, 00:00:06
B      192.168.30.0/24 [110/2] via 172.168.0.2, 00:00:36
```

VPNA SiteA verification result

```
VPNA-SITEA# show ip route
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C      192.168.10.0/24 is directly connected, GigabitEthernet 0/1, 00:00:06
O IA    192.168.30.0/24 [110/2] via 192.168.10.1, GigabitEthernet 0/1, 00:00:36
```

6. Configuration Files

VPNA SiteA configuration file

```
hostname VPNA-SITEA
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.10.2 255.255.255.0
!
router ospf 10
 network 192.168.10.0 0.0.0.255 area 0
!
```

VPNA SiteB configuration file

```
hostname VPNA-SITEB
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.30.2 255.255.255.0
!
router ospf 10
 network 192.168.30.0 0.0.0.255 area 0
!
```

PE1 configuration file

```
hostname PE1
```

```
!
mpls enable
!
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.10.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 172.168.0.1 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 !
 address-family ipv4
  neighbor 172.168.0.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 172.168.0.2 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNA
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 172.168.0.1 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
!
router ospf 10 vrf VPNA
 domain-id 10.10.10.10
 redistribute bgp subnets
 network 192.168.10.0 0.0.0.255 area 0
!
```

```
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE2 configuration file

```
hostname PE2
!
mpls enable
!
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.30.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.40.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 172.168.0.2 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 172.168.0.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNA
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 172.168.0.2 0.0.0.0 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
```

```
router ospf 10 vrf VPNA
 domain-id 10.10.10.10
 redistribute bgp subnets
 network 192.168.30.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

P1 configuration file

```
hostname P1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.10.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.20.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 172.168.0.3 255.255.255.255
!
router ospf 1
 network 172.168.0.3 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
 network 172.168.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

P2 configuration file

```
hostname P2
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.20.2 255.255.255.0
 label-switching
```

```
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.40.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 172.168.0.4 255.255.255.255
!
router ospf 1
 network 172.168.0.4 0.0.0.0 area 0
 network 172.168.20.0 0.0.0.255 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

7. Common Errors

The router ID is not 32 bits. As a result, an LDP session fails to be established.

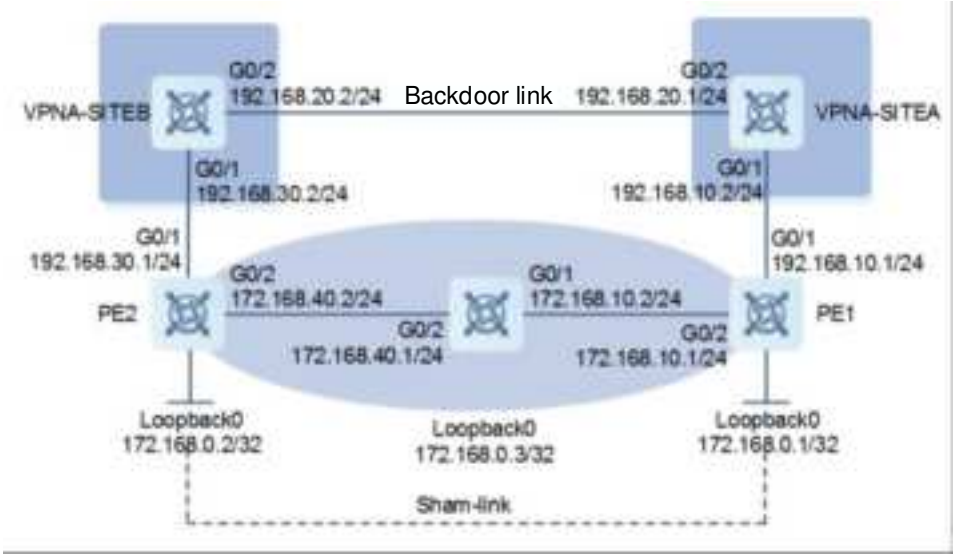
1.16.16 Configuring OSPF VPN Extended Features (Sham Link)

1. Requirements

Two sites of a customer exchange VPN routes through the MPLS backbone network. A backdoor link is established between the two sites. When the MPLS backbone network is faulty, the two sites can exchange information through the backdoor link.

2. Topology

Figure 1-34 Configuring OSPF VPN Extended Features (Sham Link)



3. Notes

- On VPNA SiteA, configure OSPF run with PE1 and VPNA SiteB. VPNA SiteA and VPNA SiteB run OSPF through the backdoor link. Configure the OSPF cost value of interface G0/2.
- On VPNA SiteB, configure OSPF run with PE2 and VPNA SiteA. VPNA SiteB and VPNA SiteA run OSPF through the backdoor link. Configure the OSPF cost value of the interfaces.
- On PE1, configure a loopback interface, create VRF instance VPNA, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure a loopback interface for the VRF instance to establish a sham link. Configure the BGP, establish an MP-IBGP session with PE2, exchange routes with VPNA SiteA using OSPF, and establish a sham link with the OSPF process on PE2. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- On PE2, configure a loopback interface, create VRF instance VPNA, define RD and RT values, and associate the VRF instance with the corresponding interface. Configure a loopback interface for the VRF instance to establish a sham link. Configure the BGP, establish an MP-IBGP session with PE1, exchange VPN routes with VPNA SiteB using OSPF, and establish a sham link with the OSPF process on PE1. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol.
- Configure the backbone network MPLS signaling on P1, enable MPLS on the interface, and configure the backbone network routing protocol.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PEs and Ps are similar. The following shows how to configure OSPF neighbors on a PE.

```
PE1> enable
PE1# configure terminal
PE1(config)# router ospf 1
PE1(config-router)# network 172.168.10.0 0.0.0.255 area 0
PE1(config-router)# network 172.168.0.1 0.0.0.0 area 0
PE1(config-router)# end
```

Configurations on VPNA SiteA and VPNA SiteB are similar. The following shows how to configure OSPF neighbors on VPNA SiteA.

```
VPNA-SITEA> enable
VPNA-SITEA# configure terminal
VPNA-SITEA(config)# router ospf 10
VPNA-SITEA(config-router)# network 192.168.10.0 255.255.255.0 area 0
VPNA-SITEA(config-router)# network 192.168.20.0 255.255.255.0 area 0
VPNA-SITEA(config-router)# exit
VPNA-SITEA(config)# interface gigabitethernet 0/1
VPNA-SITEA(config-GigabitEthernet 0/1)# ip address 192.168.10.2 255.255.255.0
VPNA-SITEA(config-GigabitEthernet 0/1)# ip ospf cost 1
```

```
VPNA-SITEA(config)# interface gigabitethernet 0/2
VPNA-SITEA(config-GigabitEthernet 0/2)# ip address 192.168.20.1 255.255.255.0
VPNA-SITEA(config-GigabitEthernet 0/2)# ip ospf cost 200
```

(3) Configure basic MPLS functions.

Configurations on PEs and Ps are similar. The following shows how to configure basic MPLS functions on PE1.

```
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-GigabitEthernet 0/2)# ip address 172.168.10.1 255.255.255.0
PE1(config-GigabitEthernet 0/2)# label-switching
PE1(config-GigabitEthernet 0/2)# mpls ldp enable
PE1(config-GigabitEthernet 0/2)# exit
```

(4) Create a VRF instance and associate it with an Ethernet interface.

Configurations on PEs are similar. The following uses PE1 as an example.

```
PE1(config)# ip vrf VPNA
PE1(config-vrf)# rd 1:100
PE1(config-vrf)# route-target both 1:100
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-GigabitEthernet 0/1)# ip vrf forwarding VPNA
PE1(config-GigabitEthernet 0/1)# ip address 192.168.10.1 255.255.255.0
PE1(config-GigabitEthernet 0/1)# exit
PE1(config)# interface loopback 10
PE1(config-Loopback 10)# ip vrf forwarding VPNA
PE1(config-Loopback 10)# ip address 192.168.0.1 255.255.255.255
PE1(config-Loopback 10)# exit
```

(5) Configure VPN routes.

Configurations on PEs are similar. The following shows how to configure VPN routes on PE1.

```
PE1(config)# router ospf 10 vrf VPNA
PE1(config-router)# network 192.168.10.0 255.255.255.0 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# area 0 sham-link 192.168.0.1 192.168.0.2
PE1(config-router)# exit
```

(6) Configure BGP neighbors to advertise VPN routes.

A PE configures IBGP neighbors to advertise VPN routes.

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 172.168.0.2 remote-as 1
PE1(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 172.168.0.2 activate
```

```
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPNA
PE1(config-router-af)# redistribute ospf 10
PE1(config-router-af)# redistribute connected
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

5. Verification

(1) After the configuration is completed, verify that an OSPF sham link exists on the PE.

PE1 verification result

```
PE1# show ip ospf 10 sham-links
Sham Link SLINK0 to address 192.168.0.2 is up
  Area 0.0.0.0 source address 192.168.0.1, Cost: 1
  Output interface is GigabitEthernet 0/2
  Nexthop address 172.16.40.2
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
    Adjacency state Full

PE1# show ip ospf 10 neighbor
OSPF process 10, 1 Neighbors, 1 is Full:
Neighbor ID     Pri   State              BFD State  Dead Time   Address
Interface
192.168.0.2       1   Full/ -            -          00:00:34    192.168.0.2
SLINK0

PE1# show ip route vrf VPNA
Routing Table: VPNA
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C    192.168.0.1/32 is directly connected, Loopback 0, 00:00:06
B    192.168.0.2/32 [200/0] via 172.168.0.2, 00:11:31
C    192.168.10.0/24 is directly connected, GigabitEthernet 0/1, 00:00:06
L    192.168.10.1/32 is directly connected, GigabitEthernet 0/1, 00:00:06
O    192.168.20.0/24 [110/201] via 192.168.10.2, GigabitEthernet 0/1, 00:16:23
O    192.168.30.0/24 [110/2] via 172.168.0.2, GigabitEthernet 0/1, 00:11:15
```

PE2 verification result

```
PE2# show ip ospf 10 sham-links
Sham Link SLINK0 to address 192.168.0.1 is up
  Area 0.0.0.0 source address 192.168.0.2, Cost: 1
  Output interface is GigabitEthernet 0/2
  Nexthop address 172.16.10.1
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
    Adjacency state Full

PE2# show ip ospf 10 neighbor
OSPF process 10, 1 Neighbors, 1 is Full:
Neighbor ID     Pri   State              BFD State  Dead Time   Address
Interface
192.168.0.1       1   Full/ -             -         00:00:34    192.168.0.1
SLINK0

PE2# show ip route vrf VPNA
Routing Table: VPNA
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
B    192.168.0.1/32 [200/0] via 172.168.0.1, 00:14:47
C    192.168.0.2/32 is directly connected, Loopback 0, 00:00:35
O    192.168.10.0/24 [110/2] via 172.168.0.1, GigabitEthernet 0/1, 00:14:35
O    192.168.20.0/24 [110/201] via 192.168.30.2, GigabitEthernet 0/1, 00:44:05
C    192.168.30.0/24 is directly connected, GigabitEthernet 0/1, 00:00:35
L    192.168.30.1/32 is directly connected, GigabitEthernet 0/1, 00:00:35
```

6. Configuration Files

VPNA SiteA configuration file

```
hostname VPNA-SITEA
!
interface GigabitEthernet 0/1
 no switchport
 ip ospf cost 1
 ip address 192.168.10.2 255.255.255.0
!
interface GigabitEthernet 0/2
```

```
 no switchport
 ip ospf cost 200
 ip address 192.168.20.1 255.255.255.0
!
router ospf 10
 network 192.168.10.0 0.0.0.255 area 0
 network 192.168.20.0 0.0.0.255 area 0
!
VPNA SiteB configuration file
hostname VPNA-SITEB
!
interface GigabitEthernet 0/1
 no switchport
 ip ospf cost 1
 ip address 192.168.30.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip ospf cost 200
 ip address 192.168.20.2 255.255.255.0
!
router ospf 10
 network 192.168.30.0 0.0.0.255 area 0
 network 192.168.20.0 0.0.0.255 area 0
!
```

PE1 configuration file

```
hostname PE1
!
mpls enable
!
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.10.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
```

```
 ip address 172.168.0.1 255.255.255.255
!
interface Loopback 10
 ip vrf forwarding VPNA
 ip address 192.168.0.1 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 !
 address-family ipv4
  neighbor 172.168.0.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 172.168.0.2 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNA
  redistribute connected
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 172.168.0.1 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
!
router ospf 10 vrf VPNA
 domain-id 10.10.10.10
 redistribute bgp subnets
 network 192.168.10.0 0.0.0.255 area 0
 area 0 sham-link 192.168.0.1 192.168.0.2
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE2 configuration file

```
hostname PE2
!
mpls enable
!
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
interface GigabitEthernet 0/1
```

```
 no switchport
 ip vrf forwarding VPNA
 ip address 192.168.30.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.40.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 172.168.0.2 255.255.255.255
!
interface Loopback 10
 ip vrf forwarding VPNA
 ip address 192.168.0.2 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 172.168.0.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNA
  redistribute connected
  redistribute ospf 10 match internal
 exit-address-family
!
router ospf 1
 network 172.168.0.2 0.0.0.0 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
router ospf 10 vrf VPNA
 domain-id 10.10.10.10
 redistribute bgp subnets
 network 192.168.30.0 0.0.0.255 area 0
 area 0 sham-link 192.168.0.2 192.168.0.1
!
mpls router ldp
 ldp router-id interface Loopback 0 force
```

```
!
```

P1 configuration file

```
hostname P1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.10.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.20.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 172.168.0.3 255.255.255.255
!
router ospf 1
 network 172.168.0.3 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
 network 172.168.20.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

P2 configuration file

```
hostname P2
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 172.168.20.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 no switchport
 ip address 172.168.40.1 255.255.255.0
 label-switching
 mpls ldp enable
```

```
!
interface Loopback 0
 ip address 172.168.0.4 255.255.255.255
!
router ospf 1
 network 172.168.0.4 0.0.0.0 area 0
 network 172.168.20.0 0.0.0.255 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

7. Common Errors

The router ID is not 32 bits. As a result, an LDP session fails to be established.

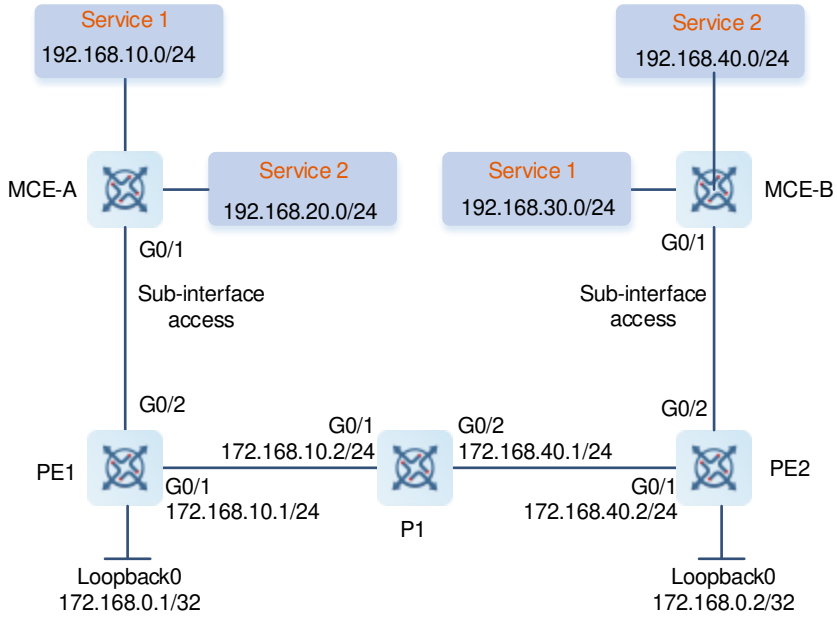
1.16.17 Configuring OSPF VPN Extended Features (Multiple OSPF Instances on the MCE)

1. Requirements

A customer site has multiple different services. Traffic for the same service can be exchanged across MPLS backbone networks, and traffic of different services is isolated.

2. Topology

Figure 1-35 Configuring OSPF VPN Extended Features (Multiple OSPF Instances on the MCE)



3. Notes

- On MCE-A, configure a trunk link between PE1 and CE1, configure two VRF instances that represent two different services and bind them to corresponding interfaces, and configure OSPF. Configurations of MCE-B

are similar to that on MCE-A.

- On PE1, configure a loopback interface, configure a trunk link between PE1 and CE1, create two VRF instances VPN1 and VPN2 that correspond to different services, associate VRF instances with the interface used to connect to CE1. Configure the BGP, establish an MP-IBGP session with PE2, and exchange routes with CE1 using OSPF. Configure the backbone network MPLS signaling, enable MPLS on the public interface, and configure the backbone network routing protocol. Configurations on PE2 are similar to that on PE1.
- Configure the backbone network MPLS signaling on P1, enable MPLS on the interface, and configure the backbone network routing protocol.

Note

For the connection between PE1 and CE1, this example uses SVI and 802.1Q configurations, which are not supported by some devices. PE1 and CE1 can be connected through any two links (physical or logical links) only if two route adjacencies are formed between them. Users can select a suitable connection method based on actual requirements.

4. Procedure

(1) Configure IP addresses for all device interfaces (omitted).

(2) Configure OSPF neighbors to ensure reachable unicast routes.

Configurations on PEs are similar. The following shows how to configure OSPF neighbors on a PE.

```
PE1(config)# router ospf 1
PE1(config-router)# network 172.168.10.0 0.0.0.255 area 0
PE1(config-router)# network 172.168.0.1 0.0.0.0 area 0
```

(3) Configure basic MPLS functions.

Configurations on PEs and Ps are similar. The following shows how to configure basic MPLS functions on PE1.

```
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface loopback 0 force
PE1(config-mpls-router)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip address 172.168.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# label-switching
PE1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/1)# exit
```

(4) Create a VRF instance and an Ethernet sub-instance and associate them.

Configurations on PEs are similar. The following uses PE1 as an example.

```
PE1(config)# ip vrf VPN1
PE1(config-vrf)# rd 1:100
PE1(config-vrf)# route-target both 1:100
PE1(config-vrf)# exit
PE1(config)# ip vrf VPN2
PE1(config-vrf)# rd 1:200
```

```
PE1(config-vrf)# route-target both 1:200
PE1(config-vrf)# exit
PE1(config)# interface gigabitethernet 0/2.1
PE1(config-if-GigabitEthernet 0/2.1)# ip vrf forwarding VPN1
PE1(config-if-GigabitEthernet 0/2.1)# ip address 192.168.10.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2.1)# exit
PE1(config)# interface gigabitethernet 0/2.2
PE1(config-if-GigabitEthernet 0/2.2)# ip vrf forwarding VPN1
PE1(config-if-GigabitEthernet 0/2.2)# ip address 192.168.20.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2.2)# exit
```

Configurations on MCEs are similar. The following shows how to create a VPN on MCE-A.

```
MCE-A> enable
MCE-A# configure terminal
MCE-A(config)# ip vrf VPN1
MCE-A(config-vrf)# exit
MCE-A(config)# ip vrf VPN2
MCE-A(config-vrf)# exit
MCE-A(config)# interface gigabitethernet 0/1.1
MCE-A(config-if-GigabitEthernet 0/1.1)# ip vrf forwarding VPN1
MCE-A(config-if-GigabitEthernet 0/1.1)# ip address 192.168.10.2 255.255.255.0
MCE-A(config-if-GigabitEthernet 0/1.1)# exit
MCE-A(config)# interface gigabitethernet 0/1.1
MCE-A(config-if-GigabitEthernet 0/1.2)# ip vrf forwarding VPN2
MCE-A(config-if-GigabitEthernet 0/1.2)# ip address 192.168.20.2 255.255.255.0
MCE-A(config-if-GigabitEthernet 0/1.2)# exit
```

(5) Configure VPN routes.

Configurations on PEs are similar. The following shows how to configure VPN routes on PE1.

```
PE1(config)# router ospf 10 vrf VPN1
PE1(config-router)# network 192.168.10.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# exit
PE1(config)# router ospf 20 vrf VPN2
PE1(config-router)# network 192.168.20.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# exit
```

Configurations on MCEs are similar. The following shows how to configure VPN routes on MCE-A.

```
MCE-A(config)# router ospf 10 vrf VPN1
MCE-A(config-router)# network 192.168.10.0 0.0.0.255 area 0
MCE-A(config-router)# capability vrf-lite
MCE-A(config-router)# exit
MCE-A(config)# router ospf 20 vrf VPN2
MCE-A(config-router)# network 192.168.20.0 0.0.0.255 area 0
MCE-A(config-router)# capability vrf-lite
```

(6) Configure BGP neighbors to advertise VPN routes.

A PE configures IBGP neighbors to advertise VPN routes.

```
PE1(config)# router bgp 1
PE1(config-router)# neighbor 172.168.0.2 remote-as 1
PE1(config-router)# neighbor 172.168.0.2 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 172.168.0.2 activate
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPN1
PE1(config-router-af)# redistribute ospf 10
PE1(config-router-af)# redistribute connected
PE1(config-router-af)# exit-address-family
PE1(config-router)# address-family ipv4 vrf VPN2
PE1(config-router-af)# redistribute ospf 20
PE1(config-router-af)# redistribute connected
PE1(config-router-af)# exit-address-family
PE1(config-router)# exit
```

5. Verification

(1) After the configuration is completed, run the **show ip route vrf** command on MCE-A and MCE-B to display the private network routing table.

MCE-A verification result

```
MCE-A# show ip route vrf VPN1
Routing Table: VPN1

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C      192.168.10.0/24 is directly connected, GigabitEthernet 0/1.1, 00:00:06
O      192.168.110.0/24 [110/101] via 192.168.21.2, GigabitEthernet 0/2,
00:56:23
O IA   192.168.130.0/24 [110/2] via 192.168.10.1, GigabitEthernet 0/1.1,
00:00:36

MCE-A# show ip route vrf VPN2
Routing Table: VPN2

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
```

```
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C      192.168.20.0/24 is directly connected, GigabitEthernet 0/1.2, 00:00:06
O      192.168.120.0/24 [110/101] via 192.168.22.2, GigabitEthernet 0/3,
00:56:23
O IA   192.168.140.0/24 [110/2] via 192.168.20.1, GigabitEthernet 0/1.2,
00:00:36
```

MCE-B verification result

```
MCE-B# show ip route vrf VPN1
Routing Table: VPN1

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
C      192.168.30.0/24 is directly connected, GigabitEthernet 0/1.3, 00:00:06
O      192.168.130.0/24 [110/101] via 192.168.23.2, GigabitEthernet 0/2,
00:56:23
O IA   192.168.110.0/24 [110/2] via 192.168.30.1, GigabitEthernet 0/1.3,
00:00:36

MCE-B# show ip route vrf VPN2
Routing Table: VPN2

Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
```

```
C      192.168.40.0/24 is directly connected, GigabitEthernet 0/1.4, 00:00:06
O      192.168.140.0/24 [110/101] via 192.168.24.2, GigabitEthernet 0/3,
00:56:23
O IA   192.168.140.0/24 [110/2] via 192.168.40.1, GigabitEthernet 0/1.4,
00:00:36
```

6. Configuration Files

MCE-A configuration file

```
hostname MCE-A
!
ip vrf VPN1
!
ip vrf VPN2
!
interface GigabitEthernet 0/1.1
 ip vrf forwarding VPN1
 ip address 192.168.10.2 255.255.255.0
!
interface GigabitEthernet 0/1.2
 ip vrf forwarding VPN2
 ip address 192.168.20.2 255.255.255.0
!
router ospf 10 vrf VPN1
 network 192.168.10.0 0.0.0.255 area 0
 capability vrf-lite
!
router ospf 20 vrf VPN2
 network 192.168.20.0 0.0.0.255 area 0
 capability vrf-lite
!
```

MCE-B configuration file

```
hostname MCE-B
!
ip vrf VPN1
!
ip vrf VPN2
!
interface GigabitEthernet 0/1.3
 ip vrf forwarding VPN1
 ip address 192.168.30.2 255.255.255.0
!
interface GigabitEthernet 0/1.4
 ip vrf forwarding VPN2
 ip address 192.168.40.2 255.255.255.0
!
router ospf 10 vrf VPN1
```

```
 network 192.168.30.0 0.0.0.255 area 0
 capability vrf-lite
!
router ospf 20 vrf VPN2
 network 192.168.40.0 0.0.0.255 area 0
 capability vrf-lite
!
```

PE1 configuration file

```
hostname PE1
!
mpls ldp enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
ip vrf VPN2
 rd 1:200
 route-target both 1:200
!
interface GigabitEthernet 0/1
 ip address 172.168.10.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2.1
 ip vrf forwarding VPN1
 ip address 192.168.10.1 255.255.255.0
!
interface GigabitEthernet 0/2.2
 ip vrf forwarding VPN2
 ip address 192.168.20.1 255.255.255.0
!
interface Loopback 0
 ip address 172.168.0.1 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 !
 address-family ipv4
  neighbor 172.168.0.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 172.168.0.2 activate
```

```
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute connected
  redistribute ospf 10 match internal
 exit-address-family
!
address-family ipv4 vrf VPN2
  redistribute connected
  redistribute ospf 20 match internal
 exit-address-family
!
router ospf 10 vrf VPN1
 network 192.168.10.0 255.255.255.0 area 0
 redistribute bgp subnets
!
router ospf 10 vrf VPN2
 network 192.168.20.0 255.255.255.0 area 0
 redistribute bgp subnets
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

PE2 configuration file

```
hostname PE2
!
mpls enable
!
ip vrf VPN1
 rd 1:100
 route-target both 1:100
!
ip vrf VPN2
 rd 1:200
 route-target both 1:200
!
interface GigabitEthernet 0/1
 ip address 172.168.40.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2.3
 ip vrf forwarding VPN1
 ip address 192.168.30.1 255.255.255.0
!
interface GigabitEthernet 0/2.4
```

```
 ip vrf forwarding VPN2
 ip address 192.168.40.1 255.255.255.0
!

interface Loopback 0
 ip address 172.168.0.2 255.255.255.255
!
router bgp 1
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 172.168.0.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  redistribute connected
  redistribute ospf 10 match internal
 exit-address-family
!
 address-family ipv4 vrf VPN2
  redistribute connected
  redistribute ospf 20 match internal
 exit-address-family
!
router ospf 1
 network 172.168.0.2 0.0.0.0 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
router ospf 10 vrf VPN1
 network 192.168.30.0 255.255.255.0 area 0
 redistribute bgp subnets
!
router ospf 10 vrf VPN2
 network 192.168.40.0 255.255.255.0 area 0
 redistribute bgp subnets
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

P1 configuration file

```
hostname P1
```

```
!
mpls enable
!
interface GigabitEthernet 0/1
 ip address 172.168.10.2 255.255.255.0
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 ip address 172.168.40.1 255.255.255.0
 label-switching
 mpls ldp enable
!
interface Loopback 0
 ip address 172.168.0.3 255.255.255.255
!
router ospf 1
 network 172.168.0.3 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
 network 172.168.40.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

7. Common Errors

The router ID is not 32 bits. As a result, an LDP session fails to be established.

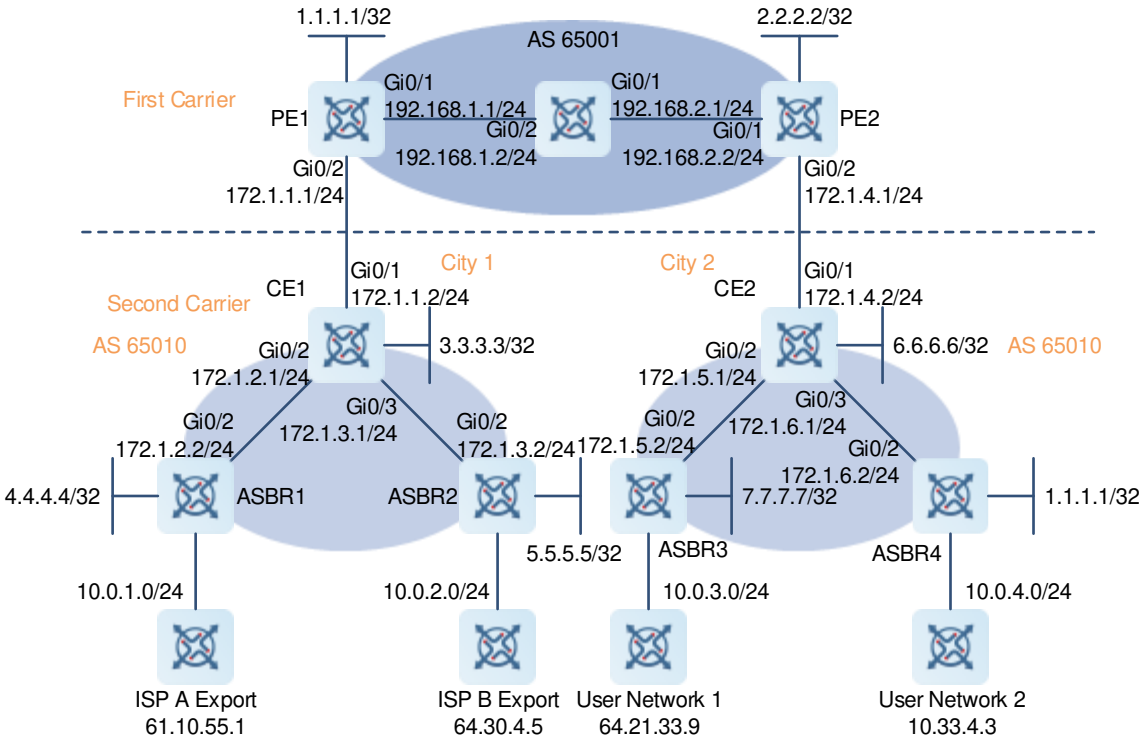
1.16.18 Configuring the Second Carrier to Provide the Internet Service Based on the IP Core

1. Requirements

The carrier has an internal network in city 1, which is connected to the egresses of ISP A and ISP B through BGP. The carrier wants to provide the Internet service to users in city 1 through the internal network. Currently, the carrier wants to expand the Internet service to city 2, so it rents the MPLS VPN service from a VPN carrier to connect the sites in the two cities over the VPN. After the two sites are connected, users in city 2 can also access the Internet through the existing egresses. After the network connection, internal routes are exchanged using IGP (OSPF) and external routes are exchanged using BGP.

2. Topology

Figure 1-36 Configuring the Second Carrier to Provide the Internet Service Based on the IP Core



3. Notes

(1) Configure basic BGP/MPLS VPN features for the first carrier: Configure loopback interfaces and configure MPLS and LDP globally and on interfaces. Configure IGP (OSPF), MP-IBGP neighbors, and VRF instances, connect CEs to PEs, and configure PEs and CEs to exchange routes.

(2) Configure the CSC feature: On the PEs, configure the CSC feature and distribute labels to BGP routes using LDP. On the CEs, configure MPLS and LDP.

(3) Configure the second carrier: Configure interfaces and IGP. On each ASBR, configure the CE as its BGP peer. On a CE, configure the corresponding ASBR and the CE in another site as the RR clients and parse the next hops in BGP routes to LSPs.

(4) Configure user access: Configure EBGP for user access.

4. Procedure

- Configure PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# interface Loopback 0
PE1(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
PE1(config-if-Loopback 0)# exit
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface Loopback 0
```

```
PE1(config-mpls-router)# exit
PE1(config)# interface GigabitEthernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# label-switching
PE1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/1)# no shutdown
PE1(config-if-GigabitEthernet 0/1)# exit
PE1(config)# router ospf 1
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1(config-router)# network 192.168.1.0 0.0.0.255 area 0
PE1(config-router)# exit
PE1(config)# router bgp 65001
PE1(config-router)# neighbor 2.2.2.2 remote-as 65001
PE1(config-router)# neighbor 2.2.2.2 update-source Loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 2.2.2.2 activate
PE1(config-router-af)# neighbor 2.2.2.2 send-community both
PE1(config)# ip vrf vpn1
PE1(config-vrf)# rd 65001:20
PE1(config-vrf)# route-target both 65001:20
PE1(config-vrf)# alloc-label per-route
PE1(config-vrf)# exit
PE1(config)# interface loopback 1
PE1(config-if-Loopback 1)# ip vrf forwarding vpn1
PE1(config-if-Loopback 1)# ip address 10.1.1.1 255.255.255.255
PE1(config-if-Loopback 1)# no shutdown
PE1(config-if-Loopback 1)# exit
PE1(config)# interface GigabitEthernet 0/2
PE1(config-if-GigabitEthernet 0/2)# ip vrf forwarding vpn1
PE1(config-if-GigabitEthernet 0/2)# ip address 172.1.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# no shutdown
PE1(config)# router ospf 100 vrf vpn1
PE1(config-router)# network 172.1.1.0 0.0.0.255 area 0
PE1(config-router)# redistribute bgp subnets
PE1(config-router)# exit
PE1(config)# router bgp 65001
PE1(config-router)# address-family ipv4 vrf vpn1
PE1(config-router-af)# redistribute ospf 100
PE1(config-router-af)# exit
PE1(config-router)# exit
PE1(config)# mpls router ldp vpn1
PE1(config-mpls-router)# ldp router-id interface Loopback 1
PE1(config-mpls-router)# advertise-labels for bgp-routes
PE1(config-mpls-router)# exit
PE1(config)# interface GigabitEthernet 0/2
PE1(config-if-GigabitEthernet 0/2)# label-switching
```

```
PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
```

- Configure PE2.

```
PE2> enable
PE2# configure terminal
PE2(config)# interface Loopback 0
PE2(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
PE2(config-if-Loopback 0)# exit
PE2(config)# mpls enable
PE2(config)# mpls router ldp
PE2(config-mpls-router)# ldp router-id interface Loopback 0
PE2(config-mpls-router)# exit
PE2(config)# interface GigabitEthernet 0/1
PE2(config-if-GigabitEthernet 0/1)# ip address 192.168.2.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/1)# label-switching
PE2(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE2(config-if-GigabitEthernet 0/1)# no shutdown
PE2(config-if-GigabitEthernet 0/1)# exit
PE2(config)# router ospf 1
PE2(config-router)# network 2.2.2.2 0.0.0.0 area 0
PE2(config-router)# network 192.168.2.0 0.0.0.255 area 0
PE2(config-router)# exit
PE2(config)# router bgp 65001
PE2(config-router)# neighbor 1.1.1.1 remote-as 65001
PE2(config-router)# neighbor 1.1.1.1 update-source Loopback 0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 1.1.1.1 activate
PE2(config-router-af)# neighbor 1.1.1.1 send-community both
PE2(config)# ip vrf vpn1
PE2(config-vrf)# rd 65001:20
PE2(config-vrf)# route-target both 65001:20
PE2(config-vrf)# alloc-label per-route
PE2(config-vrf)# exit
PE2(config)# interface loopback 1
PE2(config-if-Loopback 1)# ip vrf forwarding vpn1
PE2(config-if-Loopback 1)# ip address 10.2.2.2 255.255.255.255
PE2(config-if-Loopback 1)# no shutdown
PE2(config-if-Loopback 1)# exit
PE2(config)# interface GigabitEthernet 0/2
PE2(config-if-GigabitEthernet 0/2)# ip vrf forwarding vpn1
PE2(config-if-GigabitEthernet 0/2)# ip address 172.1.4.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/2)# no shutdown
PE2(config)# mpls router ldp vpn1
PE2(config-mpls-router)# ldp router-id interface Loopback 1
PE2(config-mpls-router)# advertise-labels for bgp-routes
PE2(config-mpls-router)# exit
PE2(config)# interface GigabitEthernet 0/2
```

```
PE2(config-if-GigabitEthernet 0/2)# label-switching
PE2(config-if-GigabitEthernet 0/2)# mpls ldp enable
```

- Configure CE1.

```
CE1> enable
CE1# configure terminal
CE1(config)# router ospf 1
CE1(config-router)# network 172.1.1.0 0.0.0.255 area 0
CE1(config-router)# exit
CE1(config)# mpls enable
CE1(config)# mpls router ldp
CE1(config-mpls-router)# ldp router-id interface Loopback 0
CE1(config-mpls-router)# exit
CE1(config)# interface GigabitEthernet 0/1
CE1(config-if-GigabitEthernet 0/1)# ip address 172.1.1.2 255.255.255.0
CE1(config-if-GigabitEthernet 0/1)# no shutdown
CE1(config-if-GigabitEthernet 0/1)# label-switching
CE1(config-if-GigabitEthernet 0/1)# mpls ldp enable
CE1(config-if-GigabitEthernet 0/1)# exit
CE1(config)# interface GigabitEthernet 0/2
CE1(config-if-GigabitEthernet 0/2)# ip address 172.1.2.1 255.255.255.0
CE1(config-if-GigabitEthernet 0/2)# no shutdown
CE1(config-if-GigabitEthernet 0/2)# exit
CE1(config)# interface GigabitEthernet 0/3
CE1(config-if-GigabitEthernet 0/3)# ip address 172.1.3.1 255.255.255.0
CE1(config-if-GigabitEthernet 0/3)# no shutdown
CE1(config-if-GigabitEthernet 0/3)# exit
CE1(config)# interface Loopback 0
CE1(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
CE1(config-if-Loopback 0)# exit
CE1(config)# router ospf 1
CE1(config-router)# network 3.3.3.3 0.0.0.0 area 0
CE1(config-router)# network 172.1.2.0 0.0.0.255 area 0
CE1(config-router)# network 172.1.3.0 0.0.0.255 area 0
CE1(config-router)# exit
CE1(config)# router bgp 65010
CE1(config-router)# neighbor 4.4.4.4 remote-as 65010
CE1(config-router)# neighbor 4.4.4.4 update-source Loopback 0
CE1(config-router)# neighbor 4.4.4.4 route-reflector-client
CE1(config-router)# neighbor 5.5.5.5 remote-as 65010
CE1(config-router)# neighbor 5.5.5.5 update-source Loopback 0
CE1(config-router)# neighbor 5.5.5.5 route-reflector-client
CE1(config-router)# neighbor 6.6.6.6 remote-as 65010
CE1(config-router)# neighbor 6.6.6.6 update-source Loopback 0
CE1(config-router)# neighbor 6.6.6.6 route-reflector-client
CE1(config-router)# exit
CE1(config)# recursive-route lookup lsp
```

- Configure CE2.

```
CE2> enable
CE2# configure terminal
CE2(config)# router ospf 1
CE2(config-router)# network 172.1.4.0 0.0.0.255 area 0
CE2(config-router)# exit
CE2(config)# mpls enable
CE2(config)# mpls router ldp
CE2(config-mpls-router)# ldp router-id interface Loopback 0
CE2(config-mpls-router)# exit
CE2(config)# interface GigabitEthernet 0/1
CE2(config-if-GigabitEthernet 0/1)# ip address 172.1.4.2 255.255.255.0
CE2(config-if-GigabitEthernet 0/1)# no shutdown
CE2(config-if-GigabitEthernet 0/1)# label-switching
CE2(config-if-GigabitEthernet 0/1)# mpls ldp enable
CE2(config-if-GigabitEthernet 0/1)# exit
CE2(config)# interface GigabitEthernet 0/2
CE2(config-if-GigabitEthernet 0/2)# ip address 172.1.5.1 255.255.255.0
CE2(config-if-GigabitEthernet 0/2)# no shutdown
CE2(config-if-GigabitEthernet 0/2)# exit
CE2(config)# interface GigabitEthernet 0/3
CE2(config-if-GigabitEthernet 0/3)# ip address 172.1.6.1 255.255.255.0
CE2(config-if-GigabitEthernet 0/3)# no shutdown
CE2(config-if-GigabitEthernet 0/3)# exit
CE2(config)# interface Loopback 0
CE2(config-if-Loopback 0)# ip address 6.6.6.6 255.255.255.255
CE2(config-if-Loopback 0)# exit
CE2(config)# router ospf 1
CE2(config-router)# network 6.6.6.6 0.0.0.0 area 0
CE2(config-router)# network 172.1.5.0 0.0.0.255 area 0
CE2(config-router)# network 172.1.6.0 0.0.0.255 area 0
CE2(config-router)# exit
CE2(config)# router bgp 65010
CE2(config-router)# neighbor 7.7.7.7 remote-as 65010
CE2(config-router)# neighbor 7.7.7.7 update-source Loopback 0
CE2(config-router)# neighbor 7.7.7.7 route-reflector-client
CE2(config-router)# neighbor 8.8.8.8 remote-as 65010
CE2(config-router)# neighbor 8.8.8.8 update-source Loopback 0
CE2(config-router)# neighbor 8.8.8.8 route-reflector-client
CE2(config-router)# neighbor 3.3.3.3 remote-as 65010
CE2(config-router)# neighbor 3.3.3.3 update-source Loopback 0
CE2(config-router)# neighbor 3.3.3.3 route-reflector-client
CE2(config-router)# exit
CE2(config)# recursive-route lookup lsp
```

- Configure ASBR1.

```
ASBR1> enable
```

```
ASBR1# configure terminal
ASBR1(config)# interface GigabitEthernet 0/2
ASBR1(config-if-GigabitEthernet 0/2)# ip address 172.1.2.2 255.255.255.0
ASBR1(config-if-GigabitEthernet 0/2)# no shutdown
ASBR1(config-if-GigabitEthernet 0/2)# exit
ASBR1(config)# interface GigabitEthernet 0/1
ASBR1(config-if-GigabitEthernet 0/1)# ip address 10.0.1.1 255.255.255.0
ASBR1(config-if-GigabitEthernet 0/1)# no shutdown
ASBR1(config-if-GigabitEthernet 0/1)# exit
ASBR1(config)# interface Loopback 0
ASBR1(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
ASBR1(config-if-Loopback 0)# exit
ASBR1(config)# router ospf 1
ASBR1(config-router)# network 4.4.4.4 0.0.0.0 area 0
ASBR1(config-router)# network 172.1.2.0 0.0.0.255 area 0
ASBR1(config-router)# exit
ASBR1(config)# router bgp 65010
ASBR1(config-router)# neighbor 3.3.3.3 remote-as 65010
ASBR1(config-router)# neighbor 3.3.3.3 update-source Loopback 0
ASBR1(config-router)# neighbor 3.3.3.3 next-hop-self
ASBR1(config-router)# exit
ASBR1(config)# router bgp 65010
ASBR1(config-router)# neighbor 10.0.1.2 remote-as 100
ASBR1(config-router)# exit
```

- Configure ASBR2.

```
ASBR2> enable
ASBR2# configure terminal
ASBR2(config)# interface GigabitEthernet 0/2
ASBR2(config-if-GigabitEthernet 0/2)# ip address 172.1.3.2 255.255.255.0
ASBR2(config-if-GigabitEthernet 0/2)# no shutdown
ASBR2(config-if-GigabitEthernet 0/2)# exit
ASBR2(config)# interface GigabitEthernet 0/1
ASBR2(config-if-GigabitEthernet 0/1)# ip address 10.0.2.1 255.255.255.0
ASBR2(config-if-GigabitEthernet 0/1)# no shutdown
ASBR2(config-if-GigabitEthernet 0/1)# exit
ASBR2(config)# interface Loopback 0
ASBR2(config-if-Loopback 0)# ip address 5.5.5.5 255.255.255.255
ASBR2(config-if-Loopback 0)# exit
ASBR2(config)# router ospf 1
ASBR2(config-router)# network 5.5.5.5 0.0.0.0 area 0
ASBR2(config-router)# network 172.1.3.0 0.0.0.255 area 0
ASBR2(config-router)# exit
ASBR2(config)# router bgp 65010
ASBR2(config-router)# neighbor 3.3.3.3 remote-as 65010
ASBR2(config-router)# neighbor 3.3.3.3 update-source Loopback 0
ASBR2(config-router)# neighbor 3.3.3.3 next-hop-self
```

```
ASBR2(config-router)# exit
ASBR2(config)# router bgp 65010
ASBR2(config-router)# neighbor 10.0.2.2 remote-as 100
ASBR2(config-router)# exit
```

- Configure ASBR3.

```
ASBR3> enable
ASBR3# configure terminal
ASBR3(config)# interface GigabitEthernet 0/2
ASBR3(config-if-GigabitEthernet 0/2)# ip address 172.1.5.2 255.255.255.0
ASBR3(config-if-GigabitEthernet 0/2)# no shutdown
ASBR3(config-if-GigabitEthernet 0/2)# exit
ASBR3(config)# interface GigabitEthernet 0/1
ASBR3(config-if-GigabitEthernet 0/1)# ip address 10.0.3.1 255.255.255.0
ASBR3(config-if-GigabitEthernet 0/1)# no shutdown
ASBR3(config-if-GigabitEthernet 0/1)# exit
ASBR3(config)# interface Loopback 0
ASBR3(config-if-Loopback 0)# ip address 7.7.7.7 255.255.255.255
ASBR3(config-if-Loopback 0)# exit
ASBR3(config)# router ospf 1
ASBR3(config-router)# network 7.7.7.7 0.0.0.0 area 0
ASBR3(config-router)# network 172.1.5.0 0.0.0.255 area 0
ASBR3(config-router)# exit
ASBR3(config)# router bgp 65010
ASBR3(config-router)# neighbor 6.6.6.6 remote-as 65010
ASBR3(config-router)# neighbor 6.6.6.6 update-source Loopback 0
ASBR3(config-router)# neighbor 6.6.6.6 next-hop-self
ASBR3(config-router)# exit
ASBR3(config)# router bgp 65010
ASBR3(config-router)# neighbor 10.0.3.2 remote-as 100
ASBR3(config-router)# exit
```

- Configure ASBR4.

```
ASBR4> enable
ASBR4# configure terminal
ASBR4(config)# interface GigabitEthernet 0/2
ASBR4(config-if-GigabitEthernet 0/2)# ip address 172.1.6.2 255.255.255.0
ASBR4(config-if-GigabitEthernet 0/2)# no shutdown
ASBR4(config-if-GigabitEthernet 0/2)# exit
ASBR4(config)# interface GigabitEthernet 0/1
ASBR4(config-if-GigabitEthernet 0/1)# ip address 10.0.4.1 255.255.255.0
ASBR4(config-if-GigabitEthernet 0/1)# no shutdown
ASBR4(config-if-GigabitEthernet 0/1)# exit
ASBR4(config)# interface Loopback 0
ASBR4(config-if-Loopback 0)# ip address 8.8.8.8 255.255.255.255
ASBR4(config-if-Loopback 0)# exit
ASBR4(config)# router ospf 1
ASBR4(config-router)# network 8.8.8.8 0.0.0.0 area 0
```

```
ASBR4(config-router)# network 172.1.6.0 0.0.0.255 area 0
ASBR4(config-router)# exit
ASBR4(config)# router bgp 65010
ASBR4(config-router)# neighbor 6.6.6.6 remote-as 65010
ASBR4(config-router)# neighbor 6.6.6.6 update-source Loopback 0
ASBR4(config-router)# neighbor 6.6.6.6 next-hop-self
ASBR4(config-router)# exit
ASBR4(config)# router bgp 65010
ASBR4(config-router)# neighbor 10.0.4.2 remote-as 100
ASBR4(config-router)# exit
```

- Configure the edge device on user network 1.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# interface GigabitEthernet 0/0
Hostname(config-if-GigabitEthernet 0/0)# ip address 10.0.3.2 255.255.255.0
Hostname(config-if-GigabitEthernet 0/0)# no shutdown
Hostname(config-if-GigabitEthernet 0/0)# exit
Hostname(config)# interface GigabitEthernet 0/1
Hostname(config-if-GigabitEthernet 0/1)# ip address 64.21.33.9 255.255.255.0
Hostname(config-if-GigabitEthernet 0/1)# no shutdown
Hostname(config-if-GigabitEthernet 0/1)# exit
Hostname(config)# router bgp 100
Hostname(config-router)# neighbor 10.0.3.1 remote-as 65010
Hostname(config-router)# network 64.21.33.0 mask 255.255.255.0
```

5. Verification

- Check the route and label information of the VRF instance on PE1.
  Check the route information of the VRF instance.

```
PE1# show ip route vrf vpn1
Routing Table: vpn1
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
O    3.3.3.3/32 [110/11] via 172.1.1.2, GigabitEthernet 0/2, 00:00:07
C    172.1.1.0/24 is directly connected, GigabitEthernet 0/2, 00:00:03
L    172.1.1.1/32 is directly connected, GigabitEthernet 0/2, 00:00:03
O    172.1.2.0/24 [110/12] via 172.1.1.2, GigabitEthernet 0/2, 00:00:07
B    172.1.4.0/24 [200/0] via 2.2.2.2, 00:00:30
```

  Check the label information.

```
PE1# show mpls ldp bindings vrf vpn1
VRF vpn1(id 1)
    lib entry: 3.3.3.3/32
        local binding:  to lsr:   172.1.1.2:0, label: 1025
        remote binding: from lsr: 172.1.1.2:0, label: imp-null
    lib entry: 172.1.1.0/24
        local binding:  to lsr:   172.1.1.2:0, label: imp-null
        remote binding: from lsr: 172.1.1.2:0, label: imp-null
    lib entry 172.1.2.0/24
        local binding:  to lsr:   172.1.1.2:0, label: 1026
        remote binding: from lsr: 172.1.1.2:0, label: 1024
```

- Check the route and label information of the VRF instance on PE2.

  Check the route information of the VRF instance.

```
PE2# show ip route vrf vpn1
Routing Table: vpn1
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
O    6.6.6.6/32 [110/11] via 172.1.4.2, GigabitEthernet 0/2, 00:00:07
C    172.1.4.0/24 is directly connected, GigabitEthernet 0/2, 00:00:02
L    172.1.4.1/32 is directly connected, GigabitEthernet 0/2, 00:00:02
O    172.1.6.0/24 [110/12] via 172.1.4.2, GigabitEthernet 0/2, 00:00:07
B    172.1.1.0/24 [200/0] via 1.1.1.1, 00:00:30
```

  Check the label information.

```
PE1# show mpls ldp bindings vrf vpn1
VRF vpn1(id 1)
    lib entry: 6.6.6.6/32
        local binding:  to lsr:   172.1.4.2:0, label: 1025
        remote binding: from lsr: 172.1.4.2:0, label: imp-null
    lib entry: 172.1.4.0/24
        local binding:  to lsr:   172.1.4.2:0, label: imp-null
        remote binding: from lsr: 172.1.4.2:0, label: imp-null
    lib entry 172.1.6.0/24
        local binding:  to lsr:   172.1.4.2:0, label: 1026
        remote binding: from lsr: 172.1.4.2:0, label: 1024
```

- Check the routing table on ASBR3.

```
ASBR3# show ip route
Codes:  C - Connected, L - Local, S - Static
```

```
      R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      IA - Inter area, EV - BGP EVPN, A - Arp to host
      LA - Local aggregate route
      * - candidate default

Gateway of last resort is no set
......
O   3.3.3.3/24 [110/12] via 172.1.5.1, GigabitEthernet 0/2, 00:00:30
B   61.10.55.0/24 [200/0] via 10.10.10.10, 00:00:40
B   64.21.33.0/24 [200/0] via 10.0.3.2, 00:00:31
```

- Check the routing table on the edge device in user network 1.

```
Hostname# show ip route
Codes:  C - Connected, L - Local, S - Static
      R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      IA - Inter area, EV - BGP EVPN, A - Arp to host
      LA - Local aggregate route
      * - candidate default

Gateway of last resort is no set
......
B   61.10.55.0/24 [200/0] via 10.0.3.1, 00:00:40
C   64.21.33.0/24 is directly connected, GigabitEthernet 0/1, 00:00:02
L   64.21.33.9/32 is directly connected, GigabitEthernet 0/1, 00:00:02
```

- Ping the egress of ISP A from the edge device on user network 1.

```
Hostname # ping 61.10.55.1 source 64.21.33.9
Sending 5, 100-byte ICMP Echoes to 61.10.55.1, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!Success rate is 100 percent (5/5), round-trip min/avg/max = 10/20/40 ms
```

6. Configuration Files

- PE1 configuration file

```
hostname PE1
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
mpls enable
!
mpls router ldp
 ldp rouer-id interface Loopback 0
```

```
!
interface GigabitEthernet 0/1
 ip address 192.168.1.1 255.255.255.0
 label-switching
 mpls ldp enable
 no shutdown
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
router bgp 65001
 neighbor 2.2.2.2 remote-as 65001
 neighbor 2.2.2.2 update-source Loopback 0
 !
 address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community both
 exit-address-family
!
ip vrf vpn1
 rd 65001:20
 route-target both 65001:20
 alloc-label per-route
!
interface loopback 1
 ip vrf forwarding vpn1
 ip address 10.1.1.1 255.255.255.255
 no shutdown
!
interface GigabitEthernet 0/2
 ip vrf forwarding vpn1
 ip address 172.1.1.1 255.255.255.0
 no shutdown
!
router ospf 100 vrf vpn1
 network 172.1.1.0 0.0.0.255 area 0
 redistribute bgp subnets
!
router bgp 65001
 address-family ipv4 vrf vpn1
  redistribute ospf 100
 exit-address-family
!
mpls router ldp vpn1
 ldp rouer-id interface Loopback 1
```

```
 advertise-labels for bgp-routes
!
interface GigabitEthernet 0/2
 label-switching
 mpls ldp enable
!
```

- PE2 configuration file

```
hostname PE2
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
mpls enable
!
mpls router ldp
 ldp rouer-id interface Loopback 0
!
interface GigabitEthernet 0/1
 ip address 192.168.2.2 255.255.255.0
 label-switching
 mpls ldp enable
 no shutdown
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.2.0 0.0.0.255 area 0
!
router bgp 65001
 neighbor 1.1.1.1 remote-as 65001
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
!
ip vrf vpn1
 rd 65001:20
 route-target both 65001:20
 alloc-label per-route
!
interface loopback 1
 ip vrf forwarding vpn1
 ip address 10.2.2.2 255.255.255.255
 no shutdown
!
```

```
interface GigabitEthernet 0/2
 ip vrf forwarding vpn1
 ip address 172.1.4.1 255.255.255.0
 no shutdown
!
mpls router ldp vpn1
 ldp rouer-id interface Loopback 1
 advertise-labels for bgp-routes
!
interface GigabitEthernet 0/2
 label-switching
 mpls ldp enable
!
```

- CE1 configuration file

```
hostname CE1
!
router ospf 1
 network 172.1.1.0 0.0.0.255 area 0
!
mpls enable
!
mpls router ldp
 ldp rouer-id interface Loopback 0
!
interface GigabitEthernet 0/1
 ip address 172.1.1.2 255.255.255.0
 no shutdown
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 ip address 172.1.2.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/3
 ip address 172.1.3.1 255.255.255.0
 no shutdown
!
interface Loopback 0
ip address 3.3.3.3 255.255.255.255
!
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 172.1.2.0 0.0.0.255 area 0
 network 172.1.3.0 0.0.0.255 area 0
!
```

```
router bgp 65010
 neighbor 4.4.4.4 remote-as 65010
 neighbor 4.4.4.4 update-source Loopback 0
 neighbor 4.4.4.4 route-reflector-client
 neighbor 5.5.5.5 remote-as 65010
 neighbor 5.5.5.5 update-source Loopback 0
 neighbor 5.5.5.5 route-reflector-client
 neighbor 6.6.6.6 remote-as 65010
 neighbor 6.6.6.6 update-source Loopback 0
 neighbor 6.6.6.6 route-reflector-client
!
recursive-route lookup lsp
!
```

- CE2 configuration file

```
hostname CE2
!
router ospf 1
 network 172.1.4.0 0.0.0.255 area 0
!
mpls enable
!
mpls router ldp
 ldp rouer-id interface Loopback 0
!
interface GigabitEthernet 0/1
 ip address 172.1.4.2 255.255.255.0
 no shutdown
 label-switching
 mpls ldp enable
!
interface GigabitEthernet 0/2
 ip address 172.1.5.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/3
 ip address 172.1.6.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 6.6.6.6 255.255.255.255
!
router ospf 1
 network 6.6.6.6 0.0.0.0 area 0
 network 172.1.5.0 0.0.0.255 area 0
 network 172.1.6.0 0.0.0.255 area 0
!
```

```
router bgp 65010
 neighbor 7.7.7.7 remote-as 65010
 neighbor 7.7.7.7 update-source Loopback 0
 neighbor 7.7.7.7 route-reflector-client
 neighbor 8.8.8.8 remote-as 65010
 neighbor 8.8.8.8 update-source Loopback 0
 neighbor 8.8.8.8 route-reflector-client
 neighbor 3.3.3.3 remote-as 65010
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 3.3.3.3 route-reflector-client
!
recursive-route lookup lsp
!
```

- ASBR1 configuration file

```
hostname ASBR1
!
interface GigabitEthernet 0/2
 ip address 172.1.2.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.1.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router ospf 1
 network 4.4.4.4 0.0.0.0 area 0
 network 172.1.2.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 3.3.3.3 remote-as 65010
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 3.3.3.3 next-hop-self
!
router bgp 65010
 neighbor 10.0.1.2 remote-as 100
!
```

- ASBR2 configuration

```
hostname ASBR2
!
interface GigabitEthernet 0/2
 ip address 172.1.3.2 255.255.255.0
 no shutdown
!
```

```
interface GigabitEthernet 0/1
 ip address 10.0.2.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router ospf 1
 network 5.5.5.5 0.0.0.0 area 0
 network 172.1.3.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 3.3.3.3 remote-as 65010
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 3.3.3.3 next-hop-self
!
router bgp 65010
 neighbor 10.0.2.2 remote-as 100
!
```

- ASBR3 configuration

```
hostname ASBR3
!
interface GigabitEthernet 0/2
 ip address 172.1.5.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.3.1 255.255.255.0
 no shutdown
!
 interface Loopback 0
 ip address 7.7.7.7 255.255.255.255
!
 router ospf 1
 network 7.7.7.7 0.0.0.0 area 0
 network 172.1.5.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 6.6.6.6 remote-as 65010
 neighbor 6.6.6.6 update-source Loopback 0
 neighbor 6.6.6.6 next-hop-self
!
router bgp 65010
 neighbor 10.0.3.2 remote-as 100
!
```

- ASBR4 configuration

```
hostname ASBR4
!
interface GigabitEthernet 0/2
 ip address 172.1.6.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.4.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 8.8.8.8 255.255.255.255
!
router ospf 1
 network 8.8.8.8 0.0.0.0 area 0
 network 172.1.6.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 6.6.6.6 remote-as 65010
 neighbor 6.6.6.6 update-source Loopback 0
 neighbor 6.6.6.6 next-hop-self
!
router bgp 65010
 neighbor 10.0.4.2 remote-as 100
!
```

- User Network1 edge device configuration

```
!
interface GigabitEthernet 0/0
 ip address 10.0.3.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 64.21.33.9 255.255.255.0
 no shutdown
!
router bgp 100
 neighbor 10.0.3.1 remote-as 65010
 network 64.21.33.0 mask 255.255.255.0
!
```

1.16.19 Configuring the Second Carrier to Provide the Internet Service Based on the MPLS Core

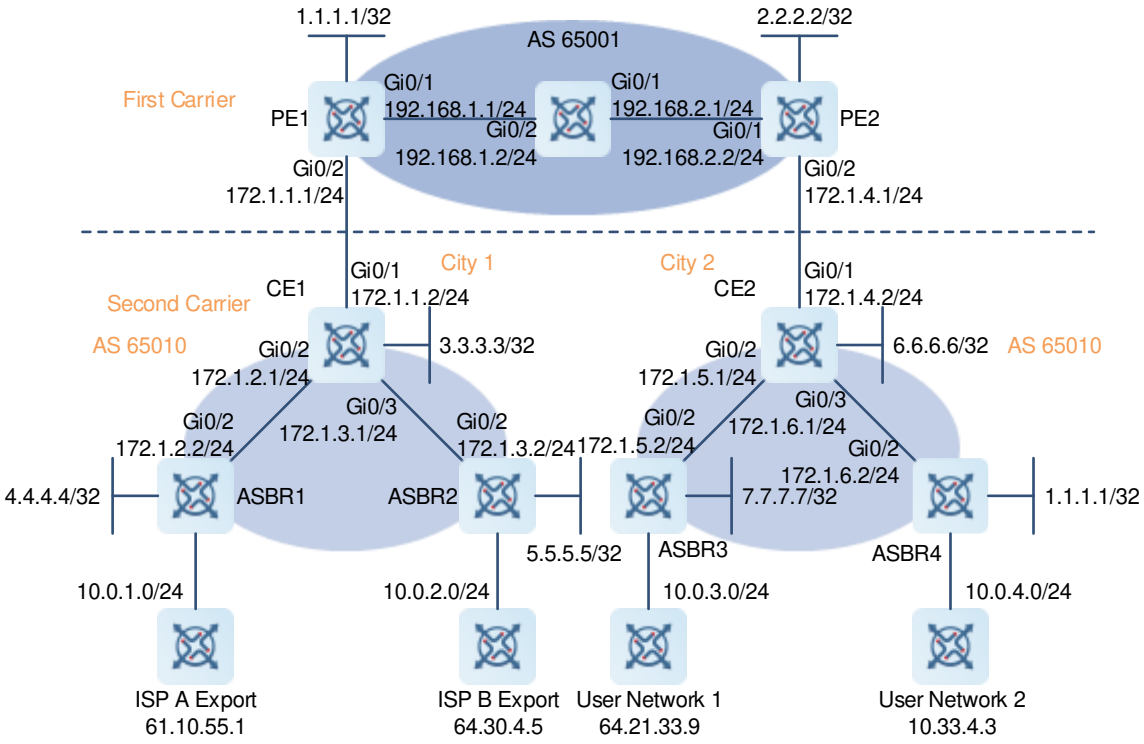
1. Requirements

The carrier provides the Internet service to users in city 1. Considering that the carrier may want to provide MPLS services to users in the future, the carrier deploys an MPLS backbone network. To expand services to city 2, the carrier deploys an MPLS network in city 2. To implement interconnection between core networks in two cities, this carrier rents the VPN service from another MPLS VPN provider. In this scenario, this carrier is the second carrier and the carrier providing the MPLS VPN services is the first carrier.

After the networking, the PEs of the first carrier exchange internal routes with the CEs of the second carrier through BGP and BGP neighbor relationships are established directly between the ASBRs of the second carrier that provide services to users to exchange external routes. Traffic coming from an external network to the second carrier network is forwarded over a tunnel until it leaves the second carrier network.

2. Topology

Figure 1-3 Configuring the Second Carrier to Provide the Internet Service Based on the MPLS Core



3. Notes

(1) Configure basic BGP/MPLS VPN features for the first carrier: Configure loopback interfaces and configure MPLS and LDP globally and on interfaces. Configure IGP (OSPF), MP-IBGP neighbors, and VRF instances, connect CEs to PEs, and configure PEs and CEs to exchange routes.

(2) Configure the CSC feature: On the PEs, configure the CSC feature and distribute MPLS labels to IPv4 routes. On the CEs, configure MPLS and LDP.

(3) Configure the second carrier: Configure interfaces and IGP. On each ASBR, configure the CE as its BGP peer. On a CE, configure the corresponding ASBR and the CE in another site as the RR clients and parse the next hops in BGP routes to LSPs.

(4) Configure user access: Configure EBGP for user access.

Note

LDP must be enabled on the CSC-CE to establish sessions with other devices in the same site so as to establish an MPLS network. If the CSC-CE and CSC-PE learn routes using BGP, you must run the **advertise-labels for bgp-routes** command on the CSC-CE to enable label distribution to BGP routes through LDP.

4. Procedure

- Configure PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# interface Loopback 0
PE1(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
PE1(config-if-Loopback 0)# exit
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface Loopback 0
PE1(config-mpls-router)# exit
PE1(config)# interface GigabitEthernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# label-switching
PE1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/1)# no shutdown
PE1(config-if-GigabitEthernet 0/1)# exit
PE1(config)# router ospf 1
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1(config-router)# network 192.168.1.0 0.0.0.255 area 0
PE1(config-router)# exit
PE1(config)# router bgp 65001
PE1(config-router)# neighbor 2.2.2.2 remote-as 65001
PE1(config-router)# neighbor 2.2.2.2 update-source Loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 2.2.2.2 activate
PE1(config-router-af)# neighbor 2.2.2.2 send-community both
PE1(config-router-af)# exit
PE1(config-router)# exit
PE1(config)# ip vrf vpn1
PE1(config-vrf)# rd 65001:20
PE1(config-vrf)# route-target both 65001:20
PE1(config-vrf)# alloc-label per-route
PE1(config-vrf)# exit
PE1(config)# interface loopback 1
```

```
PE1(config-if-Loopback 1)# ip vrf forwarding vpn1
PE1(config-if-Loopback 1)# ip address 10.1.1.1 255.255.255.255
PE1(config-if-Loopback 1)# no shutdown
PE1(config-if-Loopback 1)# exit
PE1(config)# interface GigabitEthernet 0/2
PE1(config-if-GigabitEthernet 0/2)# ip vrf forwarding vpn1
PE1(config-if-GigabitEthernet 0/2)# ip address 172.1.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# no shutdown
PE1(config-if-GigabitEthernet 0/2)# exit
PE1(config)# router bgp 65001
PE1(config-router)# address-family ipv4 vrf vpn1
PE1(config-router-af)# neighbor 172.1.1.2 remote-as 65010
PE1(config-router-af)# neighbor 172.1.1.2 as-override
PE1(config-router-af)# neighbor 172.1.1.2 send-label
PE1(config-router-af)# exit
PE1(config-router)# exit
```

- Configure PE2.

```
PE2> enable
PE2# configure terminal
PE2(config)# interface Loopback 0
PE2(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
PE2(config-if-Loopback 0)# exit
PE2(config)# mpls enable
PE2(config)# mpls router ldp
PE2(config-mpls-router)# ldp router-id interface Loopback 0
PE2(config-mpls-router)# exit
PE2(config)# interface GigabitEthernet 0/1
PE2(config-if-GigabitEthernet 0/1)# ip address 192.168.2.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/1)# label-switching
PE2(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE2(config-if-GigabitEthernet 0/1)# no shutdown
PE2(config-if-GigabitEthernet 0/1)# exit
PE2(config)# router ospf 1
PE2(config-router)# network 2.2.2.2 0.0.0.0 area 0
PE2(config-router)# network 192.168.2.0 0.0.0.255 area 0
PE2(config-router)# exit
PE2(config)# router bgp 65001
PE2(config-router)# neighbor 1.1.1.1 remote-as 65001
PE2(config-router)# neighbor 1.1.1.1 update-source Loopback 0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 1.1.1.1 activate
PE2(config-router-af)# neighbor 1.1.1.1 send-community both
PE2(config-router-af)# exit
PE2(config-router)# exit
PE2(config)# ip vrf vpn1
PE2(config-vrf)# rd 65001:20
```

```
PE2(config-vrf)# route-target both 65001:20
PE2(config-vrf)# alloc-label per-route
PE2(config-vrf)# exit
PE2(config)# interface loopback 1
PE2(config-if-Loopback 1)# ip vrf forwarding vpn1
PE2(config-if-Loopback 1)# ip address 10.1.2.1 255.255.255.255
PE2(config-if-Loopback 1)# no shutdown
PE2(config-if-Loopback 1)# exit
PE2(config)# interface GigabitEthernet 0/2
PE2(config-if-GigabitEthernet 0/2)# ip vrf forwarding vpn1
PE2(config-if-GigabitEthernet 0/2)# ip address 172.1.4.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/2)# no shutdown
PE2(config)# router bgp 65001
PE2(config-router)# address-family ipv4 vrf vpn1
PE2(config-router-af)# neighbor 172.1.4.2 remote-as 65010
PE2(config-router-af)# neighbor 172.1.4.2 as-override
PE2(config-router-af)# neighbor 172.1.4.2 send-label
PE2(config-router-af)# exit
PE2(config-router)# exit
```

- Configure CE1.

```
CE1> enable
CE1# configure terminal
CE1(config)# interface GigabitEthernet 0/1
CE1(config-if-GigabitEthernet 0/1)# ip address 172.1.1.2 255.255.255.0
CE1(config-if-GigabitEthernet 0/1)# no shutdown
CE1(config-if-GigabitEthernet 0/1)# exit
CE1(config)# router bgp 65010
CE1(config-router)# neighbor 172.1.1.2 remote-as 65001
CE1(config-router)# redistribute ospf 1
CE1(config-router)# exit
CE1(config)# router ospf 1
CE1(config-router)# redistribute bgp subnets
CE1(config-router)# exit
CE1(config)# interface GigabitEthernet 0/1
CE1(config-if-GigabitEthernet 0/1)# label-switching
CE1(config-if-GigabitEthernet 0/1)# ip address 172.1.1.2 255.255.255.0
CE1(config-if-GigabitEthernet 0/1)# no shutdown
CE1(config-if-GigabitEthernet 0/1)# exit
CE1(config)# router bgp 65010
CE1(config-router)# neighbor 172.1.1.1 send-label
CE1(config-router)# exit
CE1(config)# interface GigabitEthernet 0/2
CE1(config-if-GigabitEthernet 0/2)# ip address 172.1.2.1 255.255.255.0
CE1(config-if-GigabitEthernet 0/2)# no shutdown
CE1(config-if-GigabitEthernet 0/2)# exit
CE1(config)# interface GigabitEthernet 0/3
```

```
CE1(config-if-GigabitEthernet 0/3)# ip address 172.1.3.1 255.255.255.0
CE1(config-if-GigabitEthernet 0/3)# no shutdown
CE1(config-if-GigabitEthernet 0/3)# exit
CE1(config)# interface Loopback 0
CE1(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
CE1(config-if-Loopback 0)# exit
CE1(config)# router ospf 1
CE1(config-router)# network 3.3.3.3 0.0.0.0 area 0
CE1(config-router)# network 172.1.2.0 0.0.0.255 area 0
CE1(config-router)# network 172.1.3.0 0.0.0.255 area 0
CE1(config-router)# exit
```

- Configure CE2.

```
CE2> enable
CE2# configure terminal
CE2(config)# interface GigabitEthernet 0/1
CE2(config-if-GigabitEthernet 0/1)# ip address 172.1.4.2 255.255.255.0
CE2(config-if-GigabitEthernet 0/1)# no shutdown
CE2(config)# router bgp 65010
CE2(config-router)# neighbor 172.1.4.2 remote-as 65001
CE2(config-router)# redistribute ospf 1
CE2(config-router)# exit
CE2(config)# router ospf 1
CE2(config-router)# redistribute bgp subnets
CE2(config-router)# exit
CE2(config)# interface GigabitEthernet 0/1
CE2(config-if-GigabitEthernet 0/1)# label-switching
CE2(config-if-GigabitEthernet 0/1)# ip address 172.1.5.1 255.255.255.0
CE2(config-if-GigabitEthernet 0/1)# no shutdown
CE2(config-if-GigabitEthernet 0/1)# exit
CE2(config)# router bgp 65010
CE2(config-router)# neighbor 172.1.4.1 send-label
CE2(config-router)# exit
CE2(config)# interface GigabitEthernet 0/2
CE2(config-if-GigabitEthernet 0/2)# ip address 172.1.5.1 255.255.255.0
CE2(config-if-GigabitEthernet 0/2)# no shutdown
CE2(config-if-GigabitEthernet 0/2)# exit
CE2(config)# interface GigabitEthernet 0/3
CE2(config-if-GigabitEthernet 0/3)# ip address 172.1.6.1 255.255.255.0
CE2(config-if-GigabitEthernet 0/3)# no shutdown
CE2(config-if-GigabitEthernet 0/3)# exit
CE2(config)# interface Loopback 0
CE2(config-if-Loopback 0)# ip address 6.6.6.6 255.255.255.255
CE2(config-if-Loopback 0)# exit
CE2(config)# router ospf 1
CE2(config-router)# network 6.6.6.6 0.0.0.0 area 0
CE2(config-router)# network 172.1.5.0 0.0.0.255 area 0
```

```
CE2(config-router)# network 172.1.6.0 0.0.0.255 area 0
CE2(config-router)# exit
```

- Configure ASBR1.

```
ASBR1> enable
ASBR1# configure terminal
ASBR1(config)# interface GigabitEthernet 0/2
ASBR1(config-if-GigabitEthernet 0/2)# ip address 172.1.2.2 255.255.255.0
ASBR1(config-if-GigabitEthernet 0/2)# no shutdown
ASBR1(config-if-GigabitEthernet 0/2)# exit
ASBR1(config)# interface GigabitEthernet 0/1
ASBR1(config-if-GigabitEthernet 0/1)# ip address 10.0.1.1 255.255.255.0
ASBR1(config-if-GigabitEthernet 0/1)# no shutdown
ASBR1(config-if-GigabitEthernet 0/1)# exit
ASBR1(config)# interface Loopback 0
ASBR1(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
ASBR1(config-if-Loopback 0)# exit
ASBR1(config)# router ospf 1
ASBR1(config-router)# network 4.4.4.4 0.0.0.0 area 0
ASBR1(config-router)# network 172.1.2.0 0.0.0.255 area 0
ASBR1(config-router)# exit
ASBR1(config)# router bgp 65010
ASBR1(config-router)# neighbor 5.5.5.5 remote-as 65010
ASBR1(config-router)# neighbor 5.5.5.5 update-source Loopback 0
ASBR1(config-router)# neighbor 5.5.5.5 next-hop-self
ASBR1(config-router)# exit
ASBR1(config)# recursive-route lookup lsp
ASBR1(config)# router bgp 65010
ASBR1(config-router)# neighbor 10.0.1.2 remote-as 100
ASBR1(config-router)# exit
```

- Configure ASBR2.

```
ASBR2> enable
ASBR2# configure terminal
ASBR2(config)# interface GigabitEthernet 0/2
ASBR2(config-if-GigabitEthernet 0/2)# ip address 172.1.3.2 255.255.255.0
ASBR2(config-if-GigabitEthernet 0/2)# no shutdown
ASBR2(config-if-GigabitEthernet 0/2)# exit
ASBR2(config)# interface GigabitEthernet 0/1
ASBR2(config-if-GigabitEthernet 0/1)# ip address 10.0.2.1 255.255.255.0
ASBR2(config-if-GigabitEthernet 0/1)# no shutdown
ASBR2(config-if-GigabitEthernet 0/1)# exit
ASBR2(config)# interface Loopback 0
ASBR2(config-if-Loopback 0)# ip address 5.5.5.5 255.255.255.255
ASBR2(config-if-Loopback 0)# exit
ASBR2(config)# router ospf 1
ASBR2(config-router)# network 5.5.5.5 0.0.0.0 area 0
ASBR2(config-router)# network 172.1.3.0 0.0.0.255 area 0
```

```
ASBR2(config-router)# exit
ASBR2(config)# router bgp 65010
ASBR2(config-router)# neighbor 4.4.4.4 remote-as 65010
ASBR2(config-router)# neighbor 4.4.4.4 update-source Loopback 0
ASBR2(config-router)# neighbor 4.4.4.4 next-hop-self
ASBR2(config-router)# exit
ASBR2(config)# recursive-route lookup lsp
ASBR2(config)# router bgp 65010
ASBR2(config-router)# neighbor 10.0.2.2 remote-as 100
ASBR2(config-router)# exit
```

- Configure ASBR3.

```
ASBR3> enable
ASBR3# configure terminal
ASBR3(config)# interface GigabitEthernet 0/2
ASBR3(config-if-GigabitEthernet 0/2)# ip address 172.1.5.2 255.255.255.0
ASBR3(config-if-GigabitEthernet 0/2)# no shutdown
ASBR3(config-if-GigabitEthernet 0/2)# exit
ASBR3(config)# interface GigabitEthernet 0/1
ASBR3(config-if-GigabitEthernet 0/1)# ip address 10.0.3.1 255.255.255.0
ASBR3(config-if-GigabitEthernet 0/1)# no shutdown
ASBR3(config-if-GigabitEthernet 0/1)# exit
ASBR3(config)# interface Loopback 0
ASBR3(config-if-Loopback 0)# ip address 7.7.7.7 255.255.255.255
ASBR3(config-if-Loopback 0)# exit
ASBR3(config)# router ospf 1
ASBR3(config-router)# network 7.7.7.7 0.0.0.0 area 0
ASBR3(config-router)# network 172.1.5.0 0.0.0.255 area 0
ASBR3(config-router)# exit
ASBR3(config)# router bgp 65010
ASBR3(config-router)# neighbor 8.8.8.8 remote-as 65010
ASBR3(config-router)# neighbor 8.8.8.8 update-source Loopback 0
ASBR3(config-router)# neighbor 8.8.8.8 next-hop-self
ASBR3(config-router)# exit
ASBR3(config)# recursive-route lookup lsp
ASBR3(config)# router bgp 65010
ASBR3(config-router)# neighbor 10.0.3.2 remote-as 100
ASBR3(config-router)# exit
```

- Configure ASBR4.

```
ASBR4> enable
ASBR4# configure terminal
ASBR4(config)# interface GigabitEthernet 0/2
ASBR4(config-if-GigabitEthernet 0/2)# ip address 172.1.6.2 255.255.255.0
ASBR4(config-if-GigabitEthernet 0/2)# no shutdown
ASBR4(config-if-GigabitEthernet 0/2)# exit
ASBR4(config)# interface GigabitEthernet 0/1
ASBR4(config-if-GigabitEthernet 0/1)# ip address 10.0.4.1 255.255.255.0
```

```
ASBR4(config-if-GigabitEthernet 0/1)# no shutdown
ASBR4(config-if-GigabitEthernet 0/1)# exit
ASBR4(config)# interface Loopback 0
ASBR4(config-if-Loopback 0)# ip address 8.8.8.8 255.255.255.255
ASBR4(config-if-Loopback 0)# exit
ASBR4(config)# router ospf 1
ASBR4(config-router)# network 8.8.8.8 0.0.0.0 area 0
ASBR4(config-router)# network 172.1.6.0 0.0.0.255 area 0
ASBR4(config-router)# exit
ASBR4(config)# router bgp 65010
ASBR4(config-router)# neighbor 7.7.7.7 remote-as 65010
ASBR4(config-router)# neighbor 7.7.7.7 update-source Loopback 0
ASBR4(config-router)# neighbor 7.7.7.7 next-hop-self
ASBR4(config-router)# exit
ASBR4(config)# recursive-route lookup lsp
ASBR4(config)# router bgp 65010
ASBR4(config-router)# neighbor 10.0.4.2 remote-as 100
ASBR4(config-router)# exit
```

- Configure the edge device on user network 1.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# interface GigabitEthernet 0/2
Hostname(config-if-GigabitEthernet 0/2)# ip address 10.0.3.2 255.255.255.0
Hostname(config-if-GigabitEthernet 0/2)# no shutdown
Hostname(config-if-GigabitEthernet 0/2)# exit
Hostname(config)# interface GigabitEthernet 0/1
Hostname(config-if-GigabitEthernet 0/1)# ip address 64.21.33.9 255.255.255.0
Hostname(config-if-GigabitEthernet 0/1)# no shutdown
Hostname(config-if-GigabitEthernet 0/1)# exit
Hostname(config)# router bgp 100
Hostname(config-router)# neighbor 10.0.3.1 remote-as 65010
Hostname(config-router)# network 64.21.33.0 mask 255.255.255.0
```

5. Verification

- The VRF routing table on PE1 of the first carrier contains only internal routes of the second carrier, but not external routes.

  Check the VRF routing table.

```
PE1# show ip route vrf vpn1
Routing Table: vpn1
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
```

```
       * - candidate default

Gateway of last resort is no set
B    3.3.3.3/32 [200/0] via 172.1.1.2, 00:00:07
C    172.1.1.0/24 is directly connected, GigabitEthernet 0/2, 00:00:02
L    172.1.1.1/32 is directly connected, GigabitEthernet 0/2, 00:00:02
B    172.1.2.0/24 [200/0] via 172.1.1.2, 00:00:07
B    172.1.4.0/24 [200/0] via 2.2.2.2, 00:00:30
```

Check the label information.

```
PE1# show bgp vpnv4 unicast vrf vpn1 labels
BGP table version is 1, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
            S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network         Next Hop       In Label/Out Label
Route Distinguisher: 65001:20 (Default for VRF vpn1)
*> 3.3.3.3/32      172.1.1.2        2048/1024
*> 172.1.2.0/24    172.1.1.2       2049/1025
*>i6.6.6.6/32      2.2.2.2          2050/2112
```

- The VRF routing table on PE2 of the first carrier contains only internal routes of the second carrier, but not external routes.

  Check the VRF routing table.

```
PE2# show ip route vrf vpn1
Routing Table: vpn1
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
B    3.3.3.3/32 [200/0] via 172.1.1.2, 00:00:07
C    172.1.1.0/24 is directly connected, GigabitEthernet 0/2, 00:00:02
L    172.1.1.1/32 is directly connected, GigabitEthernet 0/2, 00:00:02
B    172.1.2.0/24 [200/0] via 172.1.1.2, 00:00:07
B    172.1.4.0/24 [200/0] via 2.2.2.2, 00:00:30
```

Check the label information.

```
PE2# show bgp vpnv4 unicast vrf vpn1 labels
BGP table version is 1, local router ID is 1.1.1.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
            S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network         Next Hop       In Label/Out Label
Route Distinguisher: 65001:20 (Default for VRF vpn1)
*> 3.3.3.3/32      172.1.1.2        2048/1024
*> 172.1.2.0/24    172.1.1.2       2049/1025
*>i6.6.6.6/32      2.2.2.2          2050/2112
```

- Check the routing table on ASBR3.

```
ASBR3# show ip route
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
......
B    61.10.55.0/24 [200/0] via 4.4.4.4, 00:00:40
B    64.21.33.0/24 [200/0] via 10.0.3.2, 00:00:31
```

- Check the routing table on the edge device on user network 1.

```
Hostname# show ip route
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, A - Arp to host
       LA - Local aggregate route
       * - candidate default

Gateway of last resort is no set
......
B    61.10.55.0/24 [200/0] via 10.0.3.1, 00:00:40
C    64.21.33.0/24 is directly connected, GigabitEthernet 0/1, 00:00:02
L    64.21.33.9/32 is directly connected, GigabitEthernet 0/1, 00:00:02
```

- Ping the egress of ISP A from the edge device on user network 1.

```
Hostname # ping 61.10.55.1 source 64.21.33.9
Sending 5, 100-byte ICMP Echoes to 61.10.55.1, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/20/40 ms
```

6. Configuration Files

- PE1 configuration file

```
hostname PE1
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
mpls enable
!
mpls router ldp
 ldp rouer-id interface Loopback 0
!
interface GigabitEthernet 0/1
 ip address 192.168.1.1 255.255.255.0
 label-switching
 mpls ldp enable
 no shutdown
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
router bgp 65001
 neighbor 2.2.2.2 remote-as 65001
 neighbor 2.2.2.2 update-source Loopback 0
 !
 address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community both
 exit-address-family
!
ip vrf vpn1
 rd 65001:20
 route-target both 65001:20
 alloc-label per-route
!
interface loopback 1
 ip vrf forwarding vpn1
 ip address 10.1.1.1 255.255.255.255
 no shutdown
!
interface GigabitEthernet 0/2
 ip vrf forwarding vpn1
 ip address 172.1.1.1 255.255.255.0
```

```
 no shutdown
!
router bgp 65001
 address-family ipv4 vrf vpn1
  neighbor 172.1.1.2 remote-as 65010
  neighbor 172.1.1.2 as-override
  neighbor 172.1.1.2 send-label
 exit-address-family
!
```

- PE2 configuration file

```
hostname PE2
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
mpls enable
!
mpls router ldp
 ldp rouer-id interface Loopback 0
!
interface GigabitEthernet 0/1
 ip address 192.168.2.2 255.255.255.0
 label-switching
 mpls ldp enable
 no shutdown
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.2.0 0.0.0.255 area 0
!
router bgp 65001
 neighbor 1.1.1.1 remote-as 65001
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
!
ip vrf vpn1
 rd 65001:20
 route-target both 65001:20
 alloc-label per-route
!
interface loopback 1
 ip vrf forwarding vpn1
```

```
 ip address 10.1.2.1 255.255.255.255
 no shutdown
!
interface GigabitEthernet 0/2
 ip vrf forwarding vpn1
 ip address 172.1.4.1 255.255.255.0
 no shutdown
!
router bgp 65001
 !
 address-family ipv4 vrf vpn1
  neighbor 172.1.4.2 remote-as 65010
  neighbor 172.1.4.2 as-override
  neighbor 172.1.4.2 send-label
 exitaddress-family
!
```

- CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
 ip address 172.1.1.2 255.255.255.0
 no shutdown
!
router bgp 65010
 neighbor 172.1.1.2 remote-as 65001
 redistribute ospf 1
!
router ospf 1
 redistribute bgp subnets
!
interface GigabitEthernet 0/1
 label-switching
 ip address 172.1.1.2 255.255.255.0
 no shutdown
!
router bgp 65010
 neighbor 172.1.1.1 send-label
!
interface GigabitEthernet 0/2
 ip address 172.1.2.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/3
 ip address 172.1.3.1 255.255.255.0
 no shutdown
!
```

```
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 172.1.2.0 0.0.0.255 area 0
 network 172.1.3.0 0.0.0.255 area 0
!
```

- CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/1
 ip address 172.1.4.2 255.255.255.0
 no shutdown
!
router bgp 65010
 neighbor 172.1.4.2 remote-as 65001
 redistribute ospf 1
!
router ospf 1
 redistribute bgp subnets
!
interface GigabitEthernet 0/1
 label-switching
 ip address 172.1.5.1 255.255.255.0
 no shutdown
!
router bgp 65010
 neighbor 172.1.4.1 send-label
!
interface GigabitEthernet 0/2
 ip address 172.1.5.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/3
 ip address 172.1.6.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 6.6.6.6 255.255.255.255
!
router ospf 1
 network 6.6.6.6 0.0.0.0 area 0
 network 172.1.5.0 0.0.0.255 area 0
 network 172.1.6.0 0.0.0.255 area 0
!
```

- ASBR1 configuration file

```
hostname ASBR1
!
interface GigabitEthernet 0/2
 ip address 172.1.2.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.1.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router ospf 1
 network 4.4.4.4 0.0.0.0 area 0
 network 172.1.2.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 5.5.5.5 remote-as 65010
 neighbor 5.5.5.5 update-source Loopback 0
 neighbor 5.5.5.5 next-hop-self
!
recursive-route lookup lsp
!
router bgp 65010
 neighbor 10.0.1.2 remote-as 100
!
```

- ASBR2 configuration file

```
hostname ASBR2
!
interface GigabitEthernet 0/2
 ip address 172.1.3.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.2.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router ospf 1
 network 5.5.5.5 0.0.0.0 area 0
 network 172.1.3.0 0.0.0.255 area 0
!
```

```
router bgp 65010
 neighbor 4.4.4.4 remote-as 65010
 neighbor 4.4.4.4 update-source Loopback 0
 neighbor 4.4.4.4 next-hop-self
!
recursive-route lookup lsp
!
router bgp 65010
 neighbor 10.0.2.2 remote-as 100
!
```

- ASBR3 configuration file

```
hostname ASBR3
!
interface GigabitEthernet 0/2
 ip address 172.1.5.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.3.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 7.7.7.7 255.255.255.255
!
router ospf 1
 network 7.7.7.7 0.0.0.0 area 0
 network 172.1.5.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 8.8.8.8 remote-as 65010
 neighbor 8.8.8.8 update-source Loopback 0
 neighbor 8.8.8.8 next-hop-self
!
recursive-route lookup lsp
!
router bgp 65010
 neighbor 10.0.3.2 remote-as 100
!
```

- ASBR4 configuration file

```
hostname ASBR4
!
interface GigabitEthernet 0/2
 ip address 172.1.6.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
```

```
 ip address 10.0.4.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 8.8.8.8 255.255.255.255
!
router ospf 1
 network 8.8.8.8 0.0.0.0 area 0
 network 172.1.6.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 7.7.7.7 remote-as 65010
 neighbor 7.7.7.7 update-source Loopback 0
 neighbor 7.7.7.7 next-hop-self
!
recursive-route lookup lsp
!
router bgp 65010
 neighbor 10.0.4.2 remote-as 100
!
```

- User Network1 edge device configuration file

```
!
interface GigabitEthernet 0/2
 ip address 10.0.3.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 64.21.33.9 255.255.255.0
 no shutdown
!
router bgp 100
 neighbor 10.0.3.1 remote-as 65010
 network 64.21.33.0 mask 255.255.255.0
!
```

1.16.20 Configuring the Second Carrier to Provide the VPN Service Based on the MPLS Core

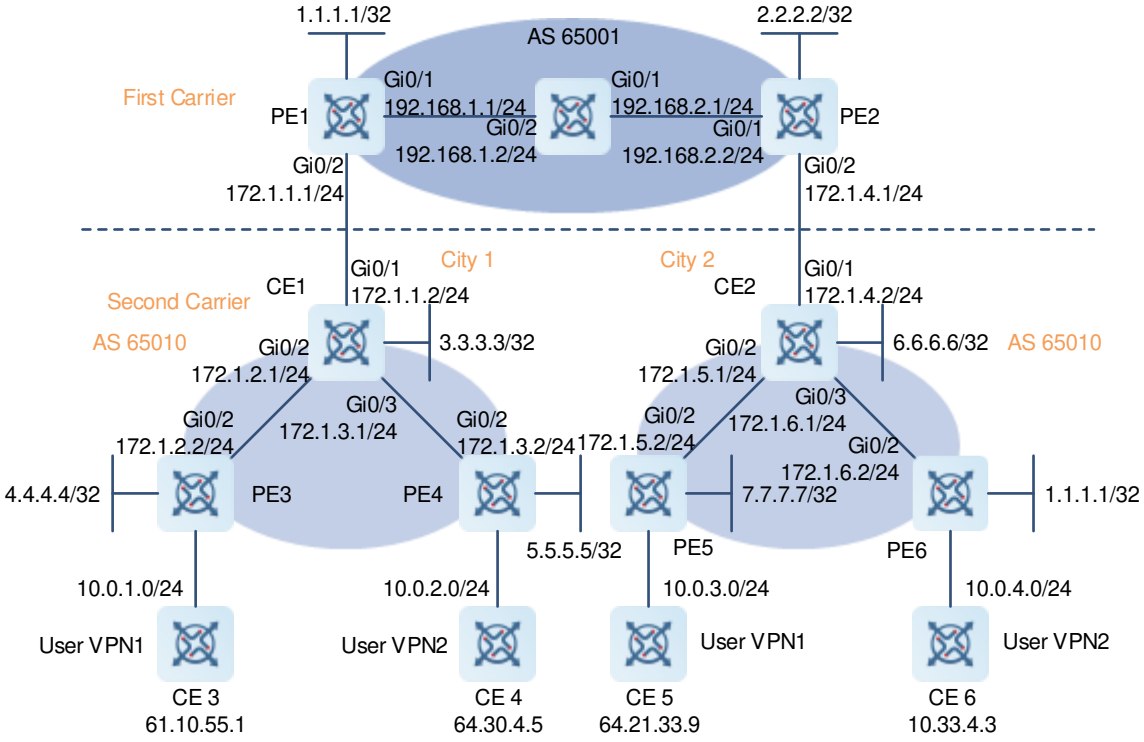
1. Requirements

The carrier has an MPLS core network in city 1, which provide the MPLS VPN services to users in this city. To expand services to city 2, the carrier deploys an MPLS core network in city 2. To implement interconnection between core networks in two cities, this carrier rents the VPN service from another MPLS VPN provider. This networking is a CSC model.

After the networking, the PEs of the first carrier exchange internal routes with the CEs of the second carrier through BGP and MP-IBGP neighbor relationships are established directly between the PEs of the second carrier to exchange VPN routes of users. The PEs of the second carrier exchange routes with the VPN CEs of users through OSPF.

2. Topology

Figure 1-3 Configuring the Second Carrier to Provide the VPN Service Based on the MPLS Core



3. Notes

(1) Configure basic BGP/MPLS VPN features for the first carrier: Configure loopback interfaces and configure MPLS and LDP globally and on interfaces. Configure IGP (OSPF), MP-IBGP neighbors, and VRF instances, connect CEs to PEs, and configure PEs and CEs to exchange routes.

(2) Configure the CSC feature: On the PEs, configure the CSC feature and distribute MPLS labels to IPv4 routes. On the CEs, configure MPLS and LDP.

(3) Configure the second carrier: Configure interfaces and IGP. On each ASBR, configure the CE as its BGP peer. On a CE, configure the corresponding ASBR and the CE in another site as the RR clients and parse the next hops in BGP routes to LSPs.

(4) Configure user access: Configure a VRF instance, bind the VRF instance to an interface, and configure OSPF for route distribution to implement VPN access.

4. Procedure

- Configure PE1.

```
PE1> enable
PE1# configure terminal
PE1(config)# interface Loopback 0
PE1(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
```

```
PE1(config-if-Loopback 0)# exit
PE1(config)# mpls enable
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface Loopback 0
PE1(config-mpls-router)# exit
PE1(config)# interface GigabitEthernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# label-switching
PE1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/1)# no shutdown
PE1(config-if-GigabitEthernet 0/1)# exit
PE1(config)# router ospf 1
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
PE1(config-router)# network 192.168.1.0 0.0.0.255 area 0
PE1(config-router)# exit
PE1(config)# router bgp 65001
PE1(config-router)# neighbor 2.2.2.2 remote-as 65001
PE1(config-router)# neighbor 2.2.2.2 update-source Loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 2.2.2.2 activate
PE1(config-router-af)# neighbor 2.2.2.2 send-community both
PE1(config-router-af)# exit
PE1(config-router)# exit
PE1(config)# ip vrf vpn1
PE1(config-vrf)# rd 65001:20
PE1(config-vrf)# route-target both 65001:20
PE1(config-vrf)# alloc-label per-route
PE1(config-vrf)# exit
PE1(config)# interface loopback 1
PE1(config-if-Loopback 1)# ip vrf forwarding vpn1
PE1(config-if-Loopback 1)# ip address 10.1.1.1 255.255.255.255
PE1(config-if-Loopback 1)# no shutdown
PE1(config-if-Loopback 1)# exit
PE1(config)# interface GigabitEthernet 0/2
PE1(config-if-GigabitEthernet 0/2)# ip vrf forwarding vpn1
PE1(config-if-GigabitEthernet 0/2)# ip address 172.1.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/2)# no shutdown
PE1(config-if-GigabitEthernet 0/2)# exit
PE1(config)# router bgp 65001
PE1(config-router)# address-family ipv4 vrf vpn1
PE1(config-router-af)# neighbor 172.1.1.2 remote-as 65010
PE1(config-router-af)# neighbor 172.1.1.2 as-override
PE1(config-router-af)# neighbor 172.1.1.2 send-label
PE1(config-router-af)# exit
PE1(config-router)# exit
```

- Configure PE2.

```
PE2> enable
PE2# configure terminal
PE2(config)# interface Loopback 0
PE2(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
PE2(config-if-Loopback 0)# exit
PE2(config)# mpls enable
PE2(config)# mpls router ldp
PE2(config-mpls-router)# ldp router-id interface Loopback 0
PE2(config-mpls-router)# exit
PE2(config)# interface GigabitEthernet 0/1
PE2(config-if-GigabitEthernet 0/1)# ip address 192.168.2.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/1)# label-switching
PE2(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE2(config-if-GigabitEthernet 0/1)# no shutdown
PE2(config-if-GigabitEthernet 0/1)# exit
PE2(config)# router ospf 1
PE2(config-router)# network 2.2.2.2 0.0.0.0 area 0
PE2(config-router)# network 192.168.2.0 0.0.0.255 area 0
PE2(config-router)# exit
PE2(config)# router bgp 65001
PE2(config-router)# neighbor 1.1.1.1 remote-as 65001
PE2(config-router)# neighbor 1.1.1.1 update-source Loopback 0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 1.1.1.1 activate
PE2(config-router-af)# neighbor 1.1.1.1 send-community both
PE2(config)# ip vrf vpn1
PE2(config-vrf)# rd 65001:20
PE2(config-vrf)# route-target both 65001:20
PE2(config-vrf)# alloc-label per-route
PE2(config-vrf)# exit
PE2(config)# interface loopback 1
PE2(config-if-Loopback 1)# ip vrf forwarding vpn1
PE2(config-if-Loopback 1)# ip address 10.1.2.1 255.255.255.255
PE2(config-if-Loopback 1)# no shutdown
PE2(config-if-Loopback 1)# exit
PE2(config)# interface GigabitEthernet 0/2
PE2(config-if-GigabitEthernet 0/2)# ip vrf forwarding vpn1
PE2(config-if-GigabitEthernet 0/2)# ip address 172.1.4.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/2)# no shutdown
PE2(config-if-GigabitEthernet 0/2)# exit
PE2(config)# router bgp 65001
PE2(config-router)# address-family ipv4 vrf vpn1
PE2(config-router-af)# neighbor 172.1.4.2 remote-as 65010
PE2(config-router-af)# neighbor 172.1.4.2 as-override
PE2(config-router-af)# neighbor 172.1.4.2 send-label
PE2(config-router-af)# exit
```

```
PE2(config-router)# exit
```

- Configure CE1.

```
CE1> enable
CE1# configure terminal
CE1(config)# interface GigabitEthernet 0/1
CE1(config-if-GigabitEthernet 0/1)# ip address 172.1.1.2 255.255.255.0
CE1(config-if-GigabitEthernet 0/1)# no shutdown
CE1(config-if-GigabitEthernet 0/1)# exit
CE1(config)# interface GigabitEthernet 0/2
CE1(config-if-GigabitEthernet 0/2)# ip address 172.1.2.1 255.255.255.0
CE1(config-if-GigabitEthernet 0/2)# no shutdown
CE1(config-if-GigabitEthernet 0/2)# exit
CE1(config)# interface GigabitEthernet 0/3
CE1(config-if-GigabitEthernet 0/3)# ip address 172.1.3.1 255.255.255.0
CE1(config-if-GigabitEthernet 0/3)# no shutdown
CE1(config-if-GigabitEthernet 0/3)# exit
CE1(config)# interface Loopback 0
CE1(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
CE1(config-if-Loopback 0)# exit
CE1(config)# router ospf 1
CE1(config-router)# network 3.3.3.3 0.0.0.0 area 0
CE1(config-router)# network 172.1.2.0 0.0.0.255 area 0
CE1(config-router)# network 172.1.3.0 0.0.0.255 area 0
CE1(config-router)# exit
CE1(config)# router bgp 65010
CE1(config-router)# neighbor 4.4.4.4 remote-as 65010
CE1(config-router)# neighbor 4.4.4.4 update-source Loopback 0
CE1(config-router)# neighbor 4.4.4.4 route-reflector-client
CE1(config-router)# neighbor 5.5.5.5 remote-as 65010
CE1(config-router)# neighbor 5.5.5.5 update-source Loopback 0
CE1(config-router)# neighbor 5.5.5.5 route-reflector-client
CE1(config-router)# neighbor 6.6.6.6 remote-as 65010
CE1(config-router)# neighbor 6.6.6.6 update-source Loopback 0
CE1(config-router)# neighbor 6.6.6.6 route-reflector-client
CE1(config-router)# exit
CE1(config)# recursive-route lookup lsp
```

- Configure CE2.

```
CE2> enable
CE2# configure terminal
CE2(config)# interface GigabitEthernet 0/1
CE2(config-if-GigabitEthernet 0/1)# ip address 172.1.4.2 255.255.255.0
CE2(config-if-GigabitEthernet 0/1)# no shutdown
CE2(config-if-GigabitEthernet 0/1)# exit
CE2(config)# interface GigabitEthernet 0/2
CE2(config-if-GigabitEthernet 0/2)# ip address 172.1.5.1 255.255.255.0
CE2(config-if-GigabitEthernet 0/2)# no shutdown
```

```
CE2(config-if-GigabitEthernet 0/2)# exit
CE2(config)# interface GigabitEthernet 0/3
CE2(config-if-GigabitEthernet 0/3)# ip address 172.1.6.1 255.255.255.0
CE2(config-if-GigabitEthernet 0/3)# no shutdown
CE2(config-if-GigabitEthernet 0/3)# exit
CE2(config)# interface Loopback 0
CE2(config-if-Loopback 0)# ip address 6.6.6.6 255.255.255.255
CE2(config-if-Loopback 0)# exit
CE2(config)# router ospf 1
CE2(config-router)# network 6.6.6.6 0.0.0.0 area 0
CE2(config-router)# network 172.1.5.0 0.0.0.255 area 0
CE2(config-router)# network 172.1.6.0 0.0.0.255 area 0
CE2(config-router)# exit
CE2(config)# router bgp 65010
CE2(config-router)# neighbor 3.3.3.3 remote-as 65010
CE2(config-router)# neighbor 3.3.3.3 update-source Loopback 0
CE2(config-router)# neighbor 3.3.3.3 route-reflector-client
CE2(config-router)# neighbor 8.8.8.8 remote-as 65010
CE2(config-router)# neighbor 8.8.8.8 update-source Loopback 0
CE2(config-router)# neighbor 8.8.8.8 route-reflector-client
CE2(config-router)# neighbor 9.9.9.9 remote-as 65010
CE2(config-router)# neighbor 9.9.9.9 update-source Loopback 0
CE2(config-router)# neighbor 9.9.9.9 route-reflector-client
CE2(config-router)# exit
CE2(config)# recursive-route lookup lsp
```

- Configure CE3.

```
CE3> enable
CE3# configure terminal
CE3(config)# interface GigabitEthernet 0/2
CE3(config-if-GigabitEthernet 0/2)# ip address 10.0.1.2 255.255.255.0
CE3(config-if-GigabitEthernet 0/2)# no shutdown
CE3(config)# interface GigabitEthernet 0/1
CE3(config-if-GigabitEthernet 0/1)# ip address 61.10.55.1 255.255.255.0
CE3(config-if-GigabitEthernet 0/1)# no shutdown
CE3(config-if-GigabitEthernet 0/1)# exit
CE3(config)# router ospf 1
CE3(config-router)# network 10.0.1.0 0.0.0.255 area 0
CE3(config-router)# network 61.10.55.0 0.0.0.255 area 0
CE3(config-router)# exit
```

- Configure CE4.

```
CE4> enable
CE4# configure terminal
CE4(config)# interface GigabitEthernet 0/2
CE4(config-if-GigabitEthernet 0/2)# ip address 10.0.2.2 255.255.255.0
CE4(config-if-GigabitEthernet 0/2)# no shutdown
CE4(config)# interface GigabitEthernet 0/1
```

```
CE4(config-if-GigabitEthernet 0/1)# ip address 64.30.4.5 255.255.255.0
CE4(config-if-GigabitEthernet 0/1)# no shutdown
CE4(config-if-GigabitEthernet 0/1)# exit
CE4(config)# router ospf 1
CE4(config-router)# network 10.0.2.0 0.0.0.255 area 0
CE4(config-router)# network 64.30.4.0 0.0.0.255 area 0
CE4(config-router)# exit
```

- Configure CE5.

```
CE5> enable
CE5# configure terminal
CE5(config)# interface GigabitEthernet 0/2
CE5(config-if-GigabitEthernet 0/2)# ip address 10.0.3.2 255.255.255.0
CE5(config-if-GigabitEthernet 0/2)# no shutdown
CE5(config)# interface GigabitEthernet 0/1
CE5(config-if-GigabitEthernet 0/1)# ip address 64.21.33.9 255.255.255.0
CE5(config-if-GigabitEthernet 0/1)# no shutdown
CE5(config-if-GigabitEthernet 0/1)# exit
CE5(config)# router ospf 1
CE5(config-router)# network 10.0.3.0 0.0.0.255 area 0
CE5(config-router)# network 64.21.33.0 0.0.0.255 area 0
CE5(config-router)# exit
```

- Configure CE6.

```
CE6> enable
CE6# configure terminal
CE6(config)# interface GigabitEthernet 0/2
CE6(config-if-GigabitEthernet 0/2)# ip address 10.0.4.2 255.255.255.0
CE6(config-if-GigabitEthernet 0/2)# no shutdown
CE6(config)# interface GigabitEthernet 0/1
CE6(config-if-GigabitEthernet 0/1)# ip address 10.33.4.3 255.255.255.0
CE6(config-if-GigabitEthernet 0/1)# no shutdown
CE6(config-if-GigabitEthernet 0/1)# exit
CE6(config)# router ospf 1
CE6(config-router)# network 10.0.4.0 0.0.0.255 area 0
CE6(config-router)# network 10.33.4.0 0.0.0.255 area 0
CE6(config-router)# exit
```

- Configure PE3.

```
PE3> enable
PE3# configure terminal
PE3(config)# interface GigabitEthernet 0/2
PE3(config-if-GigabitEthernet 0/2)# ip address 172.1.2.2 255.255.255.0
PE3(config-if-GigabitEthernet 0/2)# no shutdown
PE3(config-if-GigabitEthernet 0/2)# exit
PE3(config)# interface GigabitEthernet 0/1
PE3(config-if-GigabitEthernet 0/1)# ip address 10.0.1.1 255.255.255.0
PE3(config-if-GigabitEthernet 0/1)# no shutdown
PE3(config-if-GigabitEthernet 0/1)# exit
```

```
PE3(config)# interface Loopback 0
PE3(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
PE3(config-if-Loopback 0)# exit
PE3(config)# router ospf 1
PE3(config-router)# network 4.4.4.4 0.0.0.0 area 0
PE3(config-router)# network 172.1.2.0 0.0.0.255 area 0
PE3(config-router)# exit
PE3(config)# router bgp 65010
PE3(config-router)# neighbor 3.3.3.3 remote-as 65010
PE3(config-router)# neighbor 3.3.3.3 update-source Loopback 0
PE3(config-router)# neighbor 3.3.3.3 next-hop-self
PE3(config-router)# neighbor 5.5.5.5 remote-as 65010
PE3(config-router)# neighbor 5.5.5.5 update-source Loopback 0
PE3(config-router)# neighbor 5.5.5.5 route-reflector-client
PE3(config-router)# exit
PE3(config)# recursive-route lookup lsp
PE3(config)# ip vrf customer_vpn1
PE3(config-vrf)# rd 65010:1
PE3(config-vrf)# route-target both 65010:1
PE3(config-vrf)# exit
PE3(config)# interface GigabitEthernet 0/1
PE3(config-if-GigabitEthernet 0/1)# ip vrf forwarding customer_vpn1
PE3(config-if-GigabitEthernet 0/1)# ip address 10.0.1.1 255.255.255.0
PE3(config-if-GigabitEthernet 0/1)# no shutdown
PE3(config-if-GigabitEthernet 0/1)# exit
PE3(config)# router ospf 10 vrf customer_vpn1
PE3(config-router)# network 10.0.1.0 0.0.0.255 area 0
PE3(config-router)# redistribute bgp subnets
PE3(config-router)# exit
PE3(config)# router bgp 65010
PE3(config-router)# address-family ipv4 vrf customer_vpn1
PE3(config-router-af)# redistribute ospf 10
PE3(config-router-af)# exit
PE3(config-router)# exit
```

- Configure PE4.

```
PE4> enable
PE4# configure terminal
PE4(config)# interface GigabitEthernet 0/2
PE4(config-if-GigabitEthernet 0/2)# ip address 172.1.3.1 255.255.255.0
PE4(config-if-GigabitEthernet 0/2)# no shutdown
PE4(config-if-GigabitEthernet 0/2)# exit
PE4(config)# interface GigabitEthernet 0/1
PE4(config-if-GigabitEthernet 0/1)# ip address 10.0.2.1 255.255.255.0
PE4(config-if-GigabitEthernet 0/1)# no shutdown
PE4(config-if-GigabitEthernet 0/1)# exit
PE4(config)# interface Loopback 0
```

```
PE4(config-if-Loopback 0)# ip address 5.5.5.5 255.255.255.255
PE4(config-if-Loopback 0)# exit
PE4(config)# router ospf 1
PE4(config-router)# network 5.5.5.5 0.0.0.0 area 0
PE4(config-router)# network 172.1.3.0 0.0.0.255 area 0
PE4(config-router)# exit
PE4(config)# router bgp 65010
PE4(config-router)# neighbor 3.3.3.3 remote-as 65010
PE4(config-router)# neighbor 3.3.3.3 update-source Loopback 0
PE4(config-router)# neighbor 3.3.3.3 next-hop-self
PE4(config-router)# neighbor 4.4.4.4 remote-as 65010
PE4(config-router)# neighbor 4.4.4.4 update-source Loopback 0
PE4(config-router)# neighbor 4.4.4.4 route-reflector-client
PE4(config-router)# exit
PE4(config)# recursive-route lookup lsp
PE4(config)# ip vrf customer_vpn1
PE4(config-vrf)# rd 65010:1
PE4(config-vrf)# route-target both 65010:1
PE4(config-vrf)# exit
PE4(config)# interface GigabitEthernet 0/1
PE4(config-if-GigabitEthernet 0/1)# ip vrf forwarding customer_vpn1
PE4(config-if-GigabitEthernet 0/1)# ip address 10.0.2.1 255.255.255.0
PE4(config-if-GigabitEthernet 0/1)# no shutdown
PE4(config-if-GigabitEthernet 0/1)# exit
PE4(config)# router ospf 10 vrf customer_vpn1
PE4(config-router)# network 10.0.2.0 0.0.0.255 area 0
PE4(config-router)# redistribute bgp subnets
PE4(config-router)# exit
PE4(config)# router bgp 65010
PE4(config-router)# address-family ipv4 vrf customer_vpn1
PE4(config-router-af)# redistribute ospf 10
PE4(config-router-af)# exit
PE4(config-router)# exit
```

- Configure PE5.

```
PE5> enable
PE5# configure terminal
```

```
PE5(config)# interface GigabitEthernet 0/2
PE5(config-if-GigabitEthernet 0/2)# ip address 172.1.5.1 255.255.255.0
PE5(config-if-GigabitEthernet 0/2)# no shutdown
PE5(config-if-GigabitEthernet 0/2)# exit
PE5(config)# interface GigabitEthernet 0/1
PE5(config-if-GigabitEthernet 0/1)# ip address 10.0.3.1 255.255.255.0
PE5(config-if-GigabitEthernet 0/1)# no shutdown
PE5(config-if-GigabitEthernet 0/1)# exit
PE5(config)# interface Loopback 0
PE5(config-if-Loopback 0)# ip address 7.7.7.7 255.255.255.255
PE5(config-if-Loopback 0)# exit
PE5(config)# router ospf 1
PE5(config-router)# network 7.7.7.7 0.0.0.0 area 0
PE5(config-router)# network 172.1.5.0 0.0.0.255 area 0
PE5(config-router)# exit
PE5(config)# router bgp 65010
PE5(config-router)# neighbor 6.6.6.6 remote-as 65010
PE5(config-router)# neighbor 6.6.6.6 update-source Loopback 0
PE5(config-router)# neighbor 6.6.6.6 next-hop-self
PE5(config-router)# neighbor 8.8.8.8 remote-as 65010
PE5(config-router)# neighbor 8.8.8.8 update-source Loopback 0
PE5(config-router)# neighbor 8.8.8.8 route-reflector-client
PE5(config-router)# exit
PE5(config)# recursive-route lookup lsp
PE5(config)# ip vrf customer_vpn1
PE5(config-vrf)# rd 65010:1
PE5(config-vrf)# route-target both 65010:1
PE5(config-vrf)# exit
PE5(config)# interface GigabitEthernet 0/1
PE5(config-if-GigabitEthernet 0/1)# ip vrf forwarding customer_vpn1
PE5(config-if-GigabitEthernet 0/1)# ip address 10.0.3.1 255.255.255.0
PE5(config-if-GigabitEthernet 0/1)# no shutdown
PE5(config-if-GigabitEthernet 0/1)# exit
PE5(config)# router ospf 10 vrf customer_vpn1
PE5(config-router)# network 10.0.3.0 0.0.0.255 area 0
PE5(config-router)# redistribute bgp subnets
PE5(config-router)# exit
PE5(config)# router bgp 65010
PE5(config-router)# address-family ipv4 vrf customer_vpn1
PE5(config-router-af)# redistribute ospf 10
PE5(config-router-af)# exit
PE5(config-router)# exit
```

- Configure PE6.

```
PE6> enable
PE6# configure terminal
PE6(config)# interface GigabitEthernet 0/2
```

```
PE6(config-if-GigabitEthernet 0/2)# ip address 172.1.6.1 255.255.255.0
PE6(config-if-GigabitEthernet 0/2)# no shutdown
PE6(config-if-GigabitEthernet 0/2)# exit
PE6(config)# interface GigabitEthernet 0/1
PE6(config-if-GigabitEthernet 0/1)# ip address 10.0.4.1 255.255.255.0
PE6(config-if-GigabitEthernet 0/1)# no shutdown
PE6(config-if-GigabitEthernet 0/1)# exit
PE6(config)# interface Loopback 0
PE6(config-if-Loopback 0)# ip address 8.8.8.8 255.255.255.255
PE6(config-if-Loopback 0)# exit
PE6(config)# router ospf 1
PE6(config-router)# network 8.8.8.8 0.0.0.0 area 0
PE6(config-router)# network 172.1.6.0 0.0.0.255 area 0
PE6(config-router)# exit
PE6(config)# router bgp 65010
PE6(config-router)# neighbor 6.6.6.6 remote-as 65010
PE6(config-router)# neighbor 6.6.6.6 update-source Loopback 0
PE6(config-router)# neighbor 6.6.6.6 next-hop-self
PE6(config-router)# neighbor 7.7.7.7 remote-as 65010
PE6(config-router)# neighbor 7.7.7.7 update-source Loopback 0
PE6(config-router)# neighbor 7.7.7.7 route-reflector-client
PE6(config-router)# exit
PE6(config)# recursive-route lookup lsp
PE6(config)# ip vrf customer_vpn1
PE6(config-vrf)# rd 65010:1
PE6(config-vrf)# route-target both 65010:1
PE6(config-vrf)# exit
PE6(config)# interface GigabitEthernet 0/1
PE6(config-if-GigabitEthernet 0/1)# ip vrf forwarding customer_vpn1
PE6(config-if-GigabitEthernet 0/1)# ip address 10.0.4.1 255.255.255.0
PE6(config-if-GigabitEthernet 0/1)# no shutdown
PE6(config-if-GigabitEthernet 0/1)# exit
PE6(config)# router ospf 10 vrf customer_vpn1
PE6(config-router)# network 10.0.4.0 0.0.0.255 area 0
PE6(config-router)# redistribute bgp subnets
PE6(config-router)# exit
PE6(config)# router bgp 65010
PE6(config-router)# address-family ipv4 vrf customer_vpn1
PE6(config-router-af)# redistribute ospf 10
PE6(config-router-af)# exit
PE6(config-router)# exit
```

5. Verification

- Check the route and label information of the VRF instance on PEs of the first carrier. The VRF routing table contains only internal routes of the second carrier, but not the VPN routes.

  Check the VRF routing table on PE1.

```
PE1# show ip route vrf vpn1
Routing Table: vpn1

Codes:    C - Connected, L - Local, S - Static
          R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
          N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
          E1 - OSPF external type 1, E2 - OSPF external type 2
          SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
          IA - Inter area, EV - BGP EVPN, A - Arp to host
          LA - Local aggregate route
          * - candidate default

Gateway of last resort is no set
B       3.3.3.3/32 [200/0] via 172.1.1.2, 00:00:07
C       172.1.1.0/24 is directly connected, GigabitEthernet 0/2, 00:00:02
L       172.1.1.1/32 is directly connected, GigabitEthernet 0/2, 00:00:02
B       172.1.2.0/24 [200/0] via 172.1.1.2, 00:00:07
B       172.1.4.0/24 [200/0] via 2.2.2.2, 00:00:30
```

Check the VRF label information on PE1.

```
PE1# show bgp vpnv4 unicast vrf vpn1 labels
BGP table version is 1, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
                    S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

    Network              Next Hop            In Label/Out Label
Route Distinguisher: 65001:20 (Default for VRF vpn1)
*> 3.3.3.3/32           172.1.1.2             2048/1024
*> 172.1.2.0/24        172.1.1.2            2049/1025
*>i6.6.6.6/32          2.2.2.2               2050/2112
```

Check the VRF routing table on PE2.

```
PE2# show ip route vrf vpn1
Routing Table: vpn1

Codes:    C - Connected, L - Local, S - Static
          R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
          N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
          E1 - OSPF external type 1, E2 - OSPF external type 2
          SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
          IA - Inter area, EV - BGP EVPN, A - Arp to host
          LA - Local aggregate route
          * - candidate default

Gateway of last resort is no set
B       3.3.3.3/32 [200/0] via 172.1.1.2, 00:00:07
```

```
C       172.1.1.0/24 is directly connected, GigabitEthernet 0/2, 00:00:02
L       172.1.1.1/32 is directly connected, GigabitEthernet 0/2, 00:00:02
B       172.1.2.0/24 [200/0] via 172.1.1.2, 00:00:07
B       172.1.4.0/24 [200/0] via 2.2.2.2, 00:00:30
```

Check the VRF label information on PE2.

```
PE2# show bgp vpnv4 unicast vrf vpn1 labels
BGP table version is 1, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
                    S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

    Network             Next Hop            In Label/Out Label
Route Distinguisher: 65001:20 (Default for VRF vpn1)
*> 3.3.3.3/32          172.1.1.2            2048/1024
*> 172.1.2.0/24        172.1.1.2           2049/1025
*>i6.6.6.6/32          2.2.2.2              2050/2112
```

- Check the routing table of the VRF instance on the PE of the second carrier and check the routing table on the user VPN CE and confirm that user VPNs are reachable to each other.

  Check the VRF routing table on PE3.

```
PE3# show ip route vrf customer_vpn1
Routing Table: customer_vpn1

Codes:   C - Connected, L - Local, S - Static
         R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
         N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
         E1 - OSPF external type 1, E2 - OSPF external type 2
         SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
         IA - Inter area, EV - BGP EVPN, A - Arp to host
         LA - Local aggregate route
         * - candidate default

Gateway of last resort is no set
......
O       61.10.55.0/24 [200/0] via 10.0.1.2, GigabitEthernet 0/2, 00:00:40
B       64.21.33.0/24 [200/0] via 7.7.7.7, 00:00:31
```

Check the routing table on CE3.

```
CE3# show ip route
Codes:   C - Connected, L - Local, S - Static
         R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
         N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
         E1 - OSPF external type 1, E2 - OSPF external type 2
         SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
         IA - Inter area, EV - BGP EVPN, A - Arp to host
         LA - Local aggregate route
         * - candidate default
```

```
Gateway of last resort is no set
......
C       61.10.55.0/24 is directly connected, GigabitEthernet 0/1, 00:00:02
L       61.10.55.1/32 is directly connected, GigabitEthernet 0/1, 00:00:02
O       64.21.33.0/24 [200/0] via 10.0.1.1, GigabitEthernet 0/1, 00:00:42
```

Ping the same VPN in city 2 from CE3.

```
CE3# ping 64.21.33.9
Sending 5, 100-byte ICMP Echoes to 64.21.33.9, timeout is 2 seconds:
   < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/20/40 ms
```

6. Configuration Files

- PE1 configuration file

```
hostname PE1
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
mpls enable
!
mpls router ldp
 ldp rouer-id interface Loopback 0
!
interface GigabitEthernet 0/1
 ip address 192.168.1.1 255.255.255.0
 label-switching
 mpls ldp enable
 no shutdown
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
router bgp 65001
 neighbor 2.2.2.2 remote-as 65001
 neighbor 2.2.2.2 update-source Loopback 0
 !
 address-family vpnv4
```

```
 neighbor 2.2.2.2 activate
 neighbor 2.2.2.2 send-community both
exit-address-family
!
ip vrf vpn1
rd 65001:20
route-target both 65001:20
alloc-label per-route
!
interface loopback 1
ip vrf forwarding vpn1
ip address 10.1.1.1 255.255.255.255
no shutdown
!
interface GigabitEthernet 0/2
ip vrf forwarding vpn1
ip address 172.1.1.1 255.255.255.0
no shutdown
!
router bgp 65001
address-family ipv4 vrf vpn1
neighbor 172.1.1.2 remote-as 65010
neighbor 172.1.1.2 as-override
neighbor 172.1.1.2 send-label
exit-address-family
!
```

- PE2 configuration file

```
hostname PE2
!
interface Loopback 0
ip address 2.2.2.2 255.255.255.255
!
mpls enable
!
mpls router ldp
ldp rouer-id interface Loopback 0
!
interface GigabitEthernet 0/1
```

```
 ip address 192.168.2.2 255.255.255.0
 label-switching
 mpls ldp enable
 no shutdown
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.2.0 0.0.0.255 area 0
!
router bgp 65001
 neighbor 1.1.1.1 remote-as 65001
 neighbor 1.1.1.1 update-source Loopback 0
 !
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
!
ip vrf vpn1
 rd 65001:20
 route-target both 65001:20
 alloc-label per-route
!
interface loopback 1
 ip vrf forwarding vpn1
 ip address 10.1.2.1 255.255.255.255
 no shutdown
!
interface GigabitEthernet 0/2
 ip vrf forwarding vpn1
 ip address 172.1.4.1 255.255.255.0
 no shutdown
!
router bgp 65001
 address-family ipv4 vrf vpn1
  neighbor 172.1.4.2 remote-as 65010
  neighbor 172.1.4.2 as-override
  neighbor 172.1.4.2 send-label
```

```
 exit-adress-family
!
```

- CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
 ip address 172.1.1.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/2
 ip address 172.1.2.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/3
 ip address 172.1.3.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 172.1.2.0 0.0.0.255 area 0
 network 172.1.3.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 4.4.4.4 remote-as 65010
 neighbor 4.4.4.4 update-source Loopback 0
 neighbor 4.4.4.4 route-reflector-client
 neighbor 5.5.5.5 remote-as 65010
 neighbor 5.5.5.5 update-source Loopback 0
 neighbor 5.5.5.5 route-reflector-client
 neighbor 6.6.6.6 remote-as 65010
 neighbor 6.6.6.6 update-source Loopback 0
 neighbor 6.6.6.6 route-reflector-client
!
recursive-route lookup lsp
!
```

- CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/1
 ip address 172.1.4.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/2
 ip address 172.1.5.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/3
 ip address 172.1.6.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 6.6.6.6 255.255.255.255
!
router ospf 1
 network 6.6.6.6 0.0.0.0 area 0
 network 172.1.5.0 0.0.0.255 area 0
 network 172.1.6.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 3.3.3.3 remote-as 65010
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 3.3.3.3 route-reflector-client
 neighbor 8.8.8.8 remote-as 65010
 neighbor 8.8.8.8 update-source Loopback 0
 neighbor 8.8.8.8 route-reflector-client
 neighbor 9.9.9.9 remote-as 65010
 neighbor 9.9.9.9 update-source Loopback 0
 neighbor 9.9.9.9 route-reflector-client
!
recursive-route lookup lsp
!
```

- CE3 configuration file

```
hostname CE3
```

```
!
interface GigabitEthernet 0/2
 ip address 10.0.1.2 255.255.255.0
!
interface GigabitEthernet 0/1
 ip address 61.10.55.1 255.255.255.0
 no shutdown
!
router ospf 1
 network 10.0.1.0 0.0.0.255 area 0
 network 61.10.55.0 0.0.0.255 area 0
!
```

- CE4 configuration file

```
hostname CE4
!
interface GigabitEthernet 0/2
 ip address 10.0.2.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 64.30.4.5 255.255.255.0
 no shutdown
!
router ospf 1
 network 10.0.2.0 0.0.0.255 area 0
 network 64.30.4.0 0.0.0.255 area 0
!
```

- CE5 configuration file

```
hostname CE5
!
interface GigabitEthernet 0/2
 ip address 10.0.3.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 64.21.33.9 255.255.255.0
 no shutdown
!
```

```
router ospf 1
 network 10.0.3.0 0.0.0.255 area 0
 network 64.21.33.0 0.0.0.255 area 0
!
```

- CE6 configuration file

```
hostname CE6
!
interface GigabitEthernet 0/2
 ip address 10.0.4.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.33.4.3 255.255.255.0
 no shutdown
!
router ospf 1
 network 10.0.4.0 0.0.0.255 area 0
 network 10.33.4.0 0.0.0.255 area 0
!
```

- PE3 configuration file

```
hostname PE3
!
interface GigabitEthernet 0/2
 ip address 172.1.2.2 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.1.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 4.4.4.4 255.255.255.255
!
router ospf 1
 network 4.4.4.4 0.0.0.0 area 0
 network 172.1.2.0 0.0.0.255 area 0
!
router bgp 65010
```

```
 neighbor 3.3.3.3 remote-as 65010
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 3.3.3.3 next-hop-self
 neighbor 5.5.5.5 remote-as 65010
 neighbor 5.5.5.5 update-source Loopback 0
 neighbor 5.5.5.5 route-reflector-client
!
recursive-route lookup lsp
!
ip vrf customer_vpn1
 rd 65010:1
 route-target both 65010:1
!
interface GigabitEthernet 0/1
 ip vrf forwarding customer_vpn1
 ip address 10.0.1.1 255.255.255.0
 no shutdown
!
router ospf 10 vrf customer_vpn1
 network 10.0.1.0 0.0.0.255 area 0
 redistribute bgp subnets
!
router bgp 65010
 address-family ipv4 vrf customer_vpn1
  redistribute ospf 10
 exit-address-family
!
```

- PE4 configuration file

```
hostname PE4
!
interface GigabitEthernet 0/2
 ip address 172.1.3.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.2.1 255.255.255.0
 no shutdown
 exit
```

```
!
interface Loopback 0
 ip address 5.5.5.5 255.255.255.255
!
router ospf 1
 network 5.5.5.5 0.0.0.0 area 0
 network 172.1.3.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 3.3.3.3 remote-as 65010
 neighbor 3.3.3.3 update-source Loopback 0
 neighbor 3.3.3.3 next-hop-self
 neighbor 4.4.4.4 remote-as 65010
 neighbor 4.4.4.4 update-source Loopback 0
 neighbor 4.4.4.4 route-reflector-client
!
recursive-route lookup lsp
!
ip vrf customer_vpn1
 rd 65010:1
 route-target both 65010:1
!
interface GigabitEthernet 0/1
 ip vrf forwarding customer_vpn1
 ip address 10.0.2.1 255.255.255.0
 no shutdown
!
router ospf 10 vrf customer_vpn1
 network 10.0.2.0 0.0.0.255 area 0
 redistribute bgp subnets
!
router bgp 65010
 address-family ipv4 vrf customer_vpn1
  redistribute ospf 10
  exit-address-family
!
```

- PE5 configuration file

```
hostname PE5
```

```
!
interface GigabitEthernet 0/2
 ip address 172.1.5.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.3.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 7.7.7.7 255.255.255.255
!
router ospf 1
 network 7.7.7.7 0.0.0.0 area 0
 network 172.1.5.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 6.6.6.6 remote-as 65010
 neighbor 6.6.6.6 update-source Loopback 0
 neighbor 6.6.6.6 next-hop-self
 neighbor 8.8.8.8 remote-as 65010
 neighbor 8.8.8.8 update-source Loopback 0
 neighbor 8.8.8.8 route-reflector-client
!
recursive-route lookup lsp
!
ip vrf customer_vpn1
 rd 65010:1
 route-target both 65010:1
!
interface GigabitEthernet 0/1
 ip vrf forwarding customer_vpn1
 ip address 10.0.3.1 255.255.255.0
 no shutdown
!
router ospf 10 vrf customer_vpn1
 network 10.0.3.0 0.0.0.255 area 0
```

```
 redistribute bgp subnets
!
router bgp 65010
 address-family ipv4 vrf customer_vpn1
  redistribute ospf 10
 exit-address-family
!
```

- PE6 configuration file

```
hostname PE6
!
interface GigabitEthernet 0/2
 ip address 172.1.6.1 255.255.255.0
 no shutdown
!
interface GigabitEthernet 0/1
 ip address 10.0.4.1 255.255.255.0
 no shutdown
!
interface Loopback 0
 ip address 8.8.8.8 255.255.255.255
!
router ospf 1
 network 8.8.8.8 0.0.0.0 area 0
 network 172.1.6.0 0.0.0.255 area 0
!
router bgp 65010
 neighbor 6.6.6.6 remote-as 65010
 neighbor 6.6.6.6 update-source Loopback 0
 neighbor 6.6.6.6 next-hop-self
 neighbor 7.7.7.7 remote-as 65010
 neighbor 7.7.7.7 update-source Loopback 0
 neighbor 7.7.7.7 route-reflector-client
!
recursive-route lookup lsp
!
ip vrf customer_vpn1
 rd 65010:1
 route-target both 65010:1
```

```
!
interface GigabitEthernet 0/1
 ip vrf forwarding customer_vpn1
 ip address 10.0.4.1 255.255.255.0
 no shutdown
!
router ospf 10 vrf customer_vpn1
 network 10.0.4.0 0.0.0.255 area 0
 redistribute bgp subnets
!
router bgp 65010
 address-family ipv4 vrf customer_vpn1
  redistribute ospf 10
  exit-addrss-family
!
```

7. Common Errors

- VPN routes cannot be pinged if the capability of parsing the next hops in BGP routes to LSP tunnels is disabled.

1.17 IPv6 MPLS L3VPN Configuration Examples

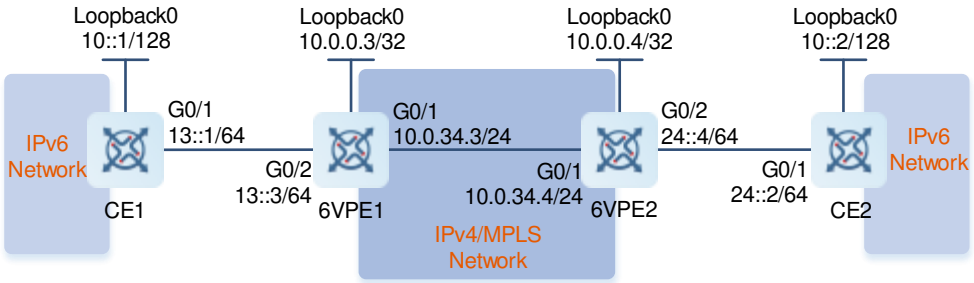
1.17.1 Configuring the 6VPE Service Model

1. Requirements

A company has branches in Fuzhou and Beijing and implements communication between the branches through an ISP. The two branches use IPv6 communication internally, and the ISP's network is an IPv4 MPLS network. The company requires that the branches can communicate with each other, VPN connections are used inside branches, and branches do not communicate with external ISPs.

2. Topology

Figure 1-37 Configuring the 6VPE Service Model



3. Notes

- On 6VPE1 and 6VPE2, configure interface IP addresses and OSPF to ensure that the routes between them are reachable.
- On 6VPE1 and 6VPE2, configure public network tunnels.
- On 6VPE1 and 6VPE2, create VRF instances.
- Configure the IPv6 addresses and routes of 6VPE1 and 6VPE2 under VRF instances.
- Establish a BGP session between 6VPE1 and 6VPE2.
- Configure 6VPE1 and 6VPE2 to redistribute IPv6 routes with CE1 and CE2 under VRF instances.
- On CE1 and CE2, configure the IPv6 addresses and static routes.

4. Procedure

(1) On 6VPE1 and 6VPE2, configure interface IP addresses and OSPF to ensure that the routes between them are reachable.

Configure 6VPE1.

```
6VPE1> enable
6VPE1# configure terminal
6VPE1(config)# interface gigabitethernet 0/1
6VPE1(config-if-GigabitEthernet 0/1)# no switchport
6VPE1(config-if-GigabitEthernet 0/1)# ip address 10.0.34.3 255.255.255.0
6VPE1(config-if-GigabitEthernet 0/1)# exit
6VPE1(config)# interface loopback 0
6VPE1(config-if-Loopback 0)# ip address 10.0.0.3 255.255.255.255
6VPE1(config-if-Loopback 0)# exit
6VPE1(config)# router ospf 1
6VPE1(config-router)# network 10.0.0.3 0.0.0.0 area 0
6VPE1(config-router)# network 10.0.34.0 0.0.0.255 area 0
6VPE1(config-router)# exit
```

Configure 6VPE2.

```
6VPE2> enable
6VPE2# configure terminal
6VPE2(config)# interface gigabitethernet 0/1
6VPE2(config-if-GigabitEthernet 0/1)# no switchport
6VPE2(config-if-GigabitEthernet 0/1)# ip address 10.0.34.4 255.255.255.0
6VPE2(config-if-GigabitEthernet 0/1)# exit
6VPE2(config)# interface loopback 0
6VPE2(config-if-Loopback 0)# ip address 10.0.0.4 255.255.255.255
6VPE2(config-if-Loopback 0)# exit
6VPE2(config)# router ospf 1
6VPE2(config-router)# network 10.0.0.4 0.0.0.0 area 0
6VPE2(config-router)# network 10.0.34.0 0.0.0.255 area 0
6VPE2(config-router)# exit
```

(2) On 6VPE1 and 6VPE2, configure public network tunnels.

Configure 6VPE1.

```
6VPE1(config)# mpls enable
6VPE1(config)# mpls router ldp
6VPE1(config-mpls-router)# ldp router-id interface loopback 0 force
6VPE1(config-mpls-router)# exit
6VPE1(config)# interface gigabitethernet 0/1
6VPE1(config-if-GigabitEthernet 0/1)# label-switching
6VPE1(config-if-GigabitEthernet 0/1)# mpls ldp enable
6VPE1(config-if-GigabitEthernet 0/1)# exit
```

Configure 6VPE2.

```
6VPE2(config)# mpls enable
6VPE2(config)# mpls router ldp
6VPE2(config-mpls-router)# ldp router-id interface loopback 0 force
6VPE2(config-mpls-router)# exit
6VPE2(config)# interface gigabitethernet 0/1
6VPE2(config-if-GigabitEthernet 0/1)# label-switching
6VPE2(config-if-GigabitEthernet 0/1)# mpls ldp enable
6VPE2(config-if-GigabitEthernet 0/1)# exit
```

(3) On 6VPE1 and 6VPE2, create VRF instances.

Configure 6VPE1.

```
6VPE1(config)# vrf definition vrf1
6VPE1(config-vrf)# rd 34:34
6VPE1(config-vrf)# address-family ipv6
6VPE1(config-vrf-af)# route-target export 34:34
6VPE1(config-vrf-af)# route-target import 34:34
6VPE1(config-vrf-af)# exit-address-family
6VPE1(config-vrf)# exit
```

Configure 6VPE2.

```
6VPE2(config)# vrf definition vrf1
6VPE2(config-vrf)# rd 34:34
6VPE2(config-vrf)# address-family ipv6
6VPE2(config-vrf-af)# route-target export 34:34
6VPE2(config-vrf-af)# route-target import 34:34
6VPE2(config-vrf-af)# exit-address-family
6VPE2(config-vrf-af)# exit
```

(4) Configure the IPv6 addresses and routes of 6VPE1 and 6VPE2 under VRF instances.

Configure 6VPE1.

```
6VPE1(config)# interface gigabitethernet 0/2
6VPE1(config-if-GigabitEthernet 0/2)# vrf forwarding vrf1
6VPE1(config-if-GigabitEthernet 0/2)# ipv6 enable
6VPE1(config-if-GigabitEthernet 0/2)# ipv6 address 13::3/64
6VPE1(config-if-GigabitEthernet 0/2)# exit
6VPE1(config)# ipv6 route vrf vrf1 10::1/128 13::1
```

Configure 6VPE2.

```
6VPE2(config)# interface gigabitethernet 0/2
6VPE2(config-if-GigabitEthernet 0/2)# vrf forwarding vrf1
6VPE2(config-if-GigabitEthernet 0/2)# ipv6 enable
6VPE2(config-if-GigabitEthernet 0/2)# ipv6 address 24::4/64
6VPE2(config-if-GigabitEthernet 0/2)# exit
6VPE2(config)# ipv6 route vrf vrf1 10::2/128 24::2
```

(5) Establish a BGP session between 6VPE1 and 6VPE2.

Configure 6VPE1.

```
6VPE1(config)# router bgp 34
6VPE1(config-router)# neighbor 10.0.0.4 remote-as 34
6VPE1(config-router)# neighbor 10.0.0.4 update-source loopback 0
6VPE1(config-router)# address-family vpnv6 unicast
6VPE1(config-router-af)# neighbor 10.0.0.4 activate
6VPE1(config-router-af)# exit-address-family
```

Configure 6VPE2.

```
6VPE2(config)# router bgp 34
6VPE2(config-router)# neighbor 10.0.0.3 remote-as 34
6VPE2(config-router)# neighbor 10.0.0.3 update-source loopback 0
6VPE2(config-router)# address-family vpnv6 unicast
6VPE2(config-router-af)# neighbor 10.0.0.3 activate
6VPE2(config-router-af)# exit-address-family
```

(6) Configure 6VPE1 and 6VPE2 to distribute IPv6 routes under VRF instances.

Configure 6VPE1.

```
6VPE1(config-router)# address-family ipv6 vrf vrf1
6VPE1(config-router-af)# redistribute static
```

Configure 6VPE2.

```
6VPE2(config-router)# address-family ipv6 vrf vrf1
6VPE2(config-router-af)# redistribute static
```

(7) On CE1 and CE2, configure the IPv6 addresses and static routes.

Configure CE1.

```
CE1> enable
CE1# configure terminal
CE1(config)# interface loopback 0
CE1(config-if-Loopback 0)# ipv6 enable
CE1(config-if-Loopback 0)# ipv6 address 10::1/128
CE1(config-if-Loopback 0)# exit
CE1(config)# interface gigabitethernet 0/1
CE1(config-if-GigabitEthernet 0/1)# ipv6 enable
CE1(config-if-GigabitEthernet 0/1)# ipv6 address 13::1/64
CE1(config-if-GigabitEthernet 0/1)# exit
CE1(config)# ipv6 route ::/0 13::3
CE1(config)# exit
```

Configure CE2.

```
CE2> enable
CE2# configure terminal
CE2(config)# interface loopback 0
CE2(config-if-Loopback 0)# ipv6 enable
CE2(config-if-Loopback 0)# ipv6 address 10::2/128
CE2(config-if-Loopback 0)# exit
CE2(config)# interface gigabitethernet 0/1
CE2(config-if-GigabitEthernet 0/1)# ipv6 enable
CE2(config-if-GigabitEthernet 0/1)# ipv6 address 24::2/64
CE2(config-if-GigabitEthernet 0/1)# exit
CE2(config)# ipv6 route ::/0 24::4
CE2(config)# exit
```

5. Verification

On CE1, run the **show ipv6 route** command to display the routing table.

```
CE1# show ipv6 route
IPv6 routing table name - Default - 5 entries
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       1L - IS-IS level-1 LOCATOR, 2L - IS-IS level-2 LOCATOR
       IA - Inter area, EV - BGP EVPN, N - Nd to host, SR - SRv6

S     ::/0 [1/0] via 13::3
              (recursive via 13::3, GigabitEthernet 0/1), 00:20:46
C     10::1/128 is directly connected, Loopback 0, 00:15:46
C     13::/64 is directly connected, GigabitEthernet 0/1, 00:15:46
L     13::1/128 is directly connected, GigabitEthernet 0/1, 00:15:46
C     FE80::/10 via ::1, Null0, 00:15:46
C     FE80::/64 is directly connected, Loopback 0, 00:15:46
L     FE80::274:9CFF:FEEE:53CB/128 is directly connected, Loopback 0, 00:15:46
C     FE80::/64 is directly connected, GigabitEthernet 0/1, 00:15:46
L     FE80::274:9CFF:FEEE:53CB/128 is directly connected, GigabitEthernet 0/1,
00:15:46
```

On CE2, run the **show ipv6 route** command to display the routing table.

```
CE2# show ipv6 route
IPv6 routing table name - Default - 5 entries
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       1L - IS-IS level-1 LOCATOR, 2L - IS-IS level-2 LOCATOR
```

```
       IA - Inter area, EV - BGP EVPN, N - Nd to host, SR - SRv6

S     ::/0 [1/0] via 24::4
              (recursive via 24::4, GigabitEthernet 0/1), 00:20:46
C     10::2/128 is directly connected, GigabitEthernet 0/1, 00:15:46
C     24::/64 is directly connected, GigabitEthernet 0/1, 00:15:46
L     24::2/128 is directly connected, GigabitEthernet 0/1, 00:15:46
C     FE80::/10 via ::1, Null0, 00:15:46
C     FE80::/64 is directly connected, GigabitEthernet 0/1, 00:15:46
L     FE80::274:9CFF:FEC8:E27B/128 is directly connected, GigabitEthernet 0/1,
00:15:46
L     FE80::274:9CFF:FEC8:E27B/128 is directly connected, Loopback 0, 00:15:46
```

On 6VPE1, run the **show ipv6 route vrf vrf1** command to display the private network routing table.

```
6VPE1# show ipv6 route vrf vrf1
IPv6 routing table name - vrf1 - 7 entries
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       1L - IS-IS level-1 LOCATOR, 2L - IS-IS level-2 LOCATOR
       IA - Inter area, EV - BGP EVPN, N - Nd to host, SR - SRv6

S     10::1/128 [1/0] via 13::1
              (recursive via 13::1, GigabitEthernet 0/2), 00:20:46
B     10::2/128 [200/0] via ::FFFF:10.0.0.4, IPv6-mpls, 00:20:46
C     13::/64 is directly connected, GigabitEthernet 0/2, 00:15:46
L     13::3/128 is directly connected, GigabitEthernet 0/2, 00:15:46
C     FE80::/10 via ::1, Null0, 00:20:46
C     FE80::/64 is directly connected, GigabitEthernet 0/2, 00:15:46
L     FE80::274:9CFF:FEEE:F49F/128 is directly connected, GigabitEthernet 0/2,
00:15:46
```

On 6VPE2, run the **show ipv6 route vrf vrf1** command to display the private network routing table.

```
6VPE2# show ipv6 route vrf vrf1
IPv6 routing table name - vrf1 - 7 entries
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       1L - IS-IS level-1 LOCATOR, 2L - IS-IS level-2 LOCATOR
       IA - Inter area, EV - BGP EVPN, N - Nd to host, SR - SRv6

B     10::1/128 [200/0] via ::FFFF:10.0.0.3, IPv6-mpls, 00:20:46
S     10::2/128 [1/0] via 24::2
```

```
              (recursive via 24::2, GigabitEthernet 0/2), 00:20:46
C     24::/64 is directly connected, GigabitEthernet 0/2, 00:15:46
L     24::4/128 is directly connected, GigabitEthernet 0/2, 00:15:46
C     FE80::/10 via ::1, Null0, 00:15:46
C     FE80::/64 is directly connected, GigabitEthernet 0/2, 00:15:46
L     FE80::2D0:F8FF:FEEE:8C1F/128 is directly connected, GigabitEthernet 0/2,
00:15:46
```

On CE1, 10::2 is pingable.

```
CE1# ping 10::2 source 10::1
Sending 5, 100-byte ICMP Echoes to 10::2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/9/10 ms.
```

On CE2, 10::1 is pingable.

```
CE2# ping 10::1 source 10::2
Sending 5, 100-byte ICMP Echoes to 10::1, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/9/12 ms.
```

6. Configuration Files

- # 6VPE1 configuration file

```
hostname 6VPE1
!
vrf definition vrf1
 rd 34:34
 address-family ipv6
  route-target both 34:34
  exit-address-family
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.0.34.3 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 vrf forwarding vrf1
 ipv6 address 13::3/64
 ipv6 enable
!
interface Loopback 0
```

```
 ip address 10.0.0.3 255.255.255.255
!
router bgp 34
 neighbor 10.0.0.4 remote-as 34
 neighbor 10.0.0.4 update-source Loopback 0
 address-family vpnv6 unicast
  neighbor 10.0.0.4 activate
  exit-address-family
!
 address-family ipv6 vrf vrf1
  redistribute static
  exit-address-family
!
router ospf 1
 network 10.0.0.3 0.0.0.0 area 0
 network 10.0.34.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
ipv6 route vrf vrf1 10::1/128 13::1
!
```

- 6VPE2 configuration file

```
hostname 6VPE2
!
vrf definition vrf1
 rd 34:34
 address-family ipv6
  route-target both 34:34
  exit-address-family
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 10.0.34.4 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 vrf forwarding vrf1
 ipv6 address 24::4/64
 ipv6 enable
!
interface Loopback 0
```

```
 ip address 10.0.0.4 255.255.255.255
!
router bgp 34
 neighbor 10.0.0.3 remote-as 34
 neighbor 10.0.0.3 update-source Loopback 0
 address-family vpnv6 unicast
  neighbor 10.0.0.3 activate
  exit-address-family
!
 address-family ipv6 vrf vrf1
  redistribute static
  exit-address-family
!
router ospf 1
 network 10.0.0.4 0.0.0.0 area 0
 network 10.0.34.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
ipv6 route vrf vrf1 10::2/128 24::2
```

- CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
 no switchport
 ipv6 address 13::1/64
 ipv6 enable
!
interface Loopback 0
 ipv6 address 10::1/128
 ipv6 enable
!
ipv6 route ::/0 13::3
!
```

- CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/1
 no switchport
 ipv6 address 24::2/64
 ipv6 enable
!
interface Loopback 0
 ipv6 address 10::2/128
```

```
 ipv6 enable
!
ipv6 route ::/0 24::4
!
```

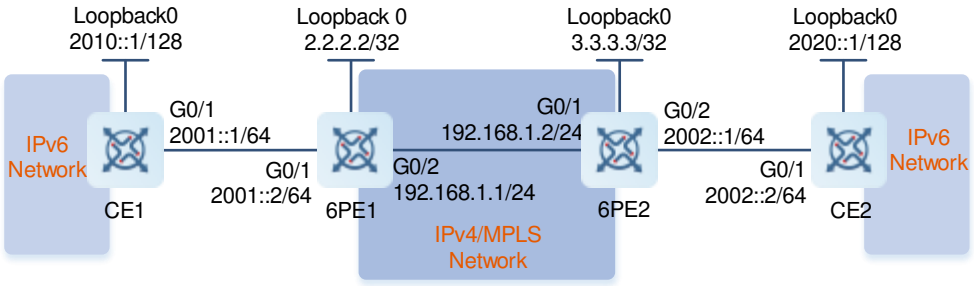
1.17.2 Configuring the 6PE Service Model

1. Requirements

A company has branches in Fuzhou and Beijing and implements communication between the branches through an ISP. The two branches use IPv6 communication internally, and the ISP's network is an IPv4 MPLS network. The company wants to access ISPs in IPv6 method without modifying IPv6 communication between branches.

2. Topology

Figure 1-38 Configuring the 6PE Service Model



3. Notes

- On 6PE1 and 6PE2, configure the interface IP addresses and OSPF to ensure that the routes between them are reachable.
- On 6PE1 and 6PE2, configure public network tunnels.
- On 6PE1 and 6PE2, configure the IPv6 addresses and OSPFv3.
- Establish a BGP session between 6PE1 and 6PE2 and redistribute the IPv6 routes between 6PE1 and CE1 and between 6PE2 and CE2.
- On CE1 and CE2, configure IPv6 addresses and OSPFv3 routes.

4. Procedure

(1) On 6PE1 and 6PE2, configure the interface IP addresses and OSPF to ensure that the routes between them are reachable.

Configure 6PE1.

```
6PE1> enable
6PE1# configure terminal
6PE1(config)# interface loopback 0
6PE1(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
6PE1(config-if-Loopback 0)# exit
6PE1(config)# interface gigabitethernet 0/2
6PE1(config-if-GigabitEthernet 0/2)# no switchport
```

```
6PE1(config-if-GigabitEthernet 0/2)# ip address 192.168.1.1 255.255.255.0
6PE1(config-if-GigabitEthernet 0/2)# exit
6PE1(config)# router ospf 1
6PE1(config-router)# network 192.168.1.0 0.0.0.255 area 0
6PE1(config-router)# network 2.2.2.2 0.0.0.0 area 0
6PE1(config-router)# exit
```

Configure 6PE2.

```
6PE2> enable
6PE2# configure terminal
6PE2(config)# interface loopback 0
6PE2(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
6PE2(config-if-Loopback 0)# exit
6PE2(config)# interface gigabitethernet 0/1
6PE2(config-if-GigabitEthernet 0/1)# no switchport
6PE2(config-if-GigabitEthernet 0/1)# ip address 192.168.1.2 255.255.255.0
6PE2(config-if-GigabitEthernet 0/1)# exit
6PE2(config)# router ospf 1
6PE2(config-router)# network 192.168.1.0 0.0.0.255 area 0
6PE2(config-router)# network 3.3.3.3 0.0.0.0 area 0
6PE2(config-router)# exit
```

(2) On 6PE1 and 6PE2, configure public network tunnels.

Configure 6PE1.

```
6PE1(config)# mpls enable
6PE1(config)# mpls router ldp
6PE1(config-mpls-router)# ldp router-id interface loopback 0 force
6PE1(config-mpls-router)# exit
6PE1(config)# interface gigabitethernet 0/2
6PE1(config-if-GigabitEthernet 0/2)# label-switching
6PE1(config-if-GigabitEthernet 0/2)# mpls ldp enable
6PE1(config-if-GigabitEthernet 0/2)# exit
```

Configure 6PE2.

```
6PE2(config)# mpls enable
6PE2(config)# mpls router ldp
6PE2(config-mpls-router)# ldp router-id interface loopback 0 force
6PE2(config-mpls-router)# exit
6PE2(config)# interface gigabitethernet 0/1
6PE2(config-if-GigabitEthernet 0/1)# label-switching
6PE2(config-if-GigabitEthernet 0/1)# mpls ldp enable
6PE2(config-if-GigabitEthernet 0/1)# exit
```

(3) On 6PE1 and 6PE2, configure IPv6 addresses and OSPFv3 routes.

Configure 6PE1.

```
6PE1(config)# ipv6 router ospf 10
6PE1(config-router)# router-id 2.2.2.2
Change router-id and update OSPFv3 process! [yes/no]:yes
```

```
6PE1(config-router)# exit
6PE1(config)# interface gigabitethernet 0/1
6PE1(config-if-GigabitEthernet 0/1)# no switchport
6PE1(config-if-GigabitEthernet 0/1)# ipv6 enable
6PE1(config-if-GigabitEthernet 0/1)# ipv6 address 2001::2/64
6PE1(config-if-GigabitEthernet 0/1)# ipv6 ospf 10 area 0
6PE1(config-if-GigabitEthernet 0/1)# exit
```

Configure 6PE2.

```
6PE2(config)# ipv6 router ospf 10
6PE2(config-router)# router-id 3.3.3.3
Change router-id and update OSPFv3 process! [yes/no]:yes
6PE2(config-router)# exit
6PE2(config)# interface gigabitethernet 0/2
6PE2(config-if-GigabitEthernet 0/2)# no switchport
6PE2(config-if-GigabitEthernet 0/2)# ipv6 enable
6PE2(config-if-GigabitEthernet 0/2)# ipv6 address 2002::1/64
6PE2(config-if-GigabitEthernet 0/2)# ipv6 ospf 10 area 0
6PE2(config-if-GigabitEthernet 0/2)# exit
```

(4) Establish a BGP session between 6PE1 and 6PE2 and redistribute the IPv6 routes between 6PE1 and CE1 and between 6PE2 and CE2.

Configure 6PE1.

```
6PE1(config)# router bgp 100
6PE1(config-router)# neighbor 3.3.3.3 remote-as 100
6PE1(config-router)# neighbor 3.3.3.3 update-source loopback 0
6PE1(config-router)# address-family ipv6 unicast
6PE1(config-router-af)# neighbor 3.3.3.3 activate
6PE1(config-router-af)# neighbor 3.3.3.3 send-label
6PE1(config-router-af)# redistribute ospf 10
6PE1(config-router-af)# exit
6PE1(config-router)# exit
6PE1(config)# ipv6 router ospf 10
6PE1(config-router)# redistribute bgp
6PE1(config-router)# end
```

Configure 6PE2.

```
6PE2(config)# router bgp 100
6PE2(config-router)# neighbor 2.2.2.2 remote-as 100
6PE2(config-router)# neighbor 2.2.2.2 update-source loopback 0
6PE2(config-router)# address-family ipv6 unicast
6PE2(config-router-af)# neighbor 2.2.2.2 activate
6PE2(config-router-af)# neighbor 2.2.2.2 send-label
6PE2(config-router-af)# redistribute ospf 10
6PE2(config-router-af)# exit
6PE2(config-router)# exit
6PE2(config)# ipv6 router ospf 10
```

```
6PE2(config-router)# redistribute bgp
6PE2(config-router)# end
```

(5) On CE1 and CE2, configure IPv6 addresses and OSPFv3 routes.

Configure CE1.

```
CE1> enable
CE1# configure terminal
CE1(config)# ipv6 router ospf 1
CE1(config-router)# router-id 1.1.1.1
Change router-id and update OSPFv3 process! [yes/no]:yes
CE1(config-router)# exit
CE1(config)# interface loopback 0
CE1(config-if-Loopback 0)# ipv6 enable
CE1(config-if-Loopback 0)# ipv6 address 2010::1/128
CE1(config-if-Loopback 0)# ipv6 ospf 1 area 0
CE1(config-if-Loopback 0)# exit
CE1(config)# interface gigabitethernet 0/1
CE1(config-if-GigabitEthernet 0/1)# no switchport
CE1(config-if-GigabitEthernet 0/1)# ipv6 enable
CE1(config-if-GigabitEthernet 0/1)# ipv6 address 2001::1/64
CE1(config-if-GigabitEthernet 0/1)# ipv6 ospf 1 area 0
CE1(config-if-GigabitEthernet 0/1)# end
```

Configure CE2.

```
CE2> enable
CE2# configure terminal
CE2(config)# ipv6 router ospf 1
CE2(config-router)# router-id 4.4.4.4
Change router-id and update OSPFv3 process! [yes/no]:yes
CE2(config)# interface loopback 0
CE2(config-if-Loopback 0)# ipv6 enable
CE2(config-if-Loopback 0)# ipv6 address 2020::1/128
CE2(config-if-Loopback 0)# ipv6 ospf 1 area 0
CE2(config-if-Loopback 0)# exit
CE2(config)# interface gigabitethernet 0/1
CE2(config-if-GigabitEthernet 0/1)# no switchport
CE2(config-if-GigabitEthernet 0/1)# ipv6 enable
CE2(config-if-GigabitEthernet 0/1)# ipv6 address 2002::2/64
CE2(config-if-GigabitEthernet 0/1)# ipv6 ospf 1 area 0
CE2(config-if-GigabitEthernet 0/1)# end
```

5. Verification

After the configuration is completed, run the **show ipv6 route** command to display IPv6 route entries.

CE1 verification result

```
CE1# show ipv6 route
IPv6 routing table name - Default - 10 entries
```

```
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       1L - IS-IS level-1 LOCATOR, 2L - IS-IS level-2 LOCATOR
       IA - Inter area, EV - BGP EVPN, N - Nd to host, SR - SRv6

C     2001::/64 is directly connected, GigabitEthernet 0/1, 00:00:06
L     2001::1/128 is directly connected, GigabitEthernet 0/1, 00:00:06
O  E2  2002::/64 [110/1] via FE80::250:56FF:FEB0:59C, GigabitEthernet 0/1,
00:20:46
C     2010::1/128 is directly connected, Loopback 0, 00:00:06
O  E2  2020::1/128 [110/1] via FE80::250:56FF:FEB0:59C, GigabitEthernet 0/1,
00:20:46
C     FE80::/10 via ::1, Null0, 00:00:06
C     FE80::/64 is directly connected, Loopback 0, 00:00:06
L     FE80::250:56FF:FEB5:E383/128 is directly connected, Loopback 0, 00:00:06
C     FE80::/64 is directly connected, GigabitEthernet 0/1, 00:00:06
L     FE80::250:56FF:FEB5:E38A/128 is directly connected, GigabitEthernet 0/1,
00:00:06
```

CE2 verification result

```
CE2# show ipv6 route
IPv6 routing table name - Default - 10 entries
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       1L - IS-IS level-1 LOCATOR, 2L - IS-IS level-2 LOCATOR
       IA - Inter area, EV - BGP EVPN, N - Nd to host, SR - SRv6

O  E2  2001::/64 [110/1] via FE80::250:56FF:FEB5:F7B6, GigabitEthernet 0/1,
00:10:05
C     2002::/64 is directly connected, GigabitEthernet 0/1, 00:00:06
L     2002::2/128 is directly connected, GigabitEthernet 0/1, 00:00:06
O  E2  2010::1/128 [110/1] via FE80::250:56FF:FEB5:F7B6, GigabitEthernet 0/1,
00:10:05
C     2020::1/128 is directly connected, Loopback 0, 00:00:06
C     FE80::/10 via ::1, Null0, 00:00:06
C     FE80::/64 is directly connected, GigabitEthernet 0/1, 00:00:06
L     FE80::250:56FF:FEB5:7BF2/128 is directly connected, GigabitEthernet 0/1,
00:00:06
C     FE80::/64 is directly connected, Loopback 0, 00:00:06
L     FE80::250:56FF:FEB5:7BF1/128 is directly connected, Loopback 0, 00:00:06
```

6PE1 verification result

```
6PE1# show ipv6 route
IPv6 routing table name - Default - 8 entries
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       IA - Inter area, EV - BGP EVPN, N - Nd to host

C     2001::/64 via GigabitEthernet 0/1, directly connected
L     2001::2/128 via GigabitEthernet 0/1, local host
B     2002::/64 [200/1] via ::FFFF:3.3.3.3, IPv6-mpls
O     2010::1/128 [110/1] via FE80::250:56FF:FEB5:E38A, GigabitEthernet 0/1
B     2020::1/128 [200/1] via ::FFFF:3.3.3.3, IPv6-mpls
C     FE80::/10 via ::1, Null0
C     FE80::/64 via GigabitEthernet 0/1, directly connected
L     FE80::250:56FF:FEB0:59C/128 via GigabitEthernet 0/1, local host
```

6PE2 verification result

```
6PE2# show ipv6 route
IPv6 routing table name - Default - 8 entries
Codes:  C - Connected, L - Local, S - Static
       R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       1L - IS-IS level-1 LOCATOR, 2L - IS-IS level-2 LOCATOR
       IA - Inter area, EV - BGP EVPN, N - Nd to host, SR - SRv6

B     2001::/64 [200/1] via ::FFFF:2.2.2.2, IPv6-mpls, 00:20:46
C     2002::/64 is directly connected, GigabitEthernet 0/2, 00:00:06
L     2002::1/128 is directly connected, GigabitEthernet 0/2, 00:00:06
B     2010::1/128 [200/1] via ::FFFF:2.2.2.2, IPv6-mpls, 00:20:46
O     2020::1/128 [110/1] via FE80::250:56FF:FEB5:7BF2, GigabitEthernet 0/2,
00:20:46
C     FE80::/10 via ::1, Null0, 00:00:06
C     FE80::/64 is directly connected, GigabitEthernet 0/2, 00:00:06
L     FE80::250:56FF:FEB5:F7B6/128 is directly connected, GigabitEthernet 0/4,
00:00:06
```

On CE1, 2020::1 is pingable.

```
CE1# ping 2020::1 source 2010::1
Sending 5, 100-byte ICMP Echoes to 2020::1, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms.
```

On CE2, 10::1 is pingable.

```
CE2# ping 2010::1 source 2020::1
Sending 5, 100-byte ICMP Echoes to 2010::1, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/11 ms.
```

6. Configuration Files

- 6PE1 configuration file

```
hostname 6PE1
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ipv6 address 2001::2/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
interface GigabitEthernet 0/2
 no switchport
 ip address 192.168.1.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 2.2.2.2 255.255.255.255
!
router bgp 100
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback 0
 address-family ipv6
  redistribute ospf 10
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-label
  exit-address-family
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
ipv6 router ospf 10
 router-id 2.2.2.2
 redistribute bgp
!
mpls router ldp
 ldp router-id interface Loopback 0 force
```

```
!
```

- 6PE2 configuration file

```
hostname 6PE2
!
mpls enable
!
interface GigabitEthernet 0/1
 no switchport
 ip address 192.168.1.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/2
 no switchport
 ipv6 address 2002::1/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
router bgp 100
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback 0
 address-family ipv6
  redistribute ospf 10
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-label
  exit-address-family
!
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
ipv6 router ospf 10
 router-id 3.3.3.3
 redistribute bgp
!
mpls router ldp
 ldp router-id interface Loopback 0 force
!
```

- CE1 configuration file

```
hostname CE1
!
interface GigabitEthernet 0/1
```

```
 no switchport
 ipv6 address 2001::1/64
 ipv6 enable
 ipv6 ospf 1 area 0
!
interface Loopback 0
 ipv6 address 2010::1/128
 ipv6 enable
 ipv6 ospf 1 area 0
!
ipv6 router ospf 1
 router-id 1.1.1.1
!
```

- CE2 configuration file

```
hostname CE2
!
interface GigabitEthernet 0/1
 no switchport
 ipv6 address 2002::2/64
 ipv6 enable
 ipv6 ospf 1 area 0
!
interface Loopback 0
 ipv6 address 2020::1/128
 ipv6 enable
 ipv6 ospf 1 area 0
!
ipv6 router ospf 1
 router-id 4.4.4.4
!
```

Contents

1 Configuring EVPN

1.1 Introduction

1.1.1 Overview

The Multiprotocol Label Switching (MPLS) based L2 Virtual Private Network (L2VPN) using pseudowires (PWs) has been widely deployed in carrier networks and enterprise networks. L2VPN can be applied to various scenarios such as Ethernet services, Fixed Mobile Convergence (FMC), and enterprise campus networks.

With the rapid development of cloud computing and data centers in recent years, Data Center Interconnectivity (DCI) has become a new application scenario for L2VPN. Data centers integrate resources including servers, networks, and storage through virtualization, demanding higher flexibility, reduced costs, and optimized resource utilization across various data centers. Services such as virtual machine migration and cluster storage necessitate that nodes and servers are on the same Layer 2 network.

However, some problems of VPLS have been unveiled in commercial use, including:

- In scenarios involving multi-homing, VPLS currently supports only the single-active redundancy mode. This means that only one active node in the redundancy group and one active link are responsible for forwarding, while backup nodes remain inactive. Therefore, VPLS cannot effectively harness the forwarding capabilities of all Provider Edge (PE) devices and lacks support for load balancing.
- As a PW connection has to be established between any two PE devices on the provider backbone network, a substantial number of PWs are essential when there are numerous sites, resulting in a waste of network resources.
- The sites learn remote MAC addresses through ARP broadcast flooding that generates numerous ARP packets and consumes excess bandwidth.
- Virtualization significantly increases the number of MAC addresses to be managed on the network. The convergence performance following a network failure relies on the capacity of the MAC address table on the PE devices.

To address these challenges, the Ethernet VPN (EVPN) technology has emerged. Different from VPLS, EVPN uses a Multiprotocol-Border Gateway Protocol (MP-BGP) control plane for MAC address learning, making it more controllable and flexible. EVPN supports the all-active redundancy modes and per-flow load balancing, with simplified configuration process. In the case of failures, the network can be recovered swiftly as the reconvergence performance of EVPN is not reliant on the capacity of the MAC address table. Drawing inspiration from the well-established L3VPN technology, EVPN boasts scalability comparable to L3VPN, no longer constrained by device PW capacity. Moreover, EVPN provides carriers with the same O&M experience as L3VPN, leading to a reduction in maintenance costs.

1.1.2 Basic Concepts

- EVI

  EVPN Instance (EVI) is an EVPN instance. A single device can accommodate multiple distinct EVIs, and each EVI can connect to one or more groups of user networks.

- ES

  In EVPN, the Ethernet Segment (ES) represents the link between a PE and a Customer Edge (CE) device.

- ESI

  The Ethernet Segment Identifier (ESI) is the identifier for ES. Each ES has a unique ESI. The ESs connected with the same CE share the same ESI.

- ET

  Each EVI contains one or more Layer 2 networks. When an EVI contains multiple Layer 2 networks, they can be distinguished through Ethernet tags (ETs).

- BD

  Bridge Domain (BD) is an instance of broadcast domain on bridge nodes.

- EVPL

  Ethernet Virtual Private LINE (EVPL) is a P2P L2VPN service.

1.1.3 Topology for EVPN

Figure 1-1 Topology for EVPN

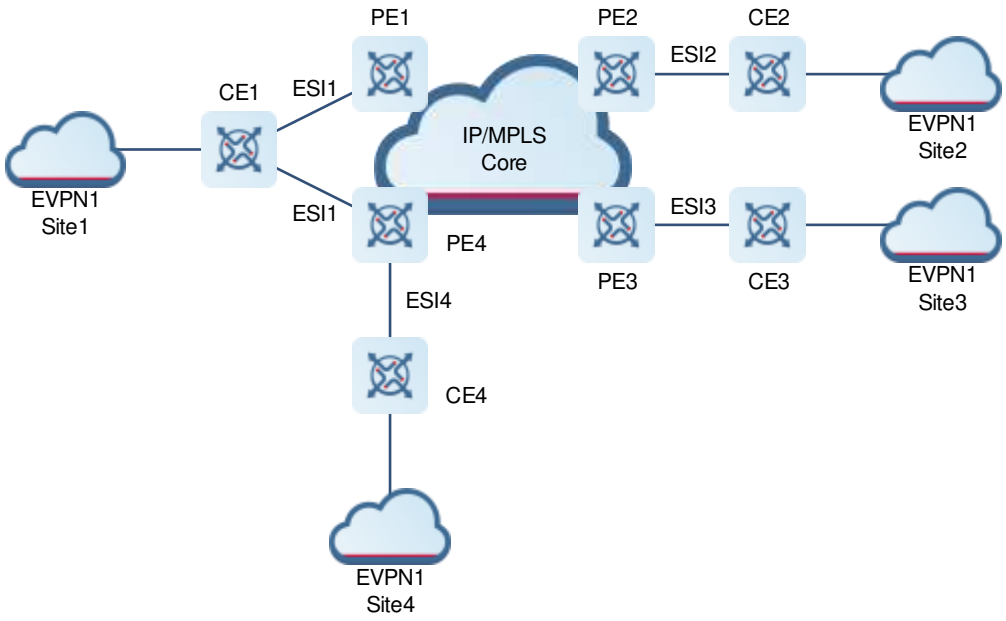


Figure 1-1 shows the topology for EVPN. Similar to MPLS-based L3VPN, an EVPN instance contains a group of CE devices connected with a PE device, which may be the PC, router, or switch. PE provides the virtual Layer 2 bridging for CE devices, enabling Layer 2 communication between sites within the EVPN instance. This requires the PE to configure EVPN instances and establish connections with CE devices. Moreover, MP-BGP connections must be set up between PE devices to exchange EVPN routes.

As is shown in Figure 1-1, CE 1, CE 2, CE 3, and CE 4 are a part of the same EVPN instance (EVPN 1) and connected to PE 1, PE 2, PE 3, and PE 4 respectively. Among these, CE 1 is dual-homed, connected to both PE 1 and PE 4 through redundant links, while CE 2, CE 3, and CE 4 are single-homed, each connecting to a single PE device. On the EVPN, the links between PE and CE devices are referred to as Ethernet Segments (ESs), and each ES has a unique ESI. Links connected to the same site share the same ESI. As shown in Figure

1-1, both PE 1 and PE 4 are connected to CE 1, and the ESIs for their ESs are identical. However, links connected to different sites have distinct ESIs. For instance, CE 1 and CE 4 are connected with different sites, resulting in different ESIs for the ESs between PE 4 and CE 4, and PE 4 and CE 1. EVPN MAC routes shared between PEs carry ESI information, allowing PEs to determine the Ethernet segment where a specific MAC address is.

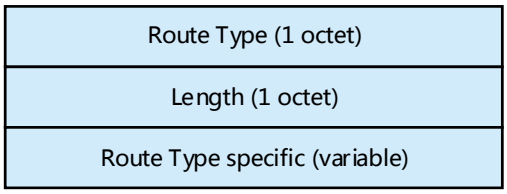
On the EVPN, the public network can be an IP/MPLS network, where MPLS label switched path (LSP) functions as the network tunnel and provides features like Fast ReRoute (FRR) for enhanced availability.

1.1.4 EVPN Routing

1. EVPN NLRI

In contrast to L2VPN, EVPN shifts Layer 2 MAC address learning from the data plane to the MP-BGP control plane. MAC addresses can be published and learned through EVPN routes. EVPN introduces a new concept, namely Network Layer Reachability Information (NLRI), known as EVPN NLRI. The format of EVPN NLRI is as follows:

Figure 1-2 Format of EVPN NLRI



In the preceding table:

- The **Route Type** field determines the format of the other parts of EVPN NLRI routes.
- The **Length** field determines the length of the **Route Type specific** field in bytes.

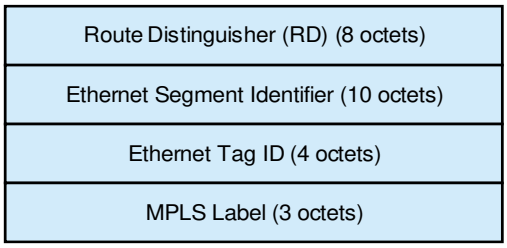
EVPN NLRI is advertised through MP-BGP. The Address Family Identifier (AFI) and Sub Address Family Identifier (SAFI) are 25 on L2VPN and 70 on EVPN respectively. EVPN NLRI is encapsulated in the **MP_REACH/MP_UNREACH** attribute. Before advertising EVPN routes, BGP speakers must conduct capabilities negotiations to determine whether to advertise the routes.

2. Message Structure of EVPN Routes

EVPN realizes functions through different route types.

- Ethernet Auto-Discovery Route

Figure 1-3 Ethernet Auto-Discovery Route



The Ethernet Auto-discovery route automatically discovers Ethernet segment (ES) information on a multi-homed network. It is also referred to as route type 1.

- MAC/IP Advertisement Route

Figure 1-4 MAC/IP Advertisement Route

RD (8 octets)
Ethernet Segment Identifier (10 octets)
Ethernet Tag ID (4 octets)
MAC Address Length (1 octet)
MAC Address (6 octets)
IP Address Length (1 octet)
IP Address (0 or 4 or 16 octets)
MPLS Label1 (3 octets)
MPLS Label2 (0 or 3 octets)

The MAC/IP Advertisement route advertises MAC addresses, ARP information, and IP information. The **IP Address Length** field is in bits. It is also referred to as route type 2.

- Inclusive Multicast Ethernet Tag Route

Figure 1-5 Inclusive Multicast Ethernet Tag Route

RD (8 octets)
Ethernet Tag ID (4 octets)
IP Address Length (1 octet)
Originating Router's IP Address (4 or 16 octets)

The Inclusive Multicast Ethernet Tag route is responsible for establishing tunnels between PE devices. It is also referred to as route type 3.

- Ethernet Segment Route

Figure 1-6 Ethernet Segment Route

RD (8 octets)
Ethernet Tag ID (4 octets)
IP Address Length (1 octet)
Originating Router's IP Address (4 or 16 octets)

The Ethernet Segment route advertises ES information. If multiple PE devices are connected with the same CE, they can discover each other through Ethernet segment routes. It is also referred to as route type 4.

- IP Prefix Route

Figure 1-7 IP Prefix Route

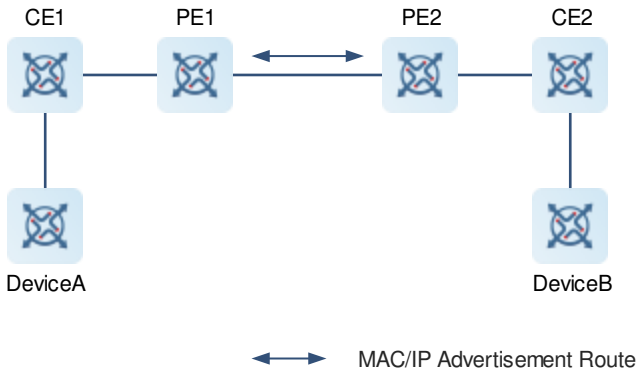
RD (8 octets)
Ethernet Segment Identifier (10 octets)
Ethernet Tag ID (4 octets)
IP Prefix Length (1 octet)
IP Prefix (4 or 16 octets)
GW IP Address (4 or 16 octets)
MPLS Label (3 octets)

The IP Prefix route advertises the IP addresses of a host or IP segment routing information. It is also referred to as route type 5.

1.1.5 Process of EVPN Packet Forwarding

The MAC/IP Advertisement route is taken as an example to describe the process of EVPN packet forwarding.

Figure 1-8 Process of EVPN Packet Forwarding



- Device A advertises its MAC address and IP address to Device B through an ARP request packet or gratuitous ARP packet. When this packet passes through PE 1, PE 1 generates a MAC/IP Advertisement

route for Device A.

- Similarly, PE 2 generates a MAC/IP Advertisement route for Device B.
- PE 1 sends a MAC/IP Advertisement route packet to PE 2, which carries the MAC address, next-hop route, and Route Target (RT) value of EVI.
- Upon reception of the MAC/IP Advertisement route packet, PE 2 determines the corresponding EVI based on the RT value and generates a local forwarding entry for Device A according to the information in the MAC/IP Advertisement route packet.

Similarly, PE 1 utilizes information from the MAC/IP Advertisement route packet sent by PE 2 to generate a local forwarding entry for Device B.

Consequently, PE 1 and PE 2 obtain the MAC addresses of devices in each other's sites. Once the public network tunnel between PE 1 and PE 2 is established, unicast packets between Device A and Device B can be routed through the tunnel, ultimately reaching the respective CE devices through the PE at the end of the tunnel. During the process, forwarding between devices and neighbor CE devices, as well as between CE devices and neighbor PE devices, operates at Layer 2.

1.1.6 Protocols and Standards

- RFC 7432: BGP MPLS-based EVPN
- RFC 8214: Virtual Private Wire Service Support in EVPN
- RFC 8365: A Network Virtualization Overlay Solution Using EVPN
- RFC 9136: IP Prefix Advertisement in EVPN
- RFC 9135: Integrated Routing and Bridging in EVPN

1.2 Configuration Task Summary

EVPN configuration includes the following tasks: Choose and complete the tasks according to the actual situation.

- Configuring Basic Features
  - a Configuring BGP EVPN Peers
  - b (Optional) Disabling Route Target Filtering
  - c (Optional) Enabling Attribute Modification on the Route Reflector
- Configuring EVPN L3VPN
  - a Configuring a VRF Instance

    Configure VRF and EVPN attributes.
  - b Binding an Interface to a VRF Instance
  - c Configuring Route Exchange between CEs
  - d Configuring BGP EVPN Peers
  - e Enabling IP Prefix Route Advertisement in EVPN
  - f (Optional) Enabling the Import of Enhanced VPN Routes
  - g (Optional) Configuring the Upper Limit of MAC Routing Prefixes Received From BGP Peers

1.3 Configuring Basic Features

1.3.1 Restrictions and Guidelines

Before configuring EVPN basic features, complete the following task:

- Configure a basic network for establishing BGP EVPN peers.

1.3.2 Configuration Tasks

The EVPN basic configuration includes the following tasks:

(1) Configuring BGP EVPN Peers

(2) (Optional) Disabling Route Target Filtering

(3) (Optional) Enabling Attribute Modification on the Route Reflector

1.3.3 Configuring BGP EVPN Peers

1. Overview

Configure BGP EVPN Peers and activate the L2VPN EVPN address family.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP, configure the local Autonomous System (AS) number, and enter the BGP routing configuration mode.

router bgp *as-number*

(4) Configure a BGP peer (group).

neighbor { *neighbor-address* | *peer-group-name* } **remote-as** { *as-number* | **route-map** *map-tag* }

(5) (Optional) Add a BGP peer into a BGP peer group.

neighbor *neighbor-address* **peer-group** *peer-group-name*

(6) Configure a network interface for establishing a BGP connection between internal BGP (iBGP) peers.

neighbor { *neighbor-address* | *peer-group-name* } **update-source** { *interface-type interface-number* | *address* }

(7) (Optional) Enable accumulated interior gateway protocol (AIGP) of BGP neighbors.

neighbor { *neighbor-ipv4-address* | *neighbor-ipv6-address* | *peer-group-name* } **aigp** [**send med**]

The AIGP of BGP neighbors is disabled by default.

(8) Enable BGP neighbors to exchange L2VPN EVPN information and enter the BGP L2VPN EVPN address family configuration mode.

address-family l2vpn evpn

BGP neighbors are not allowed to exchange L2VPN EVPN information by default.

(9) Activate neighbors or peer groups in the current address mode.

neighbor { *neighbor-address* **activate** | *peer-group-name* **activate** [**ipv4** | **ipv6**] }

(10) Advertise the community attribute to the specified BGP neighbor.

neighbor { *neighbor-ipv4-address* | *neighbor-ipv6-address* | *peer-group-name* } **send-community** [**both** | **standard** | **extended**]

No community attribute is advertised to the specified BGP neighbor by default.

(11) (Optional) Set the local BGP speaker as the next hop when routes are advertised to the specified BGP peers.

neighbor { *neighbor-ipv4-address* | *neighbor-ipv6-address* | *peer-group-name* } **next-hop-self**

By default, the next hop of routes advertised to an external BGP (eBGP) peer switches to the local GBP speaker, and the next hop of routes advertised to an iBGP peer stays unchanged.

(12) (Optional) Configure a device not to change the next hop of routes advertised to a peer (group).

neighbor { *neighbor-ipv4-address* | *neighbor-ipv6-address* | *peer-group-name* } **next-hop-unchanged**

By default, the next hop of routes advertised to an eBGP peer switches to the local BGP speaker, and the next hop of routes advertised to an iBGP peer stays unchanged.

(13) (Optional) Apply the route map to a received or advertised route.

neighbor { *neighbor-ipv4-address* | *neighbor-ipv6-address* | *peer-group-name* } **route-map** *map-tag* { **in** | **out** }

The route map is not applied to a received or advertised route by default.

(14) (Optional) Set the device as the route reflector and specify its client.

neighbor { *neighbor-ipv4-address* | *neighbor-ipv6-address* | *peer-group-name* } **route-reflector-client**

The local device is not set as the route reflector and no client is specified by default.

(15) (Optional) Configure the encapsulation type for the EVPN routes advertised for the EVPN peer.

neighbor { *neighbor-ipv4-address* | *neighbor-ipv6-address* | *peer-group-name* } **advertise encap-type mpls**

By default, EVPN routes with MPLS encapsulation are advertised for the EVPN peer.

1.3.4 Disabling Route Target Filtering

1. Overview

Restore all received EVPN routes by disabling route target filtering.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP, configure the local AS number, and enter the BGP routing configuration mode.

router bgp *as-number*

(4) Enable BGP neighbors to exchange L2VPN EVPN information and enter the BGP L2VPN EVPN address family configuration mode.

address-family l2vpn evpn

BGP neighbors are not allowed to exchange L2VPN EVPN information by default.

(5) Disable route target filtering.

no bgp default route-target filter

Route target filtering is enabled by default.

1.3.5 Enabling Attribute Modification on the Route Reflector

1. Overview

Normally, the attributes of routes reflected by a route reflector are not modified. However, the route attributes have to be modified in some scenarios. In this case, you can enable attribute modification on the route reflector.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP, configure the local AS number, and enter the BGP routing configuration mode.

router bgp *as-number*

(4) Enable BGP neighbors to exchange L2VPN EVPN information and enter the BGP L2VPN EVPN address family configuration mode.

address-family l2vpn evpn

BGP neighbors are not allowed to exchange L2VPN EVPN information by default.

(5) Enable the route reflector to modify route attributes.

bgp route-reflector attribute-change

The route reflector cannot modify route attributes by default.

1.4 Configuring EVPN L3VPN

1.4.1 Restrictions and Guidelines

Before configuring EVPN L3VPN, complete the following tasks:

- You have configured a basic network for establishing BGP EVPN peers.

1.4.2 Configuration Tasks

(1) Configuring a VRF Instance

Configure VRF and EVPN attributes.

(2) Binding an Interface to a VRF Instance

(3) Configuring Route Exchange between CEs

(4) Configuring BGP EVPN Peers

(5) Enabling IP Prefix Route Advertisement in EVPN

(6) (Optional) Enabling the Import of Enhanced VPN Routes

(7) (Optional) Configuring the Upper Limit of MAC Routing Prefixes Received From BGP Peers

1.4.3 Configuring a VRF Instance

1. Overview

Configure a VPN Routing and Forwarding (VRF) instance, RD, EVPN RT, and import and export policies.

2. Restrictions and Guidelines

- The **route-target evpn** command can be configured in the multi-protocol VRF, multi-protocol VRF IPv4 address family, and multi-protocol VRF IPv6 address family configuration modes. The priority levels of the multi-protocol VRF IPv4 address family and multi-protocol VRF IPv6 address family configuration modes are higher than that of the multi-protocol VRF configuration mode.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a VRF instance. Enter the single- or multi-protocol VRF configuration mode and configure the RD and RT.

- Run the following commands in sequence to configure the single-protocol VRF RD and RT.

  Enter the single-protocol VRF configuration mode.

  ip vrf *vrf-name*

  Configure the RD for the VRF instance.

  rd *rd-value*

  Configure the EVPN RT for the VRF instance.

  route-target { **both** | **export** | **import** } *rt-value* **evpn**

  The RD and RT of the single-protocol VRF are not configured by default.

- Run the following commands in sequence to configure the multi-protocol VRF RD and RT.

  Enter the multi-protocol VRF configuration mode.

  vrf definition *vrf-name*

  Configure the RD for the VRF instance.

  rd *rd-value*

  Enable the IPv4 or IPv6 protocol in the multi-protocol VRF mode and enter the IPv4 or IPv6 address family configuration mode.

  address-family { **ipv4** | **ipv6** }

  Configure the EVPN RT for the VRF instance.

  route-target { **both** | **export** | **import** } *rt-value* **evpn**

(4) Configure EVPN to generate and advertise the IP prefix route for the VRF instance. This configuration is required only for VPN L3VPN over MPLS.

evpn mpls routing enable

EVPN does not generate or advertise the IP prefix route for the VRF instance by default.

(5) (Optional) Configure the policy rules for importing the remote EVPN routes to the local VRF instance.

import map evpn *routemap-name*

No routing policy rule for importing the remote EVPN routes to the local VRF instance is configured by default.

(6) (Optional) Configure routing policy rules for the EVPN routes advertised by the local VRF IPv4 or IPv6 address family to the remote end.

export map evpn *routemap-name*

No routing policy rule for the EVPN routes advertised by the local VRF IPv4 or IPv6 address family to the remote end is configured by default.

1.4.4 Binding an Interface to a VRF Instance

1. Overview

Bind an interface with a VRF instance to make it a private interface. Packets passing through this interface are forwarded using the VRF instance.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

  interface *ethernet-type interface-number*

- Enter the Layer 3 aggregate interface configuration mode.

  interface aggregateport *interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

  interface *ethernet-type interface-number.subnumber*

- Enter the Layer 3 aggregate sub-interface configuration mode.

  interface aggregateport *interface-number.subnumber*

- Enter the SVI configuration mode.

  interface vlan *interface-number*

- Enter the tunnel interface configuration mode.

  interface tunnel *interface-number*

- Enter the loopback interface configuration mode.

  interface loopback *interface-number*

- Enter the virtual PPP interface configuration mode.

  interface virtual-ppp *interface-number*

(4) Perform one of the following configuration tasks.

- Bind the interface with a single-protocol VRF instance.

  ip vrf forwarding *vrf-name*

- Bind the interface with a multi-protocol VRF instance.

  vrf forwarding *vrf-name*

1.4.5 Configuring Route Exchange between CEs

1. Overview

Configure route exchange between CEs.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP, configure the local AS number, and enter the BGP routing configuration mode.

router bgp *as-number*

(4) Enable IPv4 or IPv6 routing information exchange on a VRF instance.

address-family { **ipv4** | **ipv6** } **vrf** *vrf-name*

(5) Configure route redistribution. Perform only one of the following configuration tasks as they are mutually exclusive.

- Run the following commands in sequence to enable route exchange between CEs by configuring BGP.

  neighbor { *neighbor-address* | *peer-group-name* } **remote-as** { *as-number* | **route-map** *map-tag* }

  neighbor { *neighbor-address* **activate** | *peer-group-name* **activate** [**ipv4** | **ipv6**] }

- Redistribute the routes between BGP and other routing protocols.

  redistribute *protocol-type* [**route-map** *map-tag*] [**metric** *metric-value*]

- Redistribute the routes between the Open Shortest Path First (OSPF) protocol and BGP.

  redistribute ospf *process-id* [**route-map** *map-tag*] [**metric** *metric-value*] [**match** | **internal** | **external** [**1** | **2**] | **nssa-external** [**1** | **2**]]

- Redistribute the routes of the IS-IS protocol to BGP.

  redistribute isis [i*sis-tag*] [**route-map** *map-tag*] [**metric** *metric-value*] [**level-1** | **level-1-2** | **level-2**]

- Configure the information to be advertised by the local BGP speaker.

  network { *network-number* [**mask** *mask*] | *prefix* } [**route-map** *map-tag*] [**backdoor**]

1.4.6 Configuring BGP EVPN Peers

See Configuring BGP EVPN Peers for details.

1.4.7 Enabling IP Prefix Route Advertisement in EVPN

1. Overview

You can enable IP prefix route advertisement.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP, configure the local AS number, and enter the BGP routing configuration mode.

router bgp *as-number*

(4) Enable BGP neighbors to exchange L2VPN EVPN information and enter the BGP L2VPN EVPN address family configuration mode.

address-family l2vpn evpn

BGP neighbors are not allowed to exchange L2VPN EVPN information by default.

(5) Redistribute routes.

IPv4:

advertise ipv4 unicast

IPv6:

advertise ipv6 unicast

1.4.8 Enabling the Import of Enhanced VPN Routes

1. Overview

Only the preferred next-hop routes are imported between VRF routing tables or from the remote L3VPN to the VRF routing tables by default. Enable the import of enhanced VPN routes to import all next-hop or equal-cost next-hop routes. The imported routes realize equal cost multipath (ECMP).

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP and enter the BGP routing configuration mode.

router bgp *as-number*

(4) (Optional) Enter one of the following address family configuration modes to enable the import of enhanced VPN routes for a certain address family.

- Enter the BGP IPv4 VRF address family configuration mode.

  address-family ipv4 vrf *vrf-name*

- Enter the BGP IPv6 VRF address family configuration mode.

  address-family ipv6 vrf *vrf-name*

- Enter the IPv4 address family configuration mode of the BGP scope.

scope vrf *vrf-name*

address-family ipv4 [**unicast**]

- Enter the IPv6 address family configuration mode of the BGP scope.

  scope vrf *vrf-name*

  address-family ipv6 [**unicast**]

(5) Configure an import policy.

import path evpn selection all

Only preferred routes are imported by default.

1.4.9 Configuring the Upper Limit of MAC Routing Prefixes Received From BGP Peers

1. Overview

You are advised to configure the upper limit of MAC routing prefixes from BGP peers if the EVPN instance receives excessive irrelevant MAC routes. When the number of MAC routing prefixes exceeds the upper limit, the BGP connection is disabled by default.

2. Restrictions and Guidelines

- If the following configurations are required instead of the BGP disconnection when the number of MAC routing prefixes exceeds the upper limit, run the following commands:
  - Run the **warning-only** command to report an alarm.
  - Run the **suppress** command to stop learning route entries. When the number of MAC routing prefixes exceeds the upper limit after the **suppress** command is executed, the entries learned may be different as the route learning sequence may change before and after the re-establishment of neighbor relationships.
- If you configure the upper limit for a BGP peer group, this configuration is applied to all members in the group. If you configure the upper limit for a member in the group, this configuration overwrites the BGP peer group configuration.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable BGP, configure the local AS number, and enter the BGP routing configuration mode.

router bgp *as-number*

(4) Enable BGP neighbors to exchange L2VPN EVPN information and enter the BGP L2VPN EVPN address family configuration mode.

address-family l2vpn evpn

BGP neighbors are not allowed to exchange L2VPN EVPN information by default.

(5) Configuring the upper limit of MAC routing prefixes received from BGP peers.

neighbor { *neighbor-ipv4-address* | *neighbor-ipv6-address* | *peer-group-name* } **mac-limit** *mac-limit-value* [*mac-limit-threshold*] [**restart-time** *restart-time* | **warning-only** [**suppress**]]

No upper limit of MAC routing prefixes received from BGP peers is configured by default.

1.5 Monitoring

Run the **show** command to verify the configuration result.

Run the **debug** command to output debugging information.

Caution

The output debugging information occupies system resources. Therefore, disable the debugging immediately after use.

Run the **clear** command to clear information.

Caution

Vital information may be lost if you run the **clear** command during device operation, which may cause service interruption.

Table 1-1 Monitoring

Command	Purpose
clear bgp l2vpn evpn { ***** | *as-number* | *neighbor-address* } [**soft**] [**in** | **out**]	Clears the BGP EVPN address family.
clear bgp l2vpn evpn external [**soft**] [**in** | **out**]	Clears all eBGP connections of the BGP EVPN address family.
clear bgp l2vpn evpn peer-group *peer-group-name* [**soft**] [**in** | **out**]	Clears sessions of all members in a peer group.
clear bgp l2vpn evpn update-group [*neighbor-ipv4-address* | *neighbor-ipv6-address* | *neighbor-ipv6-link-local-address* | *update-group-index*] [**soft**] [**in** | **out**]	Clears sessions of all members in the update group within the L2VPN EVPN address family.
clear bgp l2vpn evpn dampening	Clears the flapping information and removes route dampening.
clear bgp l2vpn evpn flap-statistics	Clears route flapping statistics of the BGP EVPN address family.
show bgp evpn [**evi-hash** | **status**]	Displays EVPN information.
show bgp l2vpn evpn all	Displays all routing information of BGP L2VPN.

Command	Purpose
show bgp l2vpn evpn all [{ **ethernet-ad** [*etag-id*] | **ethernet-segment** | **ip-prefix** [*ipv4-address* | *ipv6-address*] } [**from-neighbor** { *neighbor-ipv4-address* | *neighbor-ipv6-address* }] [**detail**]]	Displays the routing information of the specified route type within the BGP L2VPN EVPN address family.
show bgp l2vpn evpn all ip-prefix { *ip_addr* [**from-neighbor** *peer-address*] [**detail**] | *ipv6_addr* [**from-neighbor** *peer-address*] [**detail**] | [**from-neighbor** *peer-address*] **detail** }	Displays the routing information of the five route types within the BGP L2VPN EVPN address family.
show bgp l2vpn evpn all neighbor [{ *neighbor-ipv4-address* | *neighbor-ipv4-address/mask* | *neighbor-ipv6-address* | *neighbor-ipv6-address/prefix-length* } [**advertised-routes** [**check** | **detail**] | **policy** [**detail**] | **received-routes** | **routes**]]	Displays the neighbor information of the BGP L2VPN address family.
show bgp l2vpn evpn all summary	Displays the neighbor summary of the BGP L2VPN address family.
show bgp l2vpn evpn rd *vpn_rd* [{ **ethernet-ad** [*etag-id*] | **ethernet-segment** | **ip-prefix** [*ipv4-address* | *ipv6-address*] } [**detail**]]	Displays the L2VPN EVPN information of the specified RD.
show bgp l2vpn evpn all update-group [*neighbor-address* | *update-group-index*] [**summary**]	Displays the update groups within the BGP L2VPN address family.
show evpn [**name** *evi-name*] [**detail**]	Displays EVI instances.
debug ip bgp evpn	Enables BGP EVPN debugging.

1.6 Configuration Examples

1.6.1 Configuring EVPN L3VPN Over MPLS

1. Requirements

Site 1 and Site 2 are sites for the same L3VPN service operating over EVPN MPLS.

2. Topology

Figure 1-9 EVPN L3VPN Over MPLS

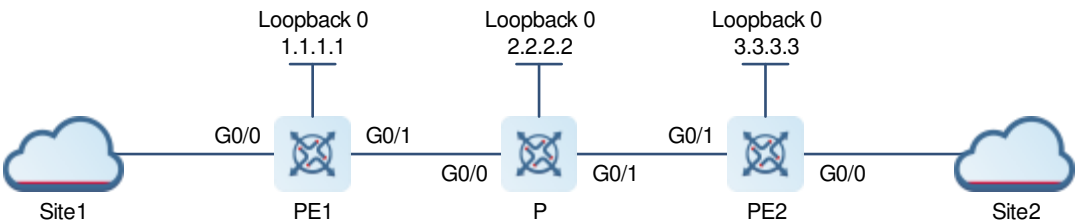


Table 1-2 Description

Device	Port	IP Address
PE 1	GigabitEthernet 0/0	172.18.1.1/24
	GigabitEthernet 0/1	10.1.1.1/24
	Loopback 0	1.1.1.1/32
P	GigabitEthernet 0/0	10.1.1.2/24
	GigabitEthernet 0/1	20.1.1.1/24
	Loopback 0	2.2.2.2/32
PE 2	GigabitEthernet 0/0	172.18.2.1/24
	GigabitEthernet 0/1	20.1.1.2/24
	Loopback 0	3.3.3.3/3

3. Note

- Configure OSPF to realize interconnection between the PEs on the backbone network.
- Configure MPLS basic functions and MPLS Label Distribution Protocol (LDP), and establish LDP Label Switched Path (LSP) on the backbone network.
- Configure a single-protocol VRF instance on the PE and bind it with interfaces between VPN sites.
- Establish interactive EVPN routes between BGP EVPN peers on PEs.
- Configure the advertisement of IP routes as EVPN prefix routes on the PEs.

4. Procedure

(1) Configure Interior Gateway Protocol (IGP) on the backbone network to realize interconnection between the PE devices and device P. The configuration of the OSPF protocol is taken as an example as follows.

Configure PE 1.

```
PE1> enable
PE1# configure terminal
PE1(config)# interface gigabitEthernet 0/1
PE1(config-if-GigabitEthernet 0/1)# ip address 10.1.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/1)# exit
PE1(config)# interface loopback 0
PE1(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
PE1(config-if-Loopback 0)# exit
PE1(config)# router ospf 1
PE1(config-router)# network 10.1.1.1 0.0.0.255 area 0
PE1(config-router)# network 1.1.1.1 0.0.0.0 area 0
```

Configure device P.

```
P> enable
P# configure terminal
P(config)# mpls enable
P(config)# interface gigabitEthernet 0/0
P(config-if-GigabitEthernet 0/0)# ip address 10.1.1.2 255.255.255.0
P(config-if-GigabitEthernet 0/0)# exit
P(config)# interface gigabitEthernet 0/1
P(config-if-GigabitEthernet 0/1)# ip address 20.1.1.1 255.255.255.0
P(config-if-GigabitEthernet 0/1)# exit
P(config)# interface loopback 0
P(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
P(config-if-Loopback 0)# exit
P(config)# router ospf 1
P(config-router)# network 10.1.1.2 0.0.0.255 area 0
P(config-router)# network 20.1.1.1 0.0.0.255 area 0
P(config-router)# network 2.2.2.2 0.0.0.0 area 0
```

Configure PE 2.

```
PE2> enable
PE2# configure terminal
PE2(config)# interface gigabitEthernet 0/1
PE2(config-if-GigabitEthernet 0/1)# ip address 20.1.1.2 255.255.255.0
PE2(config-if-GigabitEthernet 0/1)# exit
PE2(config)# interface loopback 0
PE2(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
PE2(config-if-Loopback 0)# exit
PE2(config)# router ospf 1
PE2(config-router)# network 20.1.1.1 0.0.0.255 area 0
PE2(config-router)# network 3.3.3.3 0.0.0.0 area 0
```

(2) Configure MPLS basic functions and MPLS LDP, and establish LDP LSP on the backbone network.

Configure PE 1.

```
PE1> enable
PE1# configure terminal
PE1(config)# mpls enable
PE1(config)# interface gigabitEthernet 0/1
PE1(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE1(config-if-GigabitEthernet 0/1)# label-switching
PE1(config-if-GigabitEthernet 0/1)# exit
PE1(config)# mpls router ldp
PE1(config-mpls-router)# ldp router-id interface Loopback 0 force
PE1(config-mpls-router)# exit
```

Configure device P.

```
P> enable
P# configure terminal
P(config)# mpls enable
```

```
P(config)# interface gigabitEthernet 0/0
P(config-if-GigabitEthernet 0/0)# mpls ldp enable
P(config-if-GigabitEthernet 0/0)# label-switching
P(config-if-GigabitEthernet 0/0)# exit
P(config)# interface gigabitEthernet 0/1
P(config-if-GigabitEthernet 0/1)# mpls ldp enable
P(config-if-GigabitEthernet 0/1)# label-switching
P(config-if-GigabitEthernet 0/1)# exit
P(config)# mpls router ldp
P(config-mpls-router)# ldp router-id interface Loopback 0 force
P(config-mpls-router)# exit
```

Configure PE 2.

```
PE2> enable
PE2# configure terminal
PE2(config)# mpls enable
PE2(config)# interface gigabitEthernet 0/1
PE2(config-if-GigabitEthernet 0/1)# mpls ldp enable
PE2(config-if-GigabitEthernet 0/1)# label-switching
PE2(config-if-GigabitEthernet 0/1)# exit
PE2(config)# mpls router ldp
PE2(config-mpls-router)# ldp router-id interface Loopback 0 force
PE2(config-mpls-router)# exit
```

(3) Configure a single-protocol VRF instance on the PE and bind it with interfaces between VPN sites.

Configure PE 1.

```
PE1> enable
PE1# configure terminal
PE1(config)# ip vrf vpn1
PE1(config-vrf)# rd 100:1
PE1(config-vrf)# route-target both 100:1 evpn
PE1(config-vrf)# evpn mpls routing enable
PE1(config-vrf)# exit
PE1(config)# interface gigabitEthernet 0/0
PE1(config-if-GigabitEthernet 0/0)# ip vrf forwarding vpn1
PE1(config-if-GigabitEthernet 0/0)# ip address 172.18.1.1 255.255.255.0
PE1(config-if-GigabitEthernet 0/0)# exit
PE1(config)# router bgp 100
PE1(config-router)# address-family ipv4 vrf vpn1
PE1(config-router-af)# redistribute connected
```

Configure PE 2.

```
PE2> enable
PE2# configure terminal
PE2(config)# ip vrf vpn1
PE2(config-vrf)# rd 100:1
PE2(config-vrf)# route-target both 100:1 evpn
```

```
PE2(config-vrf)# evpn mpls routing enable
PE2(config-vrf)# exit
PE2(config)# interface gigabitEthernet 0/0
PE2(config-if-GigabitEthernet 0/0)# ip vrf forwarding vpn1
PE2(config-if-GigabitEthernet 0/0)# ip address 172.18.2.1 255.255.255.0
PE2(config-if-GigabitEthernet 0/0)# exit
PE2(config)# router bgp 100
PE2(config-router)# address-family ipv4 vrf vpn1
PE2(config-router-af)# redistribute connected
```

(4) Establish interactive EVPN routes between BGP EVPN peers on PEs.

Configure PE 1.

```
PE1> enable
PE1# configure terminal
PE1(config)# router bgp 100
PE1(config-router)# neighbor 3.3.3.3 remote-as 100
PE1(config-router)# neighbor 3.3.3.3 update-source loopback 0
PE1(config-router)# address-family l2vpn evpn
PE1(config-router-af)# neighbor 3.3.3.3 activate
PE1(config-router-af)# neighbor 3.3.3.3 send-community extended
PE1(config-router-af)# neighbor 3.3.3.3 advertise encap-type mpls
PE1(config-router-af)# exit
```

Configure PE 2.

```
PE2> enable
PE2# configure terminal
PE2(config)# router bgp 100
PE2(config-router)# neighbor 1.1.1.1 remote-as 100
PE2(config-router)# neighbor 1.1.1.1 update-source loopback 0
PE2(config-router)# address-family l2vpn evpn
PE2(config-router-af)# neighbor 1.1.1.1 activate
PE2(config-router-af)# neighbor 1.1.1.1 send-community extended
PE1(config-router-af)# neighbor 1.1.1.1 advertise encap-type mpls
PE2(config-router-af)# exit
```

(5) Configure the advertisement of IP routes as EVPN prefix routes on the PEs.

Configure PE 1.

```
PE1> enable
PE1# configure terminal
PE1(config)# router bgp 100
PE1(config-router)# address-family l2vpn evpn
PE1(config-router-af)# advertise ipv4 unicast
PE1(config-router-af)# exit
```

Configure PE 2.

```
PE2> enable
PE2# configure terminal
PE2(config)# router bgp 100
```

```
PE2(config-router)# address-family l2vpn evpn
PE2(config-router-af)# advertise ipv4 unicast
PE2(config-router-af)# exit
```

5. Verification

- Run the **show bgp l2vpn evpn all neighbor** command on the PE to display the status of BGP EVPN peers.
- Run the **show bgp l2vpn evpn all** command on the PE to display the BGP EVPN routing information.
- Run the **show ip route vrf** *vrf-name* command on the PE to display VRF routing information.

6. Configuration Files

- PE 1 configuration file:

```
!
ip vrf vpn1
 rd 100:1
 route-target both 100:1 evpn
 evpn mpls routing enable
!
mpls enable
!
interface GigabitEthernet 0/0
 ip vrf forwarding vpn1
 ip address 172.18.1.1 255.255.255.0
!
interface GigabitEthernet 0/1
 ip address 10.1.1.1 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 1.1.1.1 255.255.255.255
!
router bgp 100
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback 0
 address-family ipv4
  neighbor 3.3.3.3 activate
  exit-address-family
 address-family l2vpn evpn
  advertise ipv4 unicast
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 3.3.3.3 advertise encap-type mpls
```

```
  exit-address-family
  !
 address-family ipv4 vrf vpn1
  redistribute connected
  exit-address-family
!
router ospf 1
 graceful-restart
 network 1.1.1.1 0.0.0.0 area 0
 network 10.1.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface loopback 0 force
 graceful-restart
!
```

- PE 2 configuration file:

```
!
ip vrf vpn1
 rd 100:1
 route-target both 100:1 evpn
 evpn mpls routing enable
!
mpls enable
!
interface GigabitEthernet 0/0
 ip vrf forwarding vpn1
 ip address 172.18.2.1 255.255.255.0
!
interface GigabitEthernet 0/1
 ip address 20.1.1.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface Loopback 0
 ip address 3.3.3.3 255.255.255.255
!
!
router bgp 100
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source loopback 0
 address-family ipv4
  neighbor 1.1.1.1 activate
```

```
  exit-address-family
 address-family l2vpn evpn
  advertise ipv4 unicast
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community extended
  neighbor 1.1.1.1 advertise encap-type mpls
  exit-address-family
  !
 address-family ipv4 vrf vpn1
  redistribute connected
  exit-address-family
!
router ospf 1
 graceful-restart
 network 3.3.3.3 0.0.0.0 area 0
 network 20.1.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface loopback 0 force
 graceful-restart
!
```

- Device P configuration file:

```
!
mpls enable
!
interface GigabitEthernet 0/0
 ip address 10.1.1.2 255.255.255.0
 mpls ldp enable
 label-switching
!
interface GigabitEthernet 0/1
 ip address 20.1.1.1 255.255.255.0
 mpls ldp enable
 label-switching
!
router ospf 1
 graceful-restart
 network 2.2.2.2 0.0.0.0 area 0
 network 10.1.1.0 0.0.0.255 area 0
 network 20.1.1.0 0.0.0.255 area 0
!
mpls router ldp
 ldp router-id interface loopback 0 force
 graceful-restart
!
```

Contents

1 Configuring IPsec

1.1 Introduction

1.1.1 Overview

Currently, Internet Protocol version 4 (IPv4) is the most widely used network protocol. However, security is not taken into consideration in the design of this protocol. Malicious users can forge addresses of IP packets, tamper with packet content, retransmit the same IP packets repeatedly, and intercept and check packet content at will, which bring many security risks to networks.

IP Security (IPsec) is an L3 tunnel encryption protocol formulated by the Internet Engineering Task Force (IETF). It provides high-quality, interoperable, and cryptography-based security guarantee for data transmitted over networks. Communication parties use IPsec to encrypt and authenticate data sources at the IP layer, in a bid to ensure communication security. IPsec provides the following security services:

- Data confidentiality: An IPsec sender encrypts packets before sending them over a network.
- Data integrity: An IPsec receiver authenticates packets sent by a sender to ensure that data is not tampered with during transmission.
- Data source authentication: An IPsec receiver authenticates an IPsec sender.
- Anti-replay: An IPsec receiver can detect and reject outdated or duplicate packets.

The Internet Key Exchange (IKE) protocol provides IPsec with services of automatically negotiating keys, and establishing and maintaining security associations (SAs). It helps simplify the use and management of IPsec. IKE negotiation is not mandatory. Policies and algorithms used by IPsec can also be manually configured.

1.1.2 Basic Concepts

1. Security Association

IPsec provides secure communication between two endpoints, which are called IPsec peers.

Security associations (SAs) are the foundation and essence of IPsec. An SA is an agreement on certain elements between communication peers, such as the protocol to be used (Authentication Header (AH), Encapsulating Security Payload (ESP), or both), protocol encapsulation mode (transport mode or tunnel mode), encryption algorithm (Data Encryption Standard (DES), Triple DES (3DES), or Advanced Encryption Standard (AES)), authentication algorithm (secure hash algorithm (SHA) or message digest algorithm 5 (MD5)), shared keys used to protect data in a specific flow, and key lifetime.

An SA is unidirectional but communication between two peers is bidirectional. At least one SA is needed to protect the data flow in each direction. Therefore, at least two SAs are required for secure communication between two peers. If two peers use both AH and ESP for secure communication, each peer constructs an independent SA for each protocol.

An SA is uniquely identified by a triplet, which contains the security parameter index (SPI), destination IP address, and security protocol number (AH or ESP). An SPI is a 32-bit value that uniquely identifies an SA and is transmitted in the AH and ESP headers.

SAs can be generated via manual configuration and automatic IKE negotiation.

- Manual configuration: You need to run commands to configure all information about an SA. This mode is independent of IKE but you need to update keys periodically to ensure security. The manual mode is applicable to small-sized static networking environments.
- IKE automatic negotiation: An SA is automatically generated and maintained by IKE. The configuration is simpler and this mode is more scalable than the manual mode. IKE automatic negotiation is applicable to large-sized dynamic networking environments.

Manually configured SAs never age whereas SAs automatically negotiated by IKE are valid in their lifetime. There are two types of lifetime:

- Time-based lifetime: Defines the duration from the establishment to expiration of an SA.
- Traffic-based lifetime: Defines the maximum traffic that can be processed by an SA.

You can configure both time-based and traffic-based SA lifetime. An SA will expire once its lifetime reaches specified time or traffic. Before an SA expires, IKE negotiates and establishes a new SA for IPsec to ensure that the new SA is ready before the expiration of the old SA. The old SA is still used to protect communication before the new SA is successfully negotiated. The new SA is used to protect communication immediately after it is successfully negotiated.

2. Security Protocols

IPsec implements security services by using the following two protocols:

- AH: The protocol number is 51. This protocol provides data source authentication, data integrity verification, and packet replay prevention. Available authentication algorithms include MD5 and Secure Hash Algorithm 1 (SHA-1). AH ensures the integrity and authenticity of data packets, and prevents hackers from intercepting data packets or inserting forged data packets into the network.
- ESP: The protocol number is 50. Different from AH, ESP encrypts user data to be protected and then encapsulates the data into IP packets to ensure data confidentiality. Common encryption algorithms include DES, 3DES, and AES. In addition, you can use the MD5 or SHA-1 algorithm to ensure packet integrity and authenticity.

Table 1-1 Comparison Between AH and ESP

Security Service	AH	ESP
Data confidentiality	Not supported	Supported
Data integrity	Supported (entire IP packet)	Supported (excluding the IP header)
Data source authentication	Supported	Supported
Anti-replay	Supported	Supported
NAT traversal	Not supported	Supported

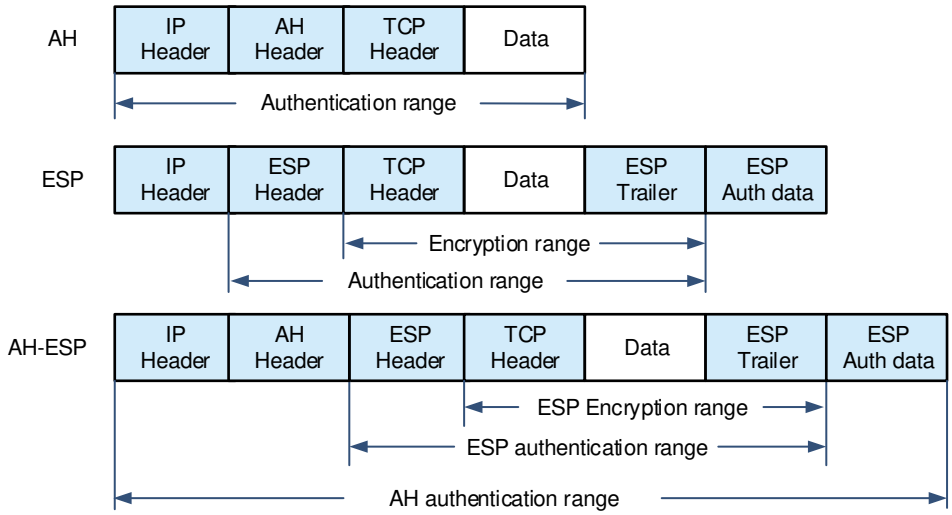
AH and ESP can be used independently or in combination. The device supports the combination of AH and ESP as follows: The device encapsulates packets through ESP and then uses AH to encapsulate packets. If you use AH first and then ESP, the length of a data packet will be changed due to the header, trailer, and padding fields of ESP. However, AH authenticates the entire IP data packet, and an authentication failure is caused.

3. Encapsulation Mode

IPsec supports two work modes:

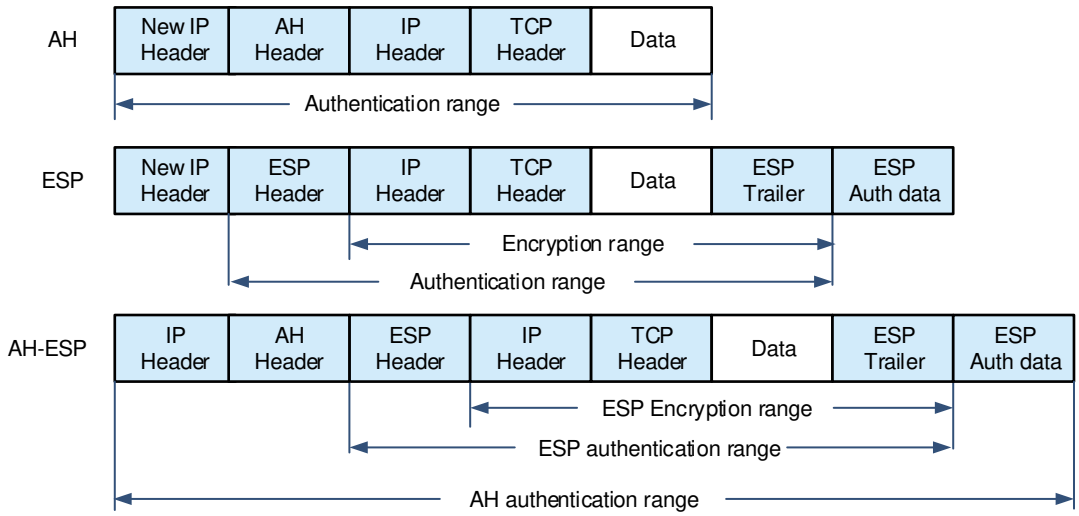
- Transport mode: The AH or ESP header and ESP-encrypted user data are placed after the original IP packet header. The transport mode is typically applied in the communication between two hosts.

Figure 1-1 Packet Encapsulation in Transport Mode



- Tunnel mode: The AH or ESP header and ESP-encrypted user data are encapsulated into a new IP packet. The tunnel mode is typically applied in the communication between two security gateways.

Figure 1-2 Packet Encapsulation in Tunnel Mode



4. Authentication Algorithm

The authentication algorithm is mainly implemented using a hash function. The hash function can accept the input of messages of any length and output a message digest of a fixed length. IPsec peers compute digests separately. If the two digests are the same, a packet is intact and not tampered with. IPsec uses the following authentication algorithms:

- MD5: Generates a 128-bit message digest based on an input message of any length.
- SHA-1: Generates a 160-bit message digest based on an input message with the bit length less than the 64th power of 2.
- SHA2-256: Generates a 256-bit message digest based on an input message with the bit length less than the 64th power of 2.
- SHA2-384: Generates a 384-bit message digest based on an input message with the bit length less than the 128th power of 2.
- SHA2-512: Generates a 512-bit message digest based on an input message with the bit length less than the 128th power of 2.
- SM3: Generates a 256-bit message digest based on an input message with the bit length less than the 64th power of 2.

A longer message digest indicates higher security and slower computation.

5. Encryption Algorithm

IPsec adopts encryption algorithms using symmetric keys, and encrypts and decrypts data with the same key. Currently, IPsec on the device supports three encryption algorithms:

- DES: Encrypts 64-bit plaintext by using a 56-bit key.
- 3DES: Encrypts plaintext by using three 56-bit DES keys.
- AES: Encrypts plaintext by using a 128-bit, 192-bit, or 256-bit key.
- SM4: Encrypts plaintext by using a 128-bit key.

The SM4 and AES encryption algorithms have higher security and faster computation speed than the 3DES and DES algorithms. You are advised to configure the AES and 3DES algorithms.

1.1.3 IPsec Tunnel

You can configure a static tunnel policy on the CLI to create a common manual tunnel. In the establishment of a tunnel via IKE negotiation, an IKE-encrypted tunnel is established through IKE and then an IPsec tunnel is negotiated through the IKE tunnel. The IKE tunnel and IPsec tunnel are independent of each other. If the IKE tunnel is deleted, the IPsec tunnel still exists. Likewise, the IKE tunnel can exist independently after the IPsec tunnel is deleted. Therefore, IPsec tunnel-related control takes effect only on IPsec tunnels.

1.1.4 Protocols and Standards

- RFC 2401: Security Architecture for the Internet Protocol. S. Kent, R. Atkinson. November 1998. (Format: TXT=168162 bytes) (Obsoletes RFC 1825) (Obsoleted by RFC 4301) (Updated by RFC 3168) (Status: PROPOSED STANDARD)
- RFC 2402: IP Authentication Header. S. Kent, R. Atkinson. November 1998. (Format: TXT=52831 bytes) (Obsoletes RFC 1826) (Obsoleted by RFC 4302, RFC 4305) (Status: PROPOSED STANDARD)
- RFC 2403: The Use of HMAC-MD5-96 within ESP and AH. C. Madson, R. Glenn. November 1998. (Format: TXT=13578 bytes) (Status: PROPOSED STANDARD)
- RFC 2404: The Use of HMAC-SHA-1-96 within ESP and AH. C. Madson, R. Glenn. November 1998. (Format: TXT=13089 bytes) (Status: PROPOSED STANDARD)
- RFC 2405: The ESP DES-CBC Cipher Algorithm With Explicit IV. C. Madson, N. Doraswamy. November

1998. (Format: TXT=20208 bytes) (Status: PROPOSED STANDARD)

- RFC 2406: IP Encapsulating Security Payload (ESP). S. Kent, R. Atkinson. November 1998. (Format: TXT=54202 bytes) (Obsoletes RFC 1827) (Obsoleted by RFC 4303, RFC 4305) (Status: PROPOSED STANDARD)
- RFC 3948: UDP Encapsulation of IPsec ESP Packets. A. Huttunen, B. Swander, V. Volpe, L. DiBurro, M. Stenberg. January 2005. (Format: TXT=30366bytes) (Status: PROPOSED STANDARD)

1.2 Restrictions and Guidelines

- Interesting flows in transport mode must be in host-host mode. Otherwise, negotiation is carried out as the flows are in tunnel mode.
- Interesting flow conflicts cannot be detected. Interesting flows specified in static crypto map entries are matched based on the configuration sequence. Dynamic crypto map entries learned or configured later have a higher priority for matching.

1.3 Configuration Task Summary

IPsec configuration includes the following tasks:

(1) Configure an IPsec tunnel. Configure one of the following tasks.

- o Configuring an IPsec Tunnel to Protect Packets Matching a Crypto ACL
- o Configuring an IPsec Tunnel to Protect Packets on a Tunnel Interface
- o Configuring an IPsec Tunnel for Routing Protocol Authentication

(2) (Optional) Configuring XAUTH Authentication

a Creating a Crypto ACLDefining a Transform Set

b Configure crypto map entries. Select one of the following to configure:

Configuring a Manual Crypto Map Entry

Creating a Static Crypto Map Entry

Creating a Dynamic Crypto Map Entry

c Creating a Client Address Pool

d Configuring a Client Policy

e Configuring XAUTH Authentication Mode for a Crypto Map Entry

f (Optional)Configuring XAUTH Domain Authentication

g (Optional)Configuring the XAUTH Timeout

h (Optional)Configuring the Timeout Period for Waiting for the AAA Server Response in XAUTH

i (Optional)Configuring XAUTH to

j (Optional)Configuring the Device Not to Forcibly Use XAUTH for IKE Negotiation

k Applying IPsec to an Interface

(3) (Optional) Configuring Optional Features of IPsec. All the configuration tasks below are optional. Select the configuration tasks as required.

- o Configuring the Global IPsec SA Lifetime

- Configuring IPsec SA Lifetime for a Specified Crypto Map
- Configuring the DF Bit Override Function for IPsec Tunnels
- Disabling IPsec Encapsulation for Multicast and Broadcast Packets
- Disabling IPsec Check
- Specifying the IPsec Local Address
- Disabling Packet Retransmission Check
- Configuring the Matching Rule for Lifetime Negotiation for IPsec Phase 2
- Configuring the MTU for the IPsec Pre-Fragmentation Mode
- Configuring RRI
- Configuring the Diffie-Hellman Group Identifier for IPsec Tunnel Encapsulation
- Setting the Work Mode to Tunnel Autoup
- Specifying the Local IP Address in a Crypto Map Entry
- Configuring Packet Matching VRF Before Encryption in a Specified Crypto Map
- Configuring the VRF to Which Decrypted Packets Belong After the Specified Crypto Map Is Configured
- Configuring the Negotiation Mode of a Specified Crypto Map
- Disabling Packet Filtering After Decryption
- Configuring Automatic Disconnection of Idle IPsec Tunnels Globally
- Configuring Automatic Disconnection of Idle IPsec Tunnels with a Specified Crypto Map
- Configuring the Bypass Function for IPsec Tunnels Globally
- Configuring the Global IPsec MIB Function
- Configuring Interesting Traffic with a Wildcard Mask of All Zeros

1.4 Configuring an IPsec Tunnel to Protect Packets Matching a Crypto ACL

1.4.1 Configuration Tasks

The tasks of configuring an IPsec tunnel to protect packets matching a crypto access control list (ACL) include the following:

(1) Creating a Crypto ACL

(2) Defining a Transform Set

(3) Configure crypto map entries. Select one of the following to configure:

- Configuring a Manual Crypto Map Entry
- Creating a Static Crypto Map Entry
- Creating a Dynamic Crypto Map Entry

(4) Applying IPsec to an Interface

1.4.2 Creating a Crypto ACL

1. Overview

Creating a crypto ACL is defining data flows to be protected. IPsec filters sent and received data packets according to a crypto ACL, protects matched sent packets, and checks the validity of matched received packets.

A crypto ACL is actually an extended ACL and is referenced in a crypto map entry. A crypto ACL is mandatory when a static crypto map is configured. In dynamic crypto map mode, a crypto ACL can be learned. In tunnel interface mode, a crypto ACL can be learned based on the tunnel configuration.

A crypto ACL specified in an IPsec crypto map entry supports the following functions:

- The deny rules in a referenced ACL are not used for tunnel negotiation. Data that matches the deny rules will not be encrypted.
- The crypto ACL filters out outbound communication data encrypted and protected by IPsec. The image filtering policy is automatically generated and it does not need to be configured in both directions.
- In the processing of inbound communication, the crypto ACL aims to filter out and discard communication packets that should be protected by IPsec but are actually not.
- In the negotiation of an IPsec SA, the crypto ACL specifies the data flows to be protected by the new SA.
- In the processing of IKE negotiation initiated by IPsec peers, the crypto ACL determines whether to accept the IPsec SA request initiated for data flows (negotiation is required only for IPsec Internet Security Association and Key Management Protocol (ISAKMP) crypto map entries). Ensure that ACLs on peers at both ends must be matched. You are advised to configure the same ACL on both peers.

2. Restrictions and Guidelines

- IPsec filters sent and received packets according to a crypto ACL. Crypto map entries in a crypto map configured on an interface are used to protect different interesting flows. Configured crypto map entries should not conflict with each other. Otherwise, a tunnel configured later cannot forward data.
- Interesting flows must be configured in static crypto maps.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a crypto ACL.

access-list *access-list-number* { **deny** | **permit** } *protocol source source-wildcard destination destination-wildcard* [**log**]

No crypto ACL exists by default.

1.4.3 Defining a Transform Set

1. Overview

A transform set defines how to protect data flows. A transform set specifies the algorithm, security protocol, and data encapsulation mode. You need to configure a transform set to define the protection degree and requirements.

During IPsec SA negotiation, peers must use the same specific transform set to protect specific data flows.

You can configure multiple transform sets and then specify one or several of them in crypto map entries. Transform sets defined in crypto map entries are used to negotiate IPsec SAs, so as to protect data flows that match the ACLs specified in the crypto map entries. During negotiation, both peers search for the same transform set that is available on both peers. When such a transform set is found, it is selected as a part of the IPsec SAs used by both peers to protect communication data.

If an SA is established via manual configuration, no parameter needs to be negotiated for the SA. Therefore, the same transform set must be specified on both peers.

The following table describes all transform sets supported by the system.

Table 1-2 List of Transform Sets

Algorithm Combination	Description
ah-md5-hmac	AH protocol and MD5 HMAC authentication algorithm
ah-sha-hmac	AH protocol and SHA HMAC authentication algorithm
esp-des	ESP protocol and DES encryption algorithm
esp-aes-128	ESP protocol and AES encryption algorithm using a 128-bit key
esp-aes-192	ESP protocol and AES encryption algorithm using a 192-bit key
esp-aes-256	ESP protocol and AES encryption algorithm using a 256-bit key
ah-md5-hmac esp-des	AH protocol and MD5 HMAC authentication algorithm outside; ESP protocol and DES encryption algorithm inside
ah-sha-hmac esp-des	AH protocol and SHA HMAC authentication algorithm outside; ESP protocol and DES encryption algorithm inside
ah-md5-hmac esp-des esp-md5-hmac	AH protocol and MD5 HMAC authentication algorithm outside; ESP protocol, DES encryption algorithm, and MD5 HMAC authentication algorithm inside
ah-md5-hmac esp-null esp-md5-hmac	AH protocol and MD5 HMAC authentication algorithm outside; ESP protocol, zero encryption algorithm, and MD5 HMAC authentication algorithm inside
ah-md5-hmac esp-des esp-sha-hmac	AH protocol and MD5 HMAC authentication algorithm outside; ESP protocol, DES encryption algorithm, and SHA HMAC authentication algorithm inside
ah-md5-hmac esp-null esp-sha-hmac	AH protocol and MD5 HMAC authentication algorithm outside; ESP protocol, zero encryption algorithm, and SHA HMAC authentication algorithm inside

Algorithm Combination	Description
ah-sha-hmac esp-des esp-md5-hmac	AH protocol and SHA HMAC authentication algorithm outside; ESP protocol, DES encryption algorithm, and MD5 HMAC authentication algorithm inside
ah-sha-hmac esp-null esp-md5-hmac	AH protocol and SHA HMAC authentication algorithm outside; ESP protocol, zero encryption algorithm, and MD5 HMAC authentication algorithm inside
ah-sha-hmac esp-des esp-sha-hmac	AH protocol and SHA HMAC authentication algorithm outside; ESP protocol, DES encryption algorithm, and SHA HMAC authentication algorithm inside
ah-sha-hmac esp-null esp-sha-hmac	AH protocol and SHA HMAC authentication algorithm outside; ESP protocol, zero encryption algorithm, and SHA HMAC authentication algorithm inside
esp-des esp-md5-hmac	ESP protocol, DES encryption algorithm, and MD5 HMAC authentication algorithm
esp-null esp-md5-hmac	ESP protocol, zero encryption algorithm, and MD5 HMAC authentication algorithm
esp-des esp-sha-hmac	ESP protocol, DES encryption algorithm, and SHA HMAC authentication algorithm
esp-null esp-sha-hmac	ESP protocol, zero encryption algorithm, and SHA HMAC authentication algorithm
esp-3des	ESP protocol and 3DES encryption algorithm
esp-3des esp-sha	ESP protocol, 3DES encryption algorithm, and SHA HMAC authentication algorithm
esp-3des esp-md5	ESP protocol, 3DES encryption algorithm, and MD5 HMAC authentication algorithm
ah-md5-hmac esp-des	AH protocol and MD5 HMAC authentication algorithm outside; ESP protocol and 3DES encryption algorithm inside
ah-sha-hmac esp-3des	AH protocol and SHA HMAC authentication algorithm outside; ESP protocol and 3DES encryption algorithm inside
ah-md5-hmac esp-3des esp-sha	AH protocol and MD5 HMAC authentication algorithm outside; ESP protocol, 3DES encryption algorithm, and SHA HMAC authentication algorithm inside
ah-sha-hmac esp-3des esp-sha	AH protocol and SHA HMAC authentication algorithm outside; ESP protocol, 3DES encryption algorithm, and SHA HMAC authentication algorithm inside
ah-md5-hmac esp-3des esp-md5	AH protocol and MD5 HMAC authentication algorithm outside; ESP protocol, 3DES encryption algorithm, and MD5 HMAC authentication algorithm inside
ah-sha-hmac esp-3des esp-md5	AH protocol and SHA HMAC authentication algorithm outside; ESP protocol, 3DES encryption algorithm, and MD5 HMAC authentication algorithm inside

2. Restrictions and Guidelines

- In general, the esp-des combination (without data authentication) can meet requirements. If data needs to be authenticated, you can use esp-des esp-md5-hmac or esp-des esp-sha-hmac.
- A transform set must be configured and can be referenced in multiple crypto maps. Multiple transform sets can be configured in one crypto map. Transform sets are matched by priority, and repetitive content of transform sets does not affect negotiation results.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Define a transform set for SA negotiation.

crypto ipsec transform-set *transform-set-name transform*&<1-3>

No transform set is configured by default.

(4) Change the mode for the transform set.

mode { **transport** | **tunnel** }

The default encapsulation mode of transform sets is tunnel mode.

Mode setting is effective only to communication using addresses of IPsec peers as the source and destination addresses. Other communication is made in tunnel mode.

If the source and destination addresses of the communication to be protected are those of IPsec peers and the transport mode is specified, the device requests the transport mode during negotiation but accepts both the transport mode and tunnel mode. If the tunnel mode is specified, the device requests the tunnel mode and accepts only the tunnel mode.

1.4.4 Configuring a Manual Crypto Map Entry

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create or modify a manual crypto map entry.

crypto map *map-name sequence-number* **ipsec-manual**

No crypto map entry is configured by default.

(4) Specify a remote peer for the crypto map entry.

set peer { *hostname* | *ipv4-address* | *ipv6-address* } [*local-trustpoint*]

No remote peer is specified for a crypto map entry by default.

A remote peer must be specified for a crypto map entry. You can configure multiple remote peers. Negotiation is initiated in the configured peer sequence. When the negotiation with a peer fails, the next peer IP address will be used for negotiation.

(5) Configure the SPI and key for the inbound direction.

set session-key inbound { **ah** *spi hex-key-data* | **esp** *spi* { **cipher** *hex-key-data* [**authenticator** *hex-key-data*] | **authenticator** *hex-key-data* }

The SPI and key are not configured by default.

(6) Configure the SPI and key for the outbound direction.

set session-key outbound { **ah** *spi hex-key-data* | **esp** *spi* { **cipher** *hex-key-data* [**authenticator** *hex-key-data*] | **authenticator** *hex-key-data* }

The SPI and key are not configured by default.

(7) (Optional) Specify an ACL for the crypto map entry.

match address *access-list-id*

No ACL is configured for a crypto map entry by default.

The ACL specified by this command is applied to both outbound and inbound communication data. If it is detected that outbound data matches an ACL and an SA already exists, the device encrypts and forwards the data. If no SA is established, the device triggers the SA negotiation (through IKE). If it is detected that inbound data matches an ACL, the device decrypts encrypted data and directly discards data that is not encrypted.

(8) (Optional) Specify interesting traffic using an IPv6 ACL.

match ipv6 *ipv6-acl-name*

No interesting traffic is specified through an IPv6 ACL by default.

(9) Specify a transform set for the crypto map entry.

set transform-set *transform-set-name*&<1-6>

No transform set is configured for a crypto map entry by default.

A transform set must be specified for a crypto map entry. You can configure multiple transform sets and select one of them for SA negotiation.

1.4.5 Creating a Static Crypto Map Entry

1. Overview

A crypto map entry is used to associate a predefined ACL with transform sets and define keys and peer addresses to form a complete IPsec solution.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create or modify a static crypto map entry.

crypto map *map-name sequence-number* **ipsec-isakmp**

No crypto map entry is configured by default.

(4) Specify a remote peer for the crypto map entry.

set peer { *hostname* | *ipv4-address* | *ipv6-address* } [*local-trustpoint*]

No remote peer is specified for a crypto map entry by default.

A remote peer must be specified for a crypto map entry. You can configure multiple remote peers. Negotiation is initiated in the configured peer sequence. When the negotiation with a peer fails, the next peer IP address will be used for negotiation.

(5) (Optional) Specify an ACL for the crypto map entry.

match address *access-list-id*

No ACL is configured for a crypto map entry by default.

The ACL specified by this command is applied to both outbound and inbound communication data. If it is detected that outbound data matches an ACL and an SA already exists, the device encrypts and forwards the data. If no SA is established, the device triggers the SA negotiation (through IKE). If it is detected that inbound data matches an ACL, the device decrypts encrypted data and directly discards data that is not encrypted.

(6) (Optional) Specify interesting traffic using an IPv6 ACL.

match ipv6 *ipv6-acl-name*

By default, interesting traffic is not specified through an IPv6 ACL.

(7) Specify a transform set for the crypto map entry.

set transform-set *transform-set-name*&<1-6>

No transform set is configured for a crypto map entry by default.

A transform set must be specified for a crypto map entry. You can configure multiple transform sets and select one of them for SA negotiation.

1.4.6 Creating a Dynamic Crypto Map Entry

1. Overview

Dynamic crypto map entries apply to scenarios with unknown peer addresses. The device, on which a dynamic crypto map entry is configured, cannot initiate negotiation but only responds to negotiation requests from a peer device.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a dynamic crypto map entry and enter the crypto map configuration mode.

crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

No dynamic crypto map is configured by default.

(4) (Optional) Specify a remote peer for the crypto map entry.

set peer { *hostname* | *ipv4-address* | *ipv6-address* } [*local-trustpoint*]

No remote peer is specified for a crypto map entry by default.

(5) (Optional) Specify an ACL for the crypto map entry.

match address *access-list-id*

No ACL is configured for a crypto map entry by default.

The ACL specified by this command is applied to both outbound and inbound communication data. If it is detected that outbound data matches an ACL and an SA already exists, the device encrypts and forwards the data. If no SA is established, the device triggers the SA negotiation (through IKE). If it is detected that inbound data matches an ACL, the device decrypts encrypted data and directly discards data that is not encrypted.

(6) (Optional) Specify interesting traffic using an IPv6 ACL.

match ipv6 *ipv6-acl-name*

No interesting traffic is specified through an IPv6 ACL by default.

(7) (Optional) Configure the dynamic crypto map set to deny any-to-any data flows from a remote peer.

match no-any-to-any

A dynamic crypto map set does not reject any-to-any data flows from a remote peer by default.

(8) (Optional) Configure an ACL for the dynamic crypto map set.

match range-address *acl-number*

No ACL is configured for a dynamic crypto map set by default.

(9) Specify a transform set for the crypto map entry.

set transform-set *transform-set-name*&<1-6>

No transform set is configured for a crypto map entry by default.

A transform set must be specified for a crypto map entry. You can configure multiple transform sets and select one of them for SA negotiation.

(10) Return to the global configuration mode.

exit

(11) Create a crypto map and specify the dynamic crypto map entry as a policy template of the crypto map.

crypto map *map-name sequence-number* **ipsec-isakmp dynamic** *dynamic-map-name*

No crypto map entry is configured by default.

1.4.7 Applying IPsec to an Interface

1. Overview

To activate a defined IPsec solution, you need to apply a crypto map entry to an interface so that the crypto map takes effect on the interface.

2. Restrictions and Guidelines

- Before IPsec is applied to an interface, all IPsec configurations do not take effect.
- An IPsec tunnel can be applied only to an L3 interface. It cannot be configured on L2 interfaces such as switch interfaces.
- The same crypto map can be applied to multiple interfaces, and is independent of each other after applied to different interfaces.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

  interface *ethernetinterface-type interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

  interface *ethernet-type interface-number.subnumber*

- Enter the virtual PPP interface configuration mode.

  interface virtual-ppp *interface-number*

- Enter the virtual VPDN interface configuration mode.

  interface virtual-vpdn *interface-number*

(4) Apply a crypto map entry to the interface.

crypto map *map-name*

No crypto map entry is applied to an interface by default.

If data needs to be encrypted and protected through IPsec on an interface, a crypto map must be applied to the interface. One interface can be associated with only one crypto map. If one crypto map has multiple crypto map entries, which are applied to the same interface, the crypto map entry with a smaller sequence number has a higher priority.

1.5 Configuring an IPsec Tunnel to Protect Packets on a Tunnel Interface

1.5.1 Overview

If data needs to be encrypted and protected on a tunnel interface, you need to create a profile crypto map and then apply it to the IPsec tunnel interface. In the profile crypto map, you need to define parameters for encrypted communication, including:

- IPsec policies to be applied to the communication, which can be selected from a list composed of one or more transform sets
- Lifetime of an SA.
- Whether an SA is established through manual configuration or through IKE

After a profile crypto map is applied to a tunnel interface, all IP communication data passing through the interface is encrypted according to the profile crypto map. The device automatically initiates IKE negotiation after a profile crypto map is applied to a tunnel interface, or triggers IKE negotiation after receiving packets from the interface.

Policies defined in a crypto map entry are used during SA negotiation. To ensure smooth IPsec communication between two IPsec peers, the tunnel crypto map entries of the two peers must contain compatible configuration

statements. When two peers attempt to establish an SA, each peer must have at least one crypto map entry compatible with one crypto map entry of the remote peer. Both peers need to meet the following conditions:

- Crypto map entries must contain compatible crypto ACLs (such as mirror image ACL).
- The crypto map entry of each peer must specify the address of the remote peer (unless the remote peer is using a dynamic crypto map).
- The crypto map entries must have at least one identical transform set.
- Only one crypto map is applied to a single interface.

In either of the following cases, multiple crypto map entries must be created for one interface:

- Different data flows on this interface need to be processed by different IPsec peers.
- Different IPsec policies need to be applied to different types of communication (to the same peer or different peers). For example, the communication between one group of subnets needs to be authenticated while the communication between another group of subnets needs to be authenticated and encrypted. In this case, the communication types need to be defined in two different ACLs, and a separate crypto map entry must be created for each crypto ACL.

1.5.2 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a profile crypto map entry and enter the profile crypto map configuration mode.

crypto ipsec profile *profile-name*

No crypto map entry is configured by default.

(4) (Optional) Specify the interesting flow with the local IP address/mask (0.0.0.0/0.0.0.0) and peer IP address/mask (0.0.0.0/0.0.0.0).

match any

The interesting flow with the local IP address/mask (0.0.0.0/0.0.0.0) and peer IP address/mask (0.0.0.0/0.0.0.0) is not specified by default.

The configuration is mandatory for IPv6, IPsec-IPv4, and IPsec-IPv6 tunnels.

(5) Specify transform sets for the crypto map entry.

set transform-set *transform-set-name*&<1-6>

No transform set is configured for a crypto map entry by default.

A transform set must be specified for a crypto map entry. You can configure multiple transform sets and select one of them for SA negotiation.

(6) Return to the global configuration mode.

exit

(7) Enter the tunnel interface configuration mode.

Interface tunnel *interface-number*

(8) Apply the profile crypto map entry to the tunnel interface.

tunnel protection ipsec profile *profile-name*

1.6 Configuring an IPsec Tunnel for Routing Protocol Authentication

1.6.1 Overview

IPsec authentication is used to authenticate sending and receiving of routing protocol packets to prevent attacks on the device from forged routing protocol packets.

1.6.2 Restrictions and Guidelines

- One IPsec proposal can be applied to multiple IPsec SAs.
- One IPsec SA can be applied to multiple routing protocols.

1.6.3 Configuration Task Summary

IPsec tunnel configuration for routing protocol authentication includes the following tasks:

(1) Configuring an IPsec proposal

(2) Configuring an IPsec SA

(3) Applying the IPsec SA to a routing protocol

1.6.4 Configuring an IPsec Proposal

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Define an IPsec proposal for IPsec SA and enter the IPsec proposal configuration mode.

ipsec proposal proposal-name

The IPsec proposal for IPsec SA is not defined by default.

(4) Configure an encapsulation mode.

encapsulation-mode { transport | **tunnel }**

No encapsulation mode is configured by default.

(5) Configure a security protocol.

transform { ah | **ah-esp** | **esp }**

No security protocol is configured by default.

(6) (Optional) Configure an authentication algorithm for the AH protocol.

ah authentication-algorithm { **md5** | **sha1** | **sha2-256** | **sha2-384** | **sha2-512** | **sm3** }

No authentication algorithm is configured for the AH protocol by default.

(7) (Optional) Configure an authentication algorithm for the ESP protocol.

esp authentication-algorithm { **md5** | **sha1** | **sha2-256** | **sha2-384** | **sha2-512** | **sm3** }

No authentication algorithm is configured for the ESP protocol by default.

(8) (Optional) Configure an encryption algorithm for the ESP protocol.

esp encryption-algorithm { **3des** | **aes-128** | **aes-192** | **aes-256** | **des** | **sm4** }

No encryption algorithm is configured for the ESP protocol by default.

1.6.5 Configuring an IPsec SA

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create an IPsec SA and enter the IPsec SA configuration mode.

ipsec sa *sa-name*

No IPsec SA is created by default.

(4) Configure an IPsec proposal for the IPsec SA.

proposal *proposal-name*

No IPsec proposal is configured for the IPsec SA by default.

(5) Configure a security parameter index (SPI) for the IPsec SA.

sa spi { **ah** *spi* | **esp** *spi* }

No SPI is configured by default.

(6) (Optional) Configure an authentication key in hexadecimal notation.

sa authentication-hex { **ah** | **esp** } [**0** | **7**] *hex-key*

No authentication key in hexadecimal notation is configured by default.

(7) (Optional) Configure an authentication key string.

sa string-key { **ah** | **esp** } [**0** | **7**] *string-key*

No authentication key string is configured by default.

(8) (Optional) Configure an encryption key in hexadecimal notation.

sa encryption-hex esp [**0** | **7**] *encryption-key*

No encryption key in hexadecimal notation is configured by default.

1.6.6 Applying the IPsec SA to a Routing Protocol

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode.

  interface *ethernet-type interface-number*

- Enter the Layer 3 aggregate interface configuration mode.

  interface aggregateport *interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode.

interface *ethernet-type interface-number.subnumber*

- Enter the Layer 3 aggregate sub-interface configuration mode.

  interface aggregateport *interface-number.subnumber*

- Enter the SVI configuration mode.

  interface vlan *interface-number*

- Enter the loopback interface configuration mode.

  interface loopback *interface-number*

- Enter the virtual PPP interface configuration mode.

  interface virtual-ppp *interface-number*

- Enter the virtual VPDN interface configuration mode.

  interface virtual-vpdn *interface-number*

(4) Associate SA authentication with an interface.

(4) **ipv6 ospf ipsec sa** { **disable** | *sa-name* } [**instance** *instance-id*]

No interface is associated with SA authentication by default.

1.7 Configuring XAUTH Authentication for IPsec Clients

1.7.1 Overview

With the rapid development of broadband access, IPsec VPN networks are widely deployed in small- and medium-sized enterprises to provide remote clients with access to the company's central resources. When deploying IPsec VPNs for remote access, network administrators usually need to configure different VPN policies and preset passwords for each client to distinguish them. This is time-consuming and difficult to manage. Therefore, mainstream IPsec VPN gateways on the market provide another solution. That is, by configuring one VPN policy on the VPN gateway, network administrators can permit up to 1000 remote clients for simultaneous access. Network administrators only need to deliver the same policy configuration to the remote clients. This solution is convenient but lacks security because the VPN configuration policies of all remote clients are the same.

To solve this problem, users need a technology that allows a VPN gateway to configure only one policy and authenticate remote clients using different usernames and passwords. This greatly reduces the workload of network management and ensures the security of remote client access, improving the overall work efficiency of enterprises. Extended Authentication (XAUTH) is such a technology integrated into IPsec VPN. It provides an identity authentication mechanism for applications that need to authenticate users. This mechanism allows the VPN gateway to use the user information in the RADIUS server or local database to authenticate users. This authentication mode is at the same level as pre-shared key authentication and digital certificate authentication. According to the standard, it is defined as XAUTH pre-shared key authentication and XAUTH digital certificate authentication. and the first packet carries this information in the negotiation process, which is different from the negotiation process of common IPsec.

1.7.2 Configuration Tasks

(1) Creating a Crypto ACLDefining a Transform Set

(2) Configure crypto map entries. Select one of the following to configure:

- Configuring a Manual Crypto Map Entry
- Creating a Static Crypto Map Entry
- Creating a Dynamic Crypto Map Entry

(3) Creating a Client Address Pool

(4) Configuring a Client Policy

(5) Configuring XAUTH Authentication Mode for a Crypto Map Entry

(6) (Optional)Configuring XAUTH Domain Authentication

(7) (Optional)Configuring the XAUTH Timeout

(8) (Optional)Configuring the Timeout Period for Waiting for the AAA Server Response in XAUTH

(9) (Optional)Configuring XAUTH to

(10) (Optional)Configuring the Device Not to Forcibly Use XAUTH for IKE Negotiation

(11) Applying IPsec to an Interface

1.7.3 Preparation

Complete the basic configuration for an IPsec tunnel to protect packets matching a crypto ACL. For details, see1.4 Configuring an IPsec Tunnel to Protect Packets Matching a Crypto ACL.

1.7.4 Creating a Client Address Pool

1. Overview

When XAUTH is used to negotiate an IPsec tunnel, the IP address is allocated to the XAUTH client.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the IP address pool for the client and enter the ISKAMP address pool configuration mode.

crypto isakmp ippool *pool-name*

By default, the address pool is not configured.

(4) Configure the address range of the client address pool.

address *low-ipv4-address high-ipv4-address*

By default, no address range is configured for a client address pool.

1.7.5 Configuring a Client Policy

1. Overview

Configure the policy for the client to establish an IPsec connection with the device. The policy includes the pre-shared key, DNS, IP address, mask, and network permission.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a client policy and enter the ISKAMP client group policy configuration mode.

crypto isakmp client configuration group *name*

By default, no client policy is configured.

(4) Configure a pre-shared key for XAUTH authentication.

key { **0** | **7** } *keystring*

No pre-shared key is configured by default.

(5) Configure the DNS server address to be delivered to the client.

dns *primary-ipv4-addres* [*secondary-ipv4-address*]

No DNS server is configured for a client policy by default.

(6) Configure the mask delivered to the client.

netmask *mask*

No subnet mask is configured for a client policy by default.

(7) Configure the address pool used by the client.

pool *pool-name*

No IP address pool is configured for a client policy by default.

(8) (Optional) Configure the network segment address that the client can access.

network center *ipv4-address/mask-length*

No interesting traffic is configured for a client policy by default. Users can access all network segments.

(9) (Optional) Configure the domain name and VRF instance associated with XAUTH.

domain *domain-name* [**vrf** *vrf-name*]

No domain name and VRF instance are configured for XAUTH by default.

1.7.6 Configuring XAUTH Authentication Mode for a Crypto Map Entry

1. Overview

This function is used to establish an IPsec tunnel between an IPsec client and a device.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the XAUTH identity authentication mode. Select one of the following methods.

- Use AAA authentication.

  crypto map *map-name* **client authentication list** *aaa-name*

  No client authentication is configured by default.

- Use local authentication.

  Create or modify a static crypto map entry.

  crypto map *map-name sequence-number* **ipsec-isakmp**

  Configure the user name and password.

  username *name* **passwd** { **0** | **7** } *password*

No client authentication is configured by default.

(4) (Optional) Configure AAA accounting for XAUTH.

crypto map *map-name* **client accounting list** *aaa-name*

No accounting is performed on clients by default.

1.7.7 Configuring XAUTH Domain Authentication

1. Overview

This function is used to configure XAUTH to use domain name authentication.

2. Configuration Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable domain authentication.

crypto isakmp authorize [**split**]

No domain authentication mode is configured by default.

(4) (Optional) Configure the delimiter option for domain name resolution.

crypto isakmp domain-delimiter *keyword* [**prefix** / **suffix**]

By default, domain name resolution is not used.

1.7.8 Configuring the XAUTH Timeout Period

1. Overview

This function is used to set the XAUTH timeout period. If the network latency is high or the authentication server is slow, increase the timeout period.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the XATUTH timeout period.

crypto isakmp xauth timeout *seconds*

The default XAUTH timeout period is 15 seconds.

1.7.9 Configuring the Timeout Period for Waiting for the AAA Server Response in XAUTH

1. Overview

If the network latency is high or the AAA authentication server is slow, you can increase the timeout period.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the response timeout period of the AAA server.

crypto isakmp xauth server-wait *seconds*

The default AAA server response timeout period is 100 seconds.

1.7.10 Configuring XAUTH to Be Compatible with Cisco Devices

1. Overview

This function is used to configure Cisco-compatible XAUTH for negotiation. This function needs to be configured when the device negotiates with a Cisco device using XAUTH.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure XAUTH to be compatible with Cisco devices.

crypto isakmp xauth cisco_comp

XAUTH compatibility with Cisco is not configured by default.

1.7.11 Configuring the Device Not to Forcibly Use XAUTH for IKE Negotiation

1. Overview

When the device functions as the VPN server and both non-XAUTH clients and XAUTH clients can establish IPsec tunnels with the device in the same crypto map entry, you must configure not forcibly using XAUTH for IKE negotiation.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the device not to forcibly use XAUTH.

crypto isakmp no-force-xauth

By default, XAUTH is forcibly used for all negotiations when XAUTH is configured.

1.8 Configuring Optional Features of IPsec

1.8.1 Configuring the Global IPsec SA Lifetime

1. Overview

After the global IPsec SA lifetime is configured, the default lifetime value of the system is changed. IKE will use this lifetime value for negotiation so that the lifetime of IPsec does not exceed the specified value. Shorter lifetime indicates that less encrypted data of a key can be used by an attacker for analysis and it is more difficult to crack the key. However, when the lifetime is shorter, longer CPU processing time is required for the establishment of a new SA. An SA established through manual configuration has no lifetime.

When IKE negotiates the IPsec lifetime, it takes the smaller of the lifetime values configured on the local and remote peers.

When either the running duration or the total traffic amount reaches a specified threshold, the SA will time out. The negotiation of a new SA starts before an old SA reaches the lifetime limit, to ensure that the new SA is available when the old SA times out. A new SA starts to be negotiated 30 seconds before the lifetime of an old SA expires or when data traffic passing through this tunnel is 256 KB away from the lifetime, whichever occurs first. When an IPsec SA reaches its lifetime, IKE renegotiates a new SA and uses a new set of parameters and keys for the new IPsec SA to make it function properly.

If there is no communication in the lifetime of an SA, the SA will be released and no new SA will be negotiated when the lifetime expires. A new SA will be negotiated only when IPsec identifies packets to be protected.

2. Restrictions and Guidelines

- The configuration is valid only to crypto maps that specify the establishment of IPsec SAs through IKE.
- The IPsec SA lifetime can be globally configured or configured for a specific crypto map.
- Ensure that the time-based lifetime and traffic-based lifetime are not zero at the same time. Otherwise, the negotiation will fail.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the global lifetime for the IPsec SA negotiation.

crypto ipsec security-association lifetime { **seconds** *time* | **kilobytes** *traffic* }

The default lifetime is **3,600** seconds (1 hour) or **4,608,000** KB (communication for 1 hour at a rate of 10 MB per second).

1.8.2 Configuring IPsec SA Lifetime for a Specified Crypto Map

1. Overview

All IPsec SAs use the global lifetime for negotiation by default. If you need to use different lifetime values for SA negotiation based on destination addresses, you can change the lifetime values in the crypto map entries used for negotiation with the destination addresses.

2. Restrictions and Guidelines

- The configuration is valid only to crypto maps that specify the establishment of IPsec SAs through IKE.
- This function only changes the lifetime value in a specified crypto map. It does not affect the global lifetime value.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) **Enter the crypto map configuration mode.** Dynamic crypto map configuration mode **or profile encryption mapping configuration mode.**

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp** [**dynamic** *dynamic-map-name*]

- Entry dynamic crypto map configuration mode

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

- Enter the profile crypto map configuration mode.

  crypto ipsec profile *profile-name*

(4) Configuring the lifetime used for IPsec SA negotiation in the crypto map.

set security-association lifetime { **seconds** *time* | **kilobytes** *traffic* }

IPsec SAs are negotiated based on the default lifetime value according to crypto maps by default.

1.8.3 Configuring the DF Bit Override Function for IPsec Tunnels

1. Overview

You can configure whether fragmentation is allowed for IP packets encapsulated via IPsec.

If the device allows fragmenting IPsec packets, the packet forwarding delay may increase. If the device does not allow fragmenting IPsec packets, when the length of an IPsec packet exceeds the maximum transmission unit (MTU) of an interface, the IPsec packet will be discarded.

Therefore, when you are not sure whether the MTU value of each interface in the forwarding path is greater than the length of IPsec packets, you are advised to configure IPsec packet fragmentation.

2. Restrictions and Guidelines

- This function can be configured only in tunnel mode.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the DF bit override function for IPsec tunnels.

crypto ipsec df-bit { **clear** | **set** | **copy** }

The **clear** option in the outer IP header is set to **0** by default, indicating that fragmentation is allowed.

1.8.4 Disabling IPsec Encapsulation for Multicast and Broadcast Packets

1. Overview

If a crypto ACL contains multicast and broadcast addresses, IPsec encapsulation will be performed on multicast and broadcast packets in this address range by default. If IPsec encapsulation is not required for multicast and broadcast packets, you can configure this function.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Disable IPsec encapsulation for multicast and broadcast packets.

crypto ipsec multicast disable

IPsec encapsulation is enabled for multicast and broadcast packets by default.

1.8.5 Disabling IPsec Check

1. Overview

Data security check is the basic anti-attack function of IPsec. If an IPsec receiver deems that a received packet in plaintext should be encrypted, the packet is insecure and needs to be discarded.

IPsec check consumes many resources. You can disable it.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Disable IPsec check.

crypto ipsec optional

IPsec check is enabled by default.

1.8.6 Specifying the IPsec Local Address

1. Overview

If a crypto map is applied to multiple interfaces and no IPsec local address is specified, for different interfaces having the same remote peer and the same traffic, the device creates an IPsec SA for each interface. The IP address of the outbound interface of encrypted traffic is used as the local address. After the local address is specified, only one IPsec SA is created and the same SA is used for communication no matter whether the same crypto map is applied to several interfaces.

If multiple interfaces on a device support IPsec communication, you can specify the IPsec local address to facilitate management. Then, the device uses this address to communicate with external routers.

2. Restrictions and Guidelines

- Generally, you are advised to use the loopback address as the IPsec local address.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Specify the IPsec local address.

crypto map *map-name* **local-address** *interface-type interface-number*

The IPsec local address is the outbound interface address of IPsec data by default.

1.8.7 Disabling Packet Retransmission Check

1. Overview

Retransmitted packets are processed packets that are received by the device again. After the packet retransmission check is disabled, IPsec no longer checks retransmitted packets, which improves the packet processing efficiency but increases the denial of service (DoS) attack risk.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Disable packet retransmission check.

crypto ipsec security-association replay disable

The packet retransmission check is enabled by default.

1.8.8 Configuring the Matching Rule for Lifetime Negotiation for IPsec Phase 2

1. Overview

The phase 2 lifetime negotiation result takes the lifetime configured on the device in the branch by default. That is, the device in the headquarters (HQ) and the device in the branch use the value configured on the device in the branch as the phase 2 lifetime. You can modify the matching rule for phase 2 lifetime negotiation, that is, the smaller of the lifetime configured on the devices in the HQ and branch is used as final negotiation result.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the matching rule for lifetime negotiation for IPsec phase 2.

crypto ipsec security-association lifetime not_based_on initiator

The phase 2 lifetime negotiation result takes the lifetime configured on the device in the branch by default.

1.8.9 Configuring the MTU for the IPsec Pre-Fragmentation Mode

1. Overview

After fragmentation is configured in tunnel mode, you can configure the size of data fragments prior to encapsulation. Select an appropriate fragment size based on the MTU value of each interface in the network forwarding path.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

- **Enter the crypto map configuration mode.** Dynamic crypto map configuration mode **or profile encryption mapping configuration mode.** Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp** [**dynamic** *dynamic-map-name*]

- Entry dynamic crypto map configuration mode

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

- Enter the profile crypto map configuration mode.

  crypto ipsec profile *profile-name*

(3) Configure the MTU for the IPsec pre-fragmentation mode.

set mtu *length*

No MTU is configured for the IPsec pre-fragmentation mode by default.

1.8.10 Configuring RRI

1. Overview

After the reverse route injection (RRI) function is configured and the negotiation of a tunnel is complete, the IPsec module automatically adds a static route pointing to the peer end of the tunnel or to a specified IP address.

In the large-scale HQ-branch networking, this function can reduce the workload of static route configuration on the device in the HQ and dynamically add or automatically delete static routes.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) **Entry crypto map configuration mode or dynamic crypto map configuration mode.**

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp**

- Enter the crypto map configuration mode.

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

(4) Configure the reverse IPv4 route injection function.

reverse-route [**remote-peer** *ipv4-address*] [*distance* | **tag** tag-number | **track** track-number | **weight** weight-number] *

By default, the reverse IPv4 route injection function is not configured.

(5) Configure the reverse IPv6 route injection function.

reverse-ipv6-route [**remote-peer** *ipv6-address*] [*distance* | **weight** *weight-number*] *

By default, the reverse IPv6 route injection function is not configured.

1.8.11 Configuring the Diffie-Hellman Group Identifier for IPsec Tunnel Encapsulation

1. Overview

Configure the Diffie-Hellman group identifier for IPsec tunnel encapsulation as required. Group 1, group 2, and group 5 are the 768-bit, 1024-bit, and 1536-bit Diffie-Hellman groups respectively. The security and required computation time of these groups increase in sequence.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) **Entry crypto map configuration mode, dynamic crypto map configuration mode, or profile crypto map configuration mode**

- Enter the crypto map configuration mode.

crypto map *map-name sequence-number* **ipsec-isakmp** [**dynamic** *dynamic-map-name*]

- Entry dynamic crypto map configuration mode

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

- Enter the profile crypto map configuration mode.

  crypto ipsec profile *profile-name*

(4) Configure the Diffie-Hellman group identifier for IPsec tunnel encapsulation.

set pfs *group*

No Diffie-Hellman group identifier is used for IPsec tunnel encapsulation by default.

1.8.12 Setting the Work Mode to Tunnel Autoup

1. Overview

Setting the work mode to tunnel autoup can prevent packet loss caused by tunnel negotiation. Use this function in scenarios in which data transmission is sensitive and a tunnel needs to be up any time.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the crypto map configuration mode or dynamic crypto map configuration mode.

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp**

- Enter the dynamic crypto map configuration mode

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

(4) Set the work mode to tunnel autoup.

set autoup

Tunnel autoup is disabled by default.

1.8.13 Specifying the Local IP Address in a Crypto Map Entry

1. Overview

This command is used to configure an IP address used for local negotiation. If no local IP address is configured, the master address of an interface is used. The specified local IP address will be used after configuration.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the crypto map configuration mode or dynamic crypto map configuration mode.

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp**

- Enter the dynamic crypto map configuration mode

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

(4) Specify the local IP address for the crypto map entry.

set local *ipv4-address*

No local IP address is specified for a crypto map entry by default.

1.8.14 Binding a Track Monitoring Event to a Crypto Map Entry

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the crypto map configuration mode.

crypto map *map-name sequence-number* **ipsec-isakmp** [**dynamic** *dynamic-map-name*]

(4) Bind a track monitoring event to a crypto map entry.

set track *track-id* [{ **up** | **down** } { **clear** | **negotiate** }]

No track monitoring event is bound to a crypto map entry by default.

1.8.15 Configuring Packet Matching VRF Before Encryption in a Specified Crypto Map

1. Overview

On an MPLS L3VPN network, if the data of the transmitter needs to be encrypted by IPsec, only the packets of this VRF and matching the specified access control list will be encrypted.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) **Enter the crypto map configuration mode or dynamic crypto map configuration mode.**

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp**

- Enter the dynamic crypto map configuration mode.

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

(4) Configure packet matching VRF.

match vrf *vrf-name*

No packet matching VRF is configured by default.

1.8.16 Configuring the VRF to Which Decrypted Packets Belong After the Specified Crypto Map Is Configured

1. Overview

In the VPE environment, the extranet interface and the intranet interface belong to different VRFs. To switch packets from one VRF to another VRF, the VRF of the packet needs to be set after the packet is decapsulated.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the crypto map configuration mode, dynamic crypto map configuration mode, or profile crypto map configuration mode.

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp**

- Enter the dynamic crypto map configuration mode.

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

- Enter the profile crypto map configuration mode.

  crypto ipsec profile *profile-name*

(4) Configure the VRF instance to which the decrypted packets belong.

set vrf *vrf-name*

No VRF instance to which decrypted packets belong is configured by default.

1.8.17 Configuring the Negotiation Mode of a Specified Crypto Map

1. Overview

The IKE negotiation includes two phases:

- In phase 1, a secure channel that passes authentication is established between two ISAKMP entities. The main mode or aggressive mode can be adopted in this phase.
- In phase 2, service SAs are negotiated.

The main mode is adopted in phase 1 by default. When IP addresses are not statically configured, the aggressive mode can be used.

2. Configuration Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the crypto map configuration mode, dynamic crypto map configuration mode, or profile crypto map configuration mode.

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp**

- Enter the dynamic crypto map configuration mode.

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

- Enter the profile crypto map configuration mode.

  crypto ipsec profile *profile-name*

(4) Configure the negotiation mode.

set exchange-mode { **main** | **aggressive** }

The default work mode for phase 1 of IKE negotiation between peers is the main mode.

1.8.18 Disabling Packet Filtering After Decryption

1. Overview

If the original packet after IPsec decapsulation does not need to be filtered, you can run this command to disable packet filtering after decryption.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Disable filtering of decrypted packets.

crypto ipsec no-filter [**list** [*acl-name* | *acl-number]*]

Post-decryption packet filtering is enabled by default.

1.8.19 Configuring Automatic Disconnection of Idle IPsec Tunnels Globally

1. Overview

If no traffic is transmitted over an IPsec tunnel, the tunnel connection is still maintained, which wastes system resources. The automatic disconnection of idle IPsec tunnels can be configured globally or for a specified crypto map. The configuration for a specified crypto map has a higher priority.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure automatic disconnection of idle IPsec tunnels globally.

crypto ipsec security-association idle-time *sec* [**inbound** | **outbound**]

The automatic disconnection of idle IPsec tunnels is disabled by default.

1.8.20 Configuring Automatic Disconnection of Idle IPsec Tunnels with a Specified Crypto Map

1. Overview

If no traffic is transmitted over an IPsec tunnel, the tunnel connection is still maintained, which wastes system resources. The automatic disconnection of idle IPsec tunnels can be configured globally or for a specified crypto map. The configuration for a specified crypto map has a higher priority.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the crypto map configuration mode, dynamic crypto map configuration mode, or profile crypto map configuration mode

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp**

- Enter the dynamic crypto map configuration mode.

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

- Enter the profile crypto map configuration mode.

  crypto ipsec profile *profile-name*

(4) Configure automatic IPsec tunnel disconnection when the tunnel is idle.

set security-association idle-time *sec* [**inbound** | **outbound**]

By default, automatic disconnection of an idle IPsec tunnel is not configured. If no traffic statistics direction is configured, the default bidirectional traffic statistics collection is used.

1.8.21 Configuring the Bypass Function for IPsec Tunnels Globally

1. Overview

Configure the bypass function for IPsec tunnels globally. When the IPsec tunnel is unavailable, IPsec tunnel packets that need to be encrypted are not discarded.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable the bypass function for global IPsec tunnels.

crypto ipsec tunnel-bypass enable

By default, the bypass mode is disabled for a tunnel. That is, when an IPsec tunnel becomes unavailable, IPsec discards the tunnel packets that need to be encrypted.

(4) Enter the crypto map configuration mode, dynamic crypto map configuration mode, or profile crypto map configuration mode.

- Enter the crypto map configuration mode.

  crypto map *map-name sequence-number* **ipsec-isakmp**

- Enter the dynamic crypto map configuration mode.

  crypto dynamic-map *dynamic-map-name dynamic-sequence-number*

- Enter the profile crypto map configuration mode.

  crypto ipsec profile *profile-name*

(5) Configure IPsec tunnel bypass.

set tunnel bypass

By default, the IPsec tunnel bypass function is not configured in a crypto map entry.

1.8.22 Configuring the Global IPsec MIB Function

1. Overview

IPsec MIB management involves statistics on data flows and encrypted and decrypted data packets, which may affect the performance of IPsec data communication. Therefore, the IPsec MIB statistics function is disabled by default. To access IPsec MIB nodes, you need to run this command to enable the IPsec MIB function.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable the IPsec MIB.

crypto mib enable

The IPsec MIB function is disabled by default.

(4) (Optional) Configure the interval for updating IPsec MIB information.

crypto mib collect-update *seconds*

By default, the interval for updating IPsec MIB information is 2 seconds.

1.8.23 Configuring Interesting Traffic with a Wildcard Mask of All Zeros

1. Overview

In IPv6, IPsec-IPv4, and IPsec-IPv6 tunnels, you need to configure the **match any** command in the crypto map set, that is, the command specifies the local IP address/mask (0.0.0.0/0.0.0.0) and the peer IP address/mask (0.0.0.0/0.0.0.0) of the interesting traffic. The profile map configured with the **match any** command can be used only for IPv4 over IPv4 (IPIP) and IPv6 tunnels.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the profile crypto map configuration mode.

crypto ipsec profile *profile-name*

(4) Configure interesting traffic with a wildcard mask of all zeros.

match any

By default, interesting traffic with the local IP address/mask (0.0.0.0/0.0.0.0) and the peer IP address/mask (0.0.0.0/0.0.0.0) is not configured.

1.9 Monitoring

Run the **show** command to check the running status of a configured function to verify the configuration effect.

Run the **clear** command to clear information.

Caution

- The output debugging information occupies system resources. Therefore, disable the debugging function immediately after use.
- Running the **clear** command may lose vital information and thus interrupt services.

Table 1-3 IPsec Monitoring

Command	Purpose
show crypto autoup	Displays information about the automatic IPsec tunnel connection function.
show crypto data	Displays the application layer statistics of an IPsec tunnel.
show crypto detail [*tmpmap-id* | **interface** *interface-type interface-number* | **map** *map-name* [*map-sequence*] | **other** | **profile** [**interface** *interface-type interface-number*]]	Displays detailed information about an IPsec tunnel.
show crypto dynamic-map [*map-name*]	Displays information about a dynamic crypto map.
show crypto ipsec port	Displays IPsec-related port information.
show crypto ipsec sa [*tmpmap-id* | **interface** *interface-type interface-number* | **ipv6-peer** *ipv6-peer-address* | **peer** *peer-address*]	Displays information about an IPsec SA.
show crypto ipsec transform-set	Displays the configuration of a transform set.
show crypto kernel sab *sab-id*	Displays forwarding plane information entries.
show crypto map [*map-name*]	Displays the configuration of all or a specified crypto map.

Command	Purpose
show crypto state [*state-id* | **ipv6-peer** *peer-name* | **peer** *peer-ip* | **specific** *source-ipv4-address source-mask destination-ipv4-address destination-mask* | **web**]	Displays the status of an IPsec tunnel.
show crypto timer	Displays timer information of an IPsec tunnel.
show ipsec manual sa [*sa-name*]	Displays information about a configured SA.
show ipsec proposal [*proposal-name*]	Displays information about a configured IPsec security proposal.
clear crypto sa	Clears the entire SA database. All active security threads will be also deleted after this command is executed.
clear crypto sa peer { *ipv4-address* | *peer-name* }	Clears the SA with a specific peer address.
clear crypto sa map *map-name*	Clears SAs in a specific crypto map.
clear crypto sa spi *destionation-address* { **ah** | **esp** } *spi*	Clears the SA with a specified <destination address, protocol, SPI>.

1.10 Configuration Examples

1.10.1 Configuring IPsec VPN

1. Requirements

An IPsec tunnel is established between Device A and Device B, and data flows between them are protected through IPsec.

2. Topology

Figure 1-3 Topology of IPsec VPN

3. Notes

- Configure Device A and Device B to ensure the route between them is reachable. (The details are omitted.)
- Configure IKE or configure SA parameters manually to implement key exchange.
- Configure an IPsec VPN tunnel to protect communication.

4. Procedure (IKE Negotiation)

(1) Configure Device A.

Enable IKE.

```
DeviceA> enable
DeviceA# configure terminal
DeviceA(config)# crypto isakmp enable
DeviceA(config)# crypto isakmp policy 1
DeviceA(isakmp-policy)# authentication pre-share
DeviceA(isakmp-policy)# encryption 3des
DeviceA(isakmp-policy)# exit
```

Define a crypto ACL to protect the IP communication between 1.1.1.1/32 and 1.1.2.1/32.

```
DeviceA(config)# access-list 101 permit ip 1.1.1.1 0.0.0.0 1.1.2.1 0.0.0.0
```

Configure a pre-shared key and transform set.

```
DeviceA(config)# crypto isakmp key 0 preword address 1.1.2.1
DeviceA(config)# crypto ipsec transform-set myset esp-des esp-md5-hmac
```

Define a crypto map.

```
DeviceA(config)# crypto map mymap 5 ipsec-isakmp
DeviceA(config-crypto-map)# set peer 1.1.2.1
DeviceA(config-crypto-map)# set transform-set myset
DeviceA(config-crypto-map)# match address 101
DeviceA(config-crypto-map)# exit
```

Apply the crypto map to an interface.

```
DeviceA(config)# interface gigabitethernet 0/1
DeviceA(config-if-GigabitEthernet 0/1)# ip address 1.1.1.1 255.255.255.0
DeviceA(config-if-GigabitEthernet 0/1)# crypto map mymap
```

(2) Configure Device B.

Enable IKE.

```
DeviceB> enable
DeviceB# configure terminal
DeviceB(config)# crypto isakmp enable
DeviceB(config)# crypto isakmp policy 1
DeviceB(isakmp-policy)# authentication pre-share
DeviceB(isakmp-policy)# encryption 3des
DeviceB(isakmp-policy)# exit
```

Define a crypto ACL to protect the IP communication between 1.1.2.1/32 and 1.1.1.1/32.

```
DeviceB(config)# access-list 101 permit ip 1.1.2.1 0.0.0.0 1.1.1.1 0.0.0.0
```

Configure a pre-shared key and transform set.

```
DeviceB(config)# crypto isakmp key 0 preword address 1.1.1.1
DeviceB(config)# crypto ipsec transform-set myset esp-des esp-md5-hmac
```

Define a crypto map.

```
DeviceB(config)# crypto map mymap 5 ipsec-isakmp
```

```
DeviceB(config-crypto-map)# set peer 1.1.1.1
DeviceB(config-crypto-map)# set transform-set myset
DeviceB(config-crypto-map)# match address 101
DeviceB(config-crypto-map)# exit
```

Apply the crypto map to an interface.

```
DeviceB(config)# interface gigabitethernet 0/1
DeviceB(config-if-GigabitEthernet 0/1)# ip address 1.1.2.1 255.255.255.0
DeviceB(config-if-GigabitEthernet 0/1)# crypto map mymap
```

5. Procedure (Manual Configuration for SA Establishment)

(1) Configure Device A.

Define a transform set.

```
DeviceA> enable
DeviceA# configure terminal
DeviceA(config)# crypto ipsec transform-set myset esp-des esp-md5-hmac
DeviceA(cfg-crypto-trans)# exit
```

Define a crypto ACL to protect the IP communication between 1.1.1.1/32 and 1.1.2.1/32.

```
DeviceA(config)# access-list 101 permit ip 1.1.1.1 0.0.0.0 1.1.2.1 0.0.0.0
```

Define a crypto map.

```
DeviceA(config)# crypto map mymap 5 ipsec-manual
DeviceA(config-crypto-map)# set peer 1.1.2.1
DeviceA(config-crypto-map)# set session-key inbound esp 300 cipher
abcdef1234567890 authenticator abcdef1234567890abcdef1234567890
DeviceA(config-crypto-map)# set session-key outbound esp 301 cipher
abcdef1234567890 authenticator abcdef1234567890abcdef1234567890
DeviceA(config-crypto-map)# set transform-set myset
DeviceA(config-crypto-map)# match address 101
DeviceA(config-crypto-map)# exit
```

Apply the crypto map to an interface.

```
DeviceA(config)# interface gigabitethernet 0/1
DeviceA(config-if-GigabitEthernet 0/1)# ip address 1.1.1.1 255.255.255.0
DeviceA(config-if-GigabitEthernet 0/1)# crypto map mymap
```

(2) Configure Device B.

Define a transform set.

```
DeviceB> enable
DeviceB# configure terminal
DeviceB(config)# crypto ipsec transform-set myset esp-des esp-md5-hmac
DeviceB(cfg-crypto-trans)# exit
```

Define a crypto ACL to protect the IP communication between 1.1.2.1/32 and 1.1.1.1/32.

```
DeviceB(config)# access-list 101 permit ip 1.1.2.1 0.0.0.0 1.1.1.1 0.0.0.0
```

Define a crypto map.

```
DeviceB(config)# crypto map mymap 5 ipsec-manual
```

```
DeviceB(config-crypto-map)# set peer 1.1.1.1
DeviceB(config-crypto-map)# set session-key inbound esp 301 cipher
abcdef1234567890 authenticator abcdef1234567890abcdef1234567890
DeviceB(config-crypto-map)# set session-key outbound esp 300 cipher
abcdef1234567890 authenticator abcdef1234567890abcdef1234567890
DeviceB(config-crypto-map)# set transform-set myset
DeviceB(config-crypto-map)# match address 101
DeviceB(config-crypto-map)# exit
```

Apply the crypto map to an interface.

```
DeviceB(config)# interface gigabitethernet 0/1
DeviceB(config-if-GigabitEthernet 0/1)# ip address 1.1.2.1 255.255.255.0
DeviceB(config-if-GigabitEthernet 0/1)# crypto map mymap
```

6. Verification

Check whether IKE SAs are established.

```
DeviceA# show crypto isakmp sa
destination     source          state           conn-id     lifetime(second)
1.1.2.1         1.1.1.1         IKE_IDLE          1           84518
```

Check whether IPsec SAs are established.

```
DeviceA# show crypto ipsec sa
    Crypto map tag:mymap
local ipv4 addr 1.1.1.1
media mtu 1500


    ==================================
    sub_map type:static, seqno:5, id=1
local   ident (addr/mask/prot/port): (1.1.1.1/0.0.0.255/0/0))
remote  ident (addr/mask/prot/port): (1.1.2.1/0.0.0.255/0/0))
PERMIT
#pkts encaps: 3, #pkts encrypt: 3, #pkts digest 3
#pkts decaps: 3, #pkts decrypt: 3, #pkts verify 3
#send errors 0, #recv errors 0
pkts encaps errors:
    #negotiate pkt drop: 0, #sab useless: 0, encap data fail: 0, compute hash fail: 0
pkts decrypt errors:
    #check reply wind fail: 0, #compute hash fail: 0, verify hash fail: 0
#pkts detect send req: 0, recv reply: 0, recv req: 0, send reply: 0


        Inbound esp sas:
            spi:0x9a7dd3fa (2591937530)
```

```
                transform: esp-des esp-md5-hmac
                in use settings={Tunnel Encaps,}
                crypto map mymap 5
                sa timing: remaining key lifetime (k/sec): (4606998/1427)
                IV size: 0 bytes
                Replay detection support:Y



        Outbound esp sas:
              spi:0x8997060e (2308376078)
                transform: esp-des esp-md5-hmac
                in use settings={Tunnel Encaps,}
                crypto map mymap 5
                sa timing: remaining key lifetime (k/sec): (4606998/1427)
                IV size: 0 bytes
                Replay detection support:Y
```

Check the SA established through manual configuration.

```
DeviceA# show crypto ipsec sa
    Crypto map tag:mymap
local ipv4 addr 1.1.1.1
media mtu 1500


    ================================
    sub_map type:static, seqno:5, id=1
local    ident (addr/mask/prot/port): (1.1.1.1/0.0.0.255/0/0))
remote   ident (addr/mask/prot/port): (1.1.2.1/0.0.0.255/0/0))
PERMIT
#pkts encaps: 3, #pkts encrypt: 3, #pkts digest 3
#pkts decaps: 3, #pkts decrypt: 3, #pkts verify 3
#send errors 0, #recv errors 0
pkts encaps errors:
    #negotiate pkt drop: 0, #sab useless: 0, encap data fail: 0, compute hash fail: 0
pkts decrypt errors:
    #check reply wind fail: 0, #compute hash fail: 0, verify hash fail: 0
#pkts detect send req: 0, recv reply: 0, recv req: 0, send reply: 0


        Inbound esp sas:
```

```
            spi:0x12c (300)
              transform: esp-des esp-md5-hmac
              in use settings={Tunnel Encaps,}
              crypto map mymap 5
              no sa timing
              IV size: 8 bytes
              Replay detection support:N


      Outbound esp sas:
            spi:0x12d (301)
              transform: esp-des esp-md5-hmac
              in use settings={Tunnel Encaps,}
              crypto map mymap 5
              no sa timing
              IV size: 8 bytes
              Replay detection support:N
```

7. Configuration Files

- Configuration files for IKE negotiation

  Device A configuration file

```
hostname DeviceA
!
ip access-list extended 101
 10 permit ip 1.1.1.1 0.0.0.0 1.1.2.1 0.0.0.0
!
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
!
crypto isakmp key 7 155a1f2405243e01 address 1.1.2.1
crypto ipsec transform-set myset esp-des esp-md5-hmac
!
crypto map mymap 5 ipsec-isakmp
 set peer 1.1.2.1
 set transform-set myset
 match address 101
!
interface GigabitEthernet 0/1
 ip address 1.1.1.1 255.255.255.0
 crypto map mymap
!
```

```
End
```

Device B configuration file

```
hostname DeviceB
!
ip access-list extended 101
 10 permit ip 1.1.2.1 0.0.0.0 1.1.1.1 0.0.0.0
!
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
!
crypto isakmp key 7 155a1f2405243e01 address 1.1.1.1
crypto ipsec transform-set myset esp-des esp-md5-hmac
!
crypto map mymap 5 ipsec-isakmp
 set peer 1.1.1.1
 set transform-set myset
 match address 101
!
interface GigabitEthernet 0/1
 ip address 1.1.2.1 255.255.255.0
 crypto map mymap
!
End
```

- Configuration files for manual configuration

  Device A configuration file

```
hostname DeviceA
!
ip access-list extended 101
 10 permit ip 1.1.1.1 0.0.0.0 1.1.2.1 0.0.0.0
!
crypto ipsec transform-set myset esp-des esp-md5-hmac
!
crypto map mymap 5 ipsec-manual
 set peer 1.1.2.1
set session-key inbound esp 300 cipher abcdef1234567890 authenticator
abcdef1234567890abcdef1234567890
 set session-key outbound esp 301 cipher abcdef1234567890 authenticator
abcdef1234567890abcdef1234567890
set transform-set myset
 match address 101
!
interface GigabitEthernet 0/1
 ip address 1.1.1.1 255.255.255.0
 crypto map mymap
```

```
!
end
```

Device B configuration file

```
hostname DeviceB
!
ip access-list extended 101
 10 permit ip 1.1.2.1 0.0.0.0 1.1.1.1 0.0.0.0
!
crypto ipsec transform-set myset esp-des esp-md5-hmac
!
crypto map mymap 5 ipsec-manual
 set peer 1.1.1.1
set session-key inbound esp 301 cipher abcdef1234567890 authenticator
abcdef1234567890abcdef1234567890
 set session-key outbound esp 300 cipher abcdef1234567890 authenticator
abcdef1234567890abcdef1234567890
set transform-set myset
 match address 101
!
interface GigabitEthernet 0/1
 ip address 1.1.2.1 255.255.255.0
 crypto map mymap
!
end
```

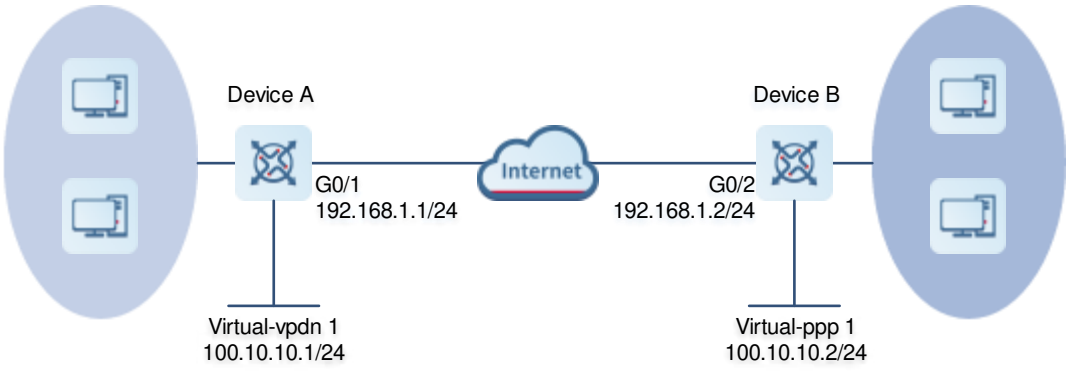
1.10.2 Configuring L2TP over IPsec Encryption

1. Requirements

The customer wants to connect to a remote private network, users on which access the network through ADSL dial-up with dynamic IP addresses.

2. Topology

Figure 1-4 Configuring L2TP over IPsec



3. Notes

- Configure an IKE policy and a pre-shared key.
- Configure a transform set used by an IPsec SA.
- Configure a dynamic crypto map entry on the L2TP server (Device A).
- Create a crypto map on the L2TP server (Device A) and apply it to G 0/1.
- Create a profile crypto map on the L2TP client (Device B) and apply it to Virtual-ppp 1.
- Establish an L2TP tunnel between Device A and Device B.

4. Procedure

(1) Configure Device A.

Configure an IP address for the interface.

```
DeviceA> enable
DeviceA# configure terminal
DeviceA(config)# interface gigabitethernet 0/1
DeviceA(config-if-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
DeviceA(config-if-GigabitEthernet 0/1)# exit
```

Configure an IKE policy and a pre-shared key.

```
DeviceA(config)# crypto isakmp policy 10
DeviceA(isakmp-policy)# encryption 3des
DeviceA(isakmp-policy)# authentication pre-share
DeviceA(isakmp-policy)# exit
DeviceA(config)# crypto isakmp key 0 policy address 192.168.1.2
```

Configure a transform set used by an IPsec SA.

```
DeviceA(config)# crypto ipsec transform-set vpdnSet esp-des esp-sha-hmac
DeviceA(cfg-crypto-trans)# mode tunnel
DeviceA(cfg-crypto-trans)# exit
```

Configure a dynamic crypto map entry and apply it to the Layer 3 Ethernet interface GigabitEthernet 0/1.

```
DeviceA(config)# crypto dynamic-map dymymap 6
DeviceA(config-crypto-map)# set security-association lifetime seconds 3600
DeviceA(config-crypto-map)# set transform-set vpdnSet
DeviceA(config-crypto-map)# set mtu 1380
DeviceA(config-crypto-map)# exit
DeviceA(config)# crypto map l2tpmap 100 ipsec-isakmp dynamic dymymap
DeviceA(config)# interface gigabitethernet 0/1
DeviceA(config-if-GigabitEthernet 0/1)# crypto map l2tpmap
DeviceA(config-if-GigabitEthernet 0/1)# exit
```

Configure the L2TP server.

```
DeviceA(config)# username admin123 password pass1234
DeviceA(config)# vpdn enable
DeviceA(config)# ip local pool l2tppool 100.10.10.2 100.10.10.254
DeviceA(config)# vpdn-group l2tp
DeviceA(config-vpdn)# accept-dialin
```

```
DeviceA(config-vpdn-acc-in) # protocol l2tp
DeviceA(config-vpdn-acc-in) # virtual-vpdn 1
DeviceA(config-vpdn-acc-in) # exit
DeviceA(config-vpdn)# l2tp tunnel force_ipsec
DeviceA(config-vpdn)# exit
DeviceA(config)# interface virtual-vpdn 1
DeviceA(config-if-Virtual-vpdn 1)# ip tcp adjust-mss 1368
DeviceA(config-if-Virtual-vpdn 1)# ip mtu 1408
DeviceA(config-if-Virtual-vpdn 1)# ip address 100.10.10.1 255.255.255.0
DeviceA(config-if-Virtual-vpdn 1)# peer default ip address pool l2tppool
DeviceA(config-if-Virtual-vpdn 1)# ppp authentication chap
DeviceA(config-if-Virtual-vpdn 1)# exit
```

(2) Configure Device B.

Configure an IP address for the interface.

```
DeviceB> enable
DeviceB# configure terminal
DeviceB(config)# interface gigabitEthernet 0/2
DeviceB(config-if-GigabitEthernet 0/2)# ip address 192.168.1.2 255.255.255.0
```

Configure an IKE policy and a pre-shared key.

```
DeviceB(config)# crypto isakmp policy 10
DeviceB(isakmp-policy)# encryption 3des
DeviceB(isakmp-policy)# authentication pre-share
DeviceB(isakmp-policy)# exit
DeviceB(config)# crypto isakmp key 0 policy address 192.168.1.1
```

Configure a transform set used by an IPsec SA.

```
DeviceB(config)# crypto ipsec transform-set sl_set_1 esp-des esp-sha-hmac
DeviceB(cfg-crypto-trans)# mode tunnel
DeviceB(cfg-crypto-trans)# exit
```

Configure and apply the profile.

```
DeviceB(config)# crypto ipsec profile s_l2tpMap_1
DeviceB(config-crypto-profile)# set transform-set sl_set_1
DeviceB(config-crypto-profile)# exit
DeviceB(config)# interface virtual-ppp 1
DeviceB(config-if-Virtual-ppp 1)# tunnel protection ipsec profile s_l2tpMap_1
```

Configure the L2TP client.

```
DeviceB(config-if-Virtual-ppp 1)# ip address negotiate
DeviceB(config-if-Virtual-ppp 1)# ppp chap hostname admin123
DeviceB(config-if-Virtual-ppp 1)# ppp chap password pass1234
DeviceB(config-if-Virtual-ppp 1)# pseudowire 192.168.1.1 1 encapsulation
l2tpv2
```

5. Verification

Check whether an L2TP over IPsec tunnel is established on Device A.

```
DeviceA# show crypto ipsec sa
      Crypto map tag:l2tpmap
local ipv4 addr 192.168.1.1
media mtu 1500

       ================================
       sub_map type:temporary, seqno:6, id=2
local     ident (addr/mask/prot/port): (192.168.1.1/0.0.0.0/17/1701))
remote    ident (addr/mask/prot/port): (192.168.1.2/0.0.0.0/17/1701))
PERMIT
#pkts encaps: 15, #pkts encrypt: 15, #pkts digest 15
#pkts decaps: 14, #pkts decrypt: 14, #pkts verify 14
#send errors 0, #recv errors 0
pkts encaps errors:
      #negotiate pkt drop: 0, #sab useless: 0, encap data fail: 0, compute hash fail: 0
pkts decrypt errors:
      #check reply wind fail: 0, #compute hash fail: 0, verify hash fail: 0
#pkts detect send req: 0, recv reply: 0, recv req: 0, send reply: 0

            Inbound esp sas:
                  spi:0xecd3b645 (3973297733)
                   transform: esp-des esp-sha-hmac
                   in use settings={Tunnel Encaps,}
                   crypto map dymymap 6
                   sa timing: remaining key lifetime (k/sec): (4607996/3574)
                   IV size: 0 bytes
                   Replay detection support:Y



            Outbound esp sas:
                  spi:0x26ad054f (648873295)
                   transform: esp-des esp-sha-hmac
                   in use settings={Tunnel Encaps,}
                   crypto map dymymap 6
                   sa timing: remaining key lifetime (k/sec): (4607996/3574)
                   IV size: 0 bytes
                   Replay detection support:Y
```

Check the VPDN tunnel.

```
DeviceA# show vpdn tunnel
L2TP Tunnel Information Total tunnels 1
LocID RemID Remote Name    State    Remote Address    Port    Sessions L2TP Class/
                                                                      VPDN Group
4       3       siteA           est      192.168.1.2       1701    1           l2tp
%No active PPTP tunnels
```

Check the VPDN session.

```
DeviceA# show vpdn session
L2TP Session Information Total sessions 1
LocID RemID TunID Username, Intf/                                State     Last Chg
1       1       4       admin123,va4                                 est       00:04:13
%No active PPTP tunnels
```

Check whether the L2TP over IPsec tunnel on Device A is reachable.

```
DeviceA# show ip route


Codes:    C - Connected, L - Local, S - Static
              R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
              N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
              E1 - OSPF external type 1, E2 - OSPF external type 2
              SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
              IA - Inter area, EV - BGP EVPN, A - Arp to host
              LA - Local aggregate route
              * - candidate default


Gateway of last resort is no set
C        100.10.10.0/24 is directly connected, Loopback 1, 05:05:36
C        100.10.10.1/32 is is directly connected, Loopback 1, 05:05:36
C        100.10.10.2/32 is directly connected, virtual-access 4, 03:43:40
C        192.168.1.0/24 is directly connected, GigabitEthernet 0/1, 04:40:40
C        192.168.1.1/32 is directly connected, GigabitEthernet 0/1, 04:40:40
DeviceA# ping 100.10.10.2
Sending 5, 100-byte ICMP Echoes to 100.10.10.2, timeout is 2 seconds:
   < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms.
```

Check whether the L2TP over IPsec tunnel on Device B is reachable.

```
DeviceB# show ip route
```

```
Codes:  C - Connected, L - Local, S - Static
        R - RIP, O - OSPF, B - BGP, I - IS-IS, V - Overflow route
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        SU - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        IA - Inter area, EV - BGP EVPN, A - Arp to host
        LA - Local aggregate route
        * - candidate default

Gateway of last resort is no set
C       100.10.10.1/32 is directly connected, Virtual-ppp 1, 05:05:36
C       100.10.10.2/32 is directly connected, Virtual-ppp 1, 05:05:36
C       192.168.1.0/24 is directly connected, GigabitEthernet 0/2, 06:05:36
C       192.168.1.2/32 is directly connected, GigabitEthernet 0/2, 06:05:36
DeviceB# ping 100.10.10.1
Sending 5, 100-byte ICMP Echoes to 100.10.10.1, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms.
```

6. Configuration Files

- Device A configuration file

```
hostname DeviceA
!
username admin123 password pass1234
!
ip local pool l2tppool 100.10.10.2 100.10.10.254
!
crypto isakmp policy 10
 encryption 3des
 authentication pre-share
!
crypto isakmp key 0 policy address 192.168.1.2
crypto ipsec transform-set vpdnSet esp-des esp-sha-hmac
!
crypto dynamic-map dymymap 6
 set security-association lifetime seconds 3600
 set transform-set vpdnSet
 set mtu 1380
!
```

```
crypto map l2tpmap 100 ipsec-isakmp dynamic dymymap
vpdn enable
!
vpdn-group l2tp
! Default L2TP VPDN group
 accept-dialin
  protocol l2tp
  virtual-vpdn 1
 l2tp tunnel force_ipsec
!
interface GigabitEthernet 0/1
 crypto map l2tpmap
 ip address 192.168.1.1 255.255.255.0
!
interface Virtual-vpdn 1
 ppp authentication chap
 ip tcp adjust-mss 1368
 ip mtu 1408
 ip address 100.10.10.1 255.255.255.0
 peer default ip address pool l2tppool
!
end
```

- Device B configuration file

```
hostname DeviceB
!
crypto isakmp policy 10
 encryption 3des
 authentication pre-share
!
crypto isakmp key 0 policy address 192.168.1.1
crypto ipsec transform-set sl_set_1 esp-des esp-sha-hmac
!
crypto ipsec profile s_l2tpMap_1
 set transform-set sl_set_1
!
interface GigabitEthernet 0/2
ip address 192.168.1.2 255.255.255.0
!
interface Virtual-ppp 1
 ppp chap hostname admin123
 ppp chap password pass1234
 ip address negotiate
 tunnel protection ipsec profile s_l2tpMap_1
 pseudowire 192.168.1.1 1 encapsulation l2tpv2
!
end
```

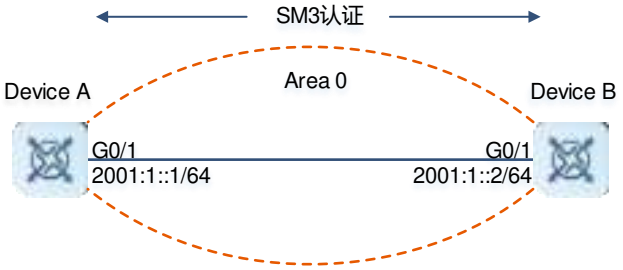
1.10.3 Configuring an IPsec Tunnel for Routing Protocol Authentication

1. Requirements

The customer wants to use SM3 authentication for OSPFv3.

2. Topology

Figure 1-5 Configuring an IPsec Tunnel for Routing Protocol Authentication



3. Notes

- Enable IPv6 on all device interfaces.
- Configure basic OSPFv3 features on all devices.
- Configure OSPFv3 IPsec authentication on all device interfaces.

4. Procedure

(1) Configure Device A.

Start an OSPFv3 process and configure a router ID.

```
DeviceA> enable
DeviceA# configure terminal
DeviceA(config)# ipv6 router ospf 1
DeviceA(config-router)# router-id 1.1.1.1
DeviceA(config-router)# exit
```

Configure an IPsec proposal.

```
DeviceA(config)# ipsec proposal proposal111
DeviceA(config-ipsec-proposal)# encapsulation-mode transport
DeviceA(config-ipsec-proposal)# transform ah
DeviceA(config-ipsec-proposal)# ah authentication-algorithm sm3
DeviceA(config-ipsec-proposal)# exit
```

Configure an IPsec SA.

```
DeviceA(config)# ipsec sa sa1
DeviceA(config-ipsec-sa)# proposal proposal111
DeviceA(config-ipsec-sa)# sa spi ah 256
DeviceA(config-ipsec-sa)# sa authentication-hex ah
0123456789012345678901234567890123456789012345678901234567890123456789abcd
DeviceA(config-ipsec-sa1)# exit
```

Configure an IP address for an interface and configure OSPFv3 on the interface.

```
DeviceA(config)# interface gigabitethernet 0/1
DeviceA(config-if-GigabitEthernet 0/1) ipv6 enable
DeviceA(config-if-GigabitEthernet 0/1) ipv6 address 2001:1::1/64
DeviceA(config-if-GigabitEthernet 0/1) ipv6 ospf 1 area 0
```

Apply the IPsec SA to the routing protocol.

```
DeviceA(config-if-GigabitEthernet 0/1)# ipv6 ospf ipsec sa sa1
```

(2) Configure Device B.

Start an OSPFv3 process and configure a router ID.

```
DeviceB> enable
DeviceB# configure terminal
DeviceB(config)# ipv6 router ospf 1
DeviceB(config-router)# router-id 2.2.2.2
DeviceB(config-router)# exit
```

Configure an IPsec proposal.

```
DeviceB> enable
DeviceB# configure terminal
DeviceB(config)# ipsec proposal proposal111
DeviceB(config-ipsec-proposal)# encapsulation-mode transport
DeviceB(config-ipsec-proposal)# transform ah
DeviceB(config-ipsec-proposal)# ah authentication-algorithm sm3
DeviceB(config-ipsec-proposal)# exit
```

Configure an IPsec SA.

```
DeviceB(config)# ipsec sa sa1
DeviceB(config-ipsec-sa)# proposal proposal111
DeviceB(config-ipsec-sa)# sa spi ah 256
DeviceB(config-ipsec-sa)# sa authentication-hex ah
0123456789012345678901234567890123456789012345678901234567890123456789abcd
DeviceB(config-ipsec-sa)# exit
```

Configure an IP address for an interface and configure OSPFv3 on the interface.

```
DeviceB(config)# interface gigabitethernet 0/1
DeviceB(config-if-GigabitEthernet 0/1) ipv6 enable
DeviceB(config-if-GigabitEthernet 0/1) ipv6 address 2001:1::2/64
DeviceB(config-if-GigabitEthernet 0/1) ipv6 ospf 1 area 0
```

Apply the IPsec SA to the routing protocol.

```
DeviceB(config-if-GigabitEthernet 0/1)# ipv6 ospf ipsec sa sa1
```

5. Verification

Check information about the configured IPsec SA.

```
DeviceA# show ipsec manual sa sa1
ip security association name sa1(len 9), ref 0
  id: 2048(Activated)
```

```
  proposal name: proposal111
    encapsulation mode: transport
    transform: ah
    AH protocol: authentication sm3
    ESP protocol: authentication none, encryption none
  AH setting:
    AH spi: 256 (0x100)
    AH string-key:
    AH authentication hex key:
$10$181$huzuRca9EbwlWTHlMox5plYAfxdN7KC/5vaqeafY8QOjiAr0XjqBD3l6qBal1f6VzI95hKUGD/VdZFd96
drCwQ==$
  ESP setting:
    ESP spi: 0 (0x0)
    ESP string-key:
    ESP encryption hex key:
    ESP authentication hex key:
```

Check whether the OSPF neighbor status is correct on Device A.

```
DeviceA# show ipv6 ospf neighbor
OSPFv3 Process (1), 1 Neighbors, 1 is Full:
Neighbor ID     Pri   State           Dead Time   Instance ID  Interface
2.2.2.2           1   Full/DR         00:00:38    0            GigabitEthernet 0/1
```

Check whether the OSPF neighbor status is correct on Device B.

```
DeviceB# show ipv6 ospf neighbor
OSPFv3 Process (1), 1 Neighbors, 1 is Full:
Neighbor ID     Pri   State           Dead Time   Instance ID  Interface
1.1.1.1           1   Full/BDR        00:00:38    0            GigabitEthernet 0/1
```

6. Configuration Files

- Device A configuration file

```
hostname DeviceA
!
ipsec proposal proposal111
 encapsulation-mode transport
 transform ah
 ah authentication-algorithm sm3
!
ipsec sa sa1
 proposal proposal111
 sa spi ah 256
```

```
 sa authentication-hex ah 7
$10$019$O9nTeIgui7CbICSsRvbHgROzFLJaegdEAXPBjmiyLPVekQiznSmx69QTO9+VrB/xRMptvc
9KqkZkRtqfdD/OkQ==$
!
interface GigabitEthernet 0/1
 ipv6 enable
 ipv6 address 2001:1::1/64
 ipv6 ospf 1 area 0
 ipv6 ospf ipsec sa sa1
!
ipv6 router ospf 1
 router-id 1.1.1.1
!
end
```

- Device B configuration file

```
hostname DeviceB
!
ipsec proposal proposal111
 encapsulation-mode transport
 transform ah
 ah authentication-algorithm sm3
!
ipsec sa sa1
 proposal proposal111
 sa spi ah 256
 sa authentication-hex ah 7
$10$019$O9nTeIgui7CbICSsRvbHgROzFLJaegdEAXPBjmiyLPVekQiznSmx69QTO9+VrB/xRMptvc
9KqkZkRtqfdD/OkQ==$
!
interface GigabitEthernet 0/1
 ipv6 enable
 ipv6 address 2001:1::2/64
 ipv6 ospf 1 area 0
 ipv6 ospf ipsec sa sa1
!
ipv6 router ospf 1
 router-id 2.2.2.2
!
end
```

2 IKE

2.1 Overview

2.1.1 IKE Overview

When you configure IPsec, you can use the Internet Key Exchange (IKE) protocol to establish an SA. IKE is a key management protocol that implements the Oakley key exchange and Skeme key exchange within the Internet Security Association and Key Management Protocol (ISAKMP) framework. ISAKMP, Oakley, and Skeme are security protocols for implementing IKE. IKE provides IPsec with services such as automatic key negotiation and SA establishment, to simplify IPsec application and management, thereby greatly reducing the IPsec configuration and maintenance workloads.

- With IKE, many IPsec parameters, such as the keys, can be automatically configured, which simplifies configuration.
- IKE enables IPsec to provide the anti-replay service. IPsec uses the sequence number in IP packet headers to prevent replay. The sequence number is a 32-bit value. If the number is overflowed, an SA needs to be re-established to prevent replay. This process requires the IKE protocol.
- The identity authentication and management of each party in secure communication affect the IPsec deployment. The Certificate Authority (CA) or other organs that manage identity data in a centralized manner must be used in the large-scale application of IPsec. IKE provides end-to-end dynamic authentication.

2.1.2 Principles

1. Security Mechanism of IKE

IKE has a self-protection mechanism, which can securely authenticate identities, distribute keys, and establish IPsec SAs on an insecure network.

(1) Data authentication

Data authentication involves two concepts:

- Identity authentication: Identity authentication verifies identities of both communication parties. Pre-shared-key authentication is supported.
- Identity protection: Identity data is encrypted for transmission after a key is generated, thereby protecting the identity data.

(2) DH

The Diffie-Hellman (DH) algorithm is a public key algorithm. Both communication parties exchange data to calculate the shared key when no key is transmitted. In this way, even if a third party (such as a hacker) intercepts all exchange data used for calculating the key, the third party cannot calculate the authentic key because of high complexity of the DH algorithm. Therefore, the DH exchange technology ensures that both parties securely obtain the shared information.

During the DH exchange of IKE, each calculation and result are unrelated to another calculation and result. The DH exchange is performed during establishment of each SA, which ensures that keys used by SAs are irrelevant.

(3) PFS

The perfect forward secrecy (PFS) feature is a security feature, which ensures that the cracking of one key does not affect the security of other keys because these keys have no derivation relationship. IPsec is implemented by adding one key exchange to IKE phase 2 negotiation. The PFS feature is ensured by the DH algorithm.

2. Exchange Process of IKE

IKE negotiates keys and establishes SAs for IPsec in two phases:

(1) Phase 1: Both communication parties establish a secure tunnel that passes identity authentication, that is, establishes an ISAKMP SA. In phase 1, there are two IKE exchange modes: main mode and aggressive mode.

(2) Phase 2: IKE uses the secure channel established in phase 1 to negotiate the security service for IPsec. That is, IKE negotiates a specific SA used for secure transmission of IP data.

The IKE negotiation in main mode in phase 1 involves three pairs of messages:

- The first pair is the SA exchange messages, which are used to negotiate and determine relevant security policies.
- The second pair is the key exchange messages, which are used to exchange the Diffie-Hellman public value and auxiliary data (such as random number). The key is generated through this pair of messages.
- The last pair is the ID and authentication data exchange messages, which are used to authenticate identities and the content exchanged in phase 1.

The major difference between exchange in aggressive mode and that in main mode is as follows: Identity protection is not provided and only three messages are exchanged in aggressive mode. In scenarios with low requirements for identity protection, the aggressive mode, in which less packets are exchanged, can improve the negotiation speed. The main mode should be used in scenarios with high requirements for identity protection.

3. Mechanism

IPsec (IKE-reliant IPsec) must be configured and applied to interfaces before IKE starts working.

When outgoing data packets that meet requirements are detected on an interface, IPsec triggers IKE to negotiate with IKE of the remote peer. The IKE of both parties establish a secure tunnel to transmit various supported IPsec parameters, and finally establish consistent SAs at both ends so that IPsec of both parties works properly.

When the lifetime of an IPsec SA expires after a period of time, if data that meets requirements needs to be transmitted, the IKE of both parties start IPsec negotiation again.

2.1.3 Protocols and Standards

- RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP)
- RFC 2409: The Internet Key Exchange (IKE)
- RFC 2412: The OAKLEY Key Determination Protocol

2.2 Restrictions and Guidelines

- IKE is an application that runs over User Datagram Protocol (UDP). It uses UDP data packets and port 500. If an ACL (firewall) is configured on the device to deny UDP communication packets, IKE negotiation will fail. Therefore, ensure that communication packets of IKE are not denied.

- IKE policies are prioritized based on the policy number. The default policy number is 65535 and the default policy is used when no policy is configured.

2.3 Configuration Task Summary

IKE configuration includes the following tasks:

(1) Enabling IKE

(2) Configuring an IKE Policy

(3) Selecting the Work Mode

(4) (Optional) Configuring Optional Features of IKE. All of the following configuration tasks are optional. Select the tasks as required.

- Configuring the Local Identity
- Configuring Automatic Identification of the Work Mode
- Configuring DPD
- Configuring the Negotiation Rate Limit Function of IKE
- Configuring NAT Traversal
- Disabling the next-payload Field Check
- Configuring the First Remote Peer for Initiating Negotiation
- Disabling the Function of Sending the Device Vendor ID
- Configuring a Negotiation Policy for a Crypto Map
- Configuring the Multi-PEER Selection Mode
- Disabling Peer ID Check
- Configuring Interoperability with the Standby Link
- Configuring Phase 1 Negotiation Only for Standby Link Detection
- Configuring Compatibility with OpenWRT and

2.4 Enabling IKE

2.4.1 Overview

IKE is enabled by default and it does not need to be configured.

If you have executed the **no crypto isakmp enable** command to disable IKE, you need to enable the IKE first.

2.4.2 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable IKE.

crypto isakmp enable

IKE is enabled by default.

(4) (Optional) Configure the maximum number of IKE negotiation attempts.

crypto isakmp session limit *numbers*

By default, the number of IKE negotiation attempts is not limited.

2.5 Configuring an IKE Policy

2.5.1 Overview

To ensure successful IKE negotiation, the two parties engaged in IKE negotiation must have at least one set of consistent IKE policy. Multiple policies must be created on each peer, to ensure that at least one policy matches the policy on the remote peer. A unique priority (1–10000, with 1 indicating the highest priority) needs to be allocated to each created policy.

IKE tries to search for a consistent policy that exists on both parties when starting negotiation. One party that initiates negotiation sends all policies to the remote response party. The remote response party searches policies received from the remote peer by priority for a policy that matches a local policy.

When policies of both parties contain the same encryption algorithm, hash algorithm, authentication algorithm, and Diffie-Hellman parameter values and the lifetime specified in the policy on the remote peer is shorter than or equal to that in the compared policy, the policies are matched (if no lifetime is specified in the policy on a party, the shorter policy lifetime specified on the remote peer is used). If no acceptable policy is found, IKE rejects negotiation and no IPsec SA is established. If a matched policy is found, IKE completes negotiation and establishes an IPsec SA.

Each IKE policy defines five parameters.

Table 2-1 IKE Policy Parameters

Parameter	Keyword	Optional Value	Default Value
Encryption algorithm	des	56-bit DES-CBC	56-bit DES-CBC
	3des	168-bit 3DES-CBC	
	aes-128	128-bit AES-CBC	
	aes-192	192-bit AES-CBC	
	aes-256	256-bit AES-CBC	
	sm4	128-bit SM4-CBC	
	sm4-draft-version	128-bit SM4-CBC (draft standard in 2013)	
	sm4-standard-version	128-bit SM4-CBC (formal standard in 2014)	
Hash algorithm	sha	SHA-1 (HMAC variant)	SHA-1 (HMAC variant)
	md5	MD5 (HMAC variant)	

Parameter	Keyword	Optional Value	Default Value
	sha2-256	256-bit SHA-2-256 (HMAC variant)	
	sha2-384	384 bits SHA-2-384 (HMAC variant)	
	sha2-512	512 bits SHA-2-512 (HMAC variant)	
	sm3	SM3 (HMAC variant)	
Test methods	pre-share	Pre-shared key	Digital signature verification
	rsa-sig	Digital signature verification	
	digital-email	Digital envelope authentication	
Diffie-Hellman group ID	1	768-bit Diffie-Hellman group	768-bit Diffie-Hellman group
	2	1024-bit Diffie-Hellman group	
	5	1536-bit Diffie-Hellman group	
	14	2048-bit Diffie-Hellman group	
	15	3072-bit Diffie-Hellman group	
	16	4096-bit Diffie-Hellman group	
	17	6144-bit Diffie-Hellman group	
	18	8192-bit Diffie-Hellman group	
IKE SA lifetime	-	60 seconds to 86,400 seconds	86,400 seconds (1 day)

2.5.2 Restrictions and Guidelines

- If no policy is configured, the device uses the default policy, which is granted the lowest priority and uses the default value of each parameter.
- When you view the device configuration, the default policy and default values of the configured policies are

not displayed in the configuration. You can run the **show crypto isakmp policy** command to view the default policy and any default values in the configured policy.

2.5.3 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create an IKE policy of a specified priority and enter the IKE policy configuration mode.

crypto isakmp policy *priority*

No IKE policy is configured by default.

(4) Configure an encryption algorithm for the IKE policy.

encryption { **3des** | **aes-128** | **aes-192** | **aes-256** | **des** | **sm4** | **sm4-draft-version** | **sm4-standard-version** }

The default encryption algorithm of an IKE policy is 56-bit DES-CBC.

The data encryption algorithm specified by the command is used for encryption of IKE SA data. It differs from the encryption algorithm used by IPsec SAs.

(5) Configure a hash algorithm for the IKE policy.

hash { **md5** | **sha** | **sha2-256** | **sha2-384** | **sha2-512** | **sm3** }

The default hash algorithm of an IKE policy is SHA.

(6) Configure an authentication method for the IKE policy.

authentication { **digital-email asymmetric sm2** | **pre-share** | **rsa-sig** }

The default authentication method of an IKE policy is RSA algorithm authentication.

(7) Configure the Diffie-Hellman group identifier for the IKE policy.

group { 1 | 2 | 5 | 14 | 15 | 16 | 17 | 18 }

An IKE policy uses the 768-bit Diffie-Hellman group (group 1) by default.

(8) (Optional) Configure the IKE SA lifetime.

lifetime *lifetime*

The default IKE SA lifetime is 86,400 seconds (1 day).

IPsec SAs are negotiated on the basis of IKE SAs. Therefore, a longer lifetime should be configured for IKE SAs to shorten the time required for negotiating IPsec SAs. However, a longer lifetime indicates that SAs are more likely to be cracked. Therefore, you need to configure a proper lifetime.

(9) Return to the global configuration mode.

exit

(10) Configure a pre-shared key.

crypto isakmp key { **0** | **7** } *keystring* { **address** *peer-address* [*mask*] | **hostname** *peer-hostname* }

No pre-shared key is specified by default.

To enable IKE to conduct negotiation by using a pre-shared key, you must use this command to configure the same pre-shared key on both communication peers. When configuring IPv4 pre-shared keys with both

the peer-address and mask set to 0.0.0.0, or configuring IPv6 pre-shared keys with the ipv6-peer-string set to ::/0, IKE uses the default pre-shared key.

To ensure security, you are advised to configure different keys for different peer pairs.

2.6 Selecting the Work Mode

2.6.1 Overview

The IKE negotiation includes two phases:

In phase 1, a secure tunnel that passes authentication is established between two ISAKMP entities. The main mode or aggressive mode can be adopted in this phase.

In phase 2, service SAs are negotiated.

The main mode is adopted in phase 1 by default. When IP addresses are not statically configured, the aggressive mode can be used.

2.6.2 Restrictions and Guidelines

- In aggressive mode, identity IDs are not protected and less packets need to be negotiated. Select a proper work mode as required.

2.6.3 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the crypto map configuration mode.

crypto map *map-name sequence-number* **ipsec-isakmp** [**dynamic** *dynamic-map-name*]

(4) Select a work mode.

set exchange-mode { **aggressive** | **main** }

The default work mode for phase 1 of IKE negotiation between peers is the main mode.

(5) (Optional) Configure IPsec encryption for the third packet in aggressive mode.

crypto isakmp aggressive-encrypt enable

The third packet is encrypted in aggressive mode by default.

2.7 Configuring Optional Features of IKE

2.7.1 Configuring the Local Identity

1. Overview

The local identity configuration does not affect negotiation in main mode. In aggressive mode, local identity configuration specifies the identity type in the first negotiation message of the initiating party. Currently, the local identity can be configured in three forms: local address, domain name, and username@domain name.

The pre-shared key negotiation uses an IP address as the local identity by default. In some cases, other identity types are configured on the remote peer and modification is required on the local end.

2. Restrictions and Guidelines

- The command is configured globally instead of for a specific tunnel.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the local identity.

self-identity { **address** | **fqdn** *fqdn* | **trustpoint** *trustpoint* | **user-fqdn** *user-fqdn* }

The default form of the local identity is the local IP address.

2.7.2 Configuring Automatic Identification of the Work Mode

1. Overview

IKE supports two work modes for negotiation in phase 1: main mode and aggressive mode. In aggressive mode, identity IDs are not protected and the security level is lower than that in main mode. The main mode is adopted for negotiation by default.

IPsec implementation is not the same on devices of different vendors. Some devices use the aggressive mode for sending packets by default. When serving as a center device, the device needs to accept negotiation requests sent in the two modes, give response, and complete negotiation. Therefore, it is necessary to configure automatic identification of the work mode.

2. Restrictions and Guidelines

- This command is effective only to the IKE negotiation receiver.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure automatic identification of the work mode.

crypto isakmp mode-detect

The main mode is adopted for negotiation by default.

2.7.3 Configuring DPD

1. Overview

You are advised to configure the dead peer detection (DPD) function when the link is unstable. The function detects whether the peer device functions properly to eliminate tunnel vulnerabilities.

Currently, DPD is implemented using two mechanisms:

- **on-demand**: After the idle time of a tunnel exceeds the configured time, if a packet is sent, a DPD detection message is sent.
- **periodic**: After the configured time expires, the device actively sends DPD detection messages. A DPD detection message can be retransmitted for a maximum of five times.

The **on-demand** mechanism can reduce the additional overhead. The **periodic** mechanism is fast and needs to be selected based on the actual network conditions.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure DPD.

crypto isakmp keepalive *keepalive-time* [*retries*] [**on-demand** | **periodic**]

The DPD function is disabled by default.

2.7.4 Configuring the Negotiation Rate Limit Function of IKE

1. Overview

When thousands of tunnels are negotiated concurrently, the convergence fails or is slow during negotiation. As a result, the entire negotiation takes several hours or even longer. You can configure this command to limit the negotiation rate, to control the number of tunnels that are being negotiated concurrently within a certain range, so as to improve the negotiation efficiency.

2. Restrictions and Guidelines

- The negotiation rate limit function of IKE is enabled by default. The default rate limit is 1000, indicating that a maximum of 1000 tunnels can be negotiated concurrently. When a large number of tunnels are negotiated concurrently, if the default rate limit is adopted but the negotiation is still slow or fails, you can adjust the rate limit value.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the negotiation rate limit function of IKE. The configuration steps below are mutually exclusive. Please configure either of the following steps.

- Configure the IKE negotiation rate.

crypto isakmp limit rate *numbers*

The default negotiation rate is 1000, indicating that 1000 IPsec tunnels can be negotiated concurrently.

- Disable the negotiation rate limit function of IKE.

crypto isakmp limit disable

The negotiation rate limit function of IKE is enabled by default and the default negotiation rate is 1000.

2.7.5 Configuring NAT Traversal

1. Overview

The network address translation (NAT) traversal problem can be solved by adding a UDP header. The IKE protocol automatically determines whether the NAT traversal takes effect and provides the default parameter value. You can modify the parameter value based on NAT configuration. When no data is transmitted, the keepalive packet is used to ensure that the NAT records are effective, to prevent tunnel data transmission interruption caused by NAT port re-assignment.

The IPsec protocol implementation on devices of different vendors is different. Some devices support NAT traversal while some do not. In this case, you can disable NAT traversal to ensure device interworking.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configuration the NAT traversal function. The configuration steps below are mutually exclusive. Please configure either of the following steps.

- Configure the interval for sending NAT keepalive packets.

crypto isakmp nat keepalive *keepalive-time*

The default interval for sending NAT keepalive packets is 300 seconds.

- Disable NAT traversal.

crypto isakmp nat-traversal disable

NAT traversal is enabled by default.

2.7.6 Disabling the next-payload Field Check

1. Overview

In some cases, the domain of interpretation (DOI) field that cannot be identified needs to be ignored during negotiation. For this, you can configure this function.

2. Restrictions and Guidelines

- After the next-payload field check is disabled, if the value of the reserved field in a packet is not 0 or the field length does not match, a failure is still returned.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Disable the next-payload field check.

crypto isakmp next-payload disable

The next-payload field check is enabled by default.

2.7.7 Configuring the First Remote Peer for Initiating Negotiation

1. Overview

When 3G links are also used, if multiple groups of 3G dial-up addresses are configured and they map to remote peers configured in an IPsec crypto map, you can enable the peer binding function to speed up dial-up. Otherwise, the device needs to try several times to find the appropriate remote peer and it will take a long time to establish a tunnel for the first time.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the first remote peer for initiating negotiation.

crypto isakmp peer { **bind** | **random** }

By default, the first peer is selected.

2.7.8 Disabling the Function of Sending the Device Vendor ID

1. Overview

Devices of some vendors cannot identify private vendor IDs during IKE negotiation, which results in a negotiation failure. In this case, you can configure this command to disable the function of sending the device vendor ID.

2. Configuration Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Disable the function of sending the device vendor ID during IKE negotiation.

crypto isakmp vendorid disable

The device vendor ID is carried during IKE negotiation by default.

2.7.9 Configuring a Negotiation Policy for a Crypto Map

1. Overview

In the aggressive mode, a device in the branch sends only the IKE policy with the highest priority to a device in the HQ for negotiation by default. Therefore, if the device in the branch negotiates with the device in the headquarters in the aggressive mode, all the IKE policies with the highest priority on the devices in the HQ must be consistent with the IKE policy on the device in the branch, which reduces device compatibility. You can use this function to specify the IKE policy for negotiation for a crypto map. In this way, the IKE policies with the

highest priority on the devices in the HQ do not need to be consistent with the IKE policy on the device in the branch.

2. Restrictions and Guidelines

- The function takes effect only in a static crypto map and cannot be configured in a dynamic crypto map.

3. Configuration Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the crypto map configuration mode.

crypto map *map-name sequence-number* **ipsec-isakmp**

(4) Configure a negotiation policy for the crypto map.

set isakmp-policy *number*

No negotiation policy is configured for a crypto map by default.

2.7.10 Configuring the Multi-PEER Selection Mode

1. Overview

When multiple peers are configured, this parameter is used to select the first peer that initiates negotiation. When multi-peer configuration is used with an LTE link, configure multiple dial-up addresses for LTE dial-up and configure the peer in the IPsec map to match the dial-up addresses. You can enable the peer binding function to speed up the dial-up process. Otherwise, you need to retry multiple times to find the current peer, causing long waiting time for establishing a tunnel for the first time.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the peer selection mode.

crypto isakmp peer { **bind** | **random** }

By default, the system selects the first peer to initiate negotiation in the order of configuration.

2.7.11 Disabling Peer ID Check

1. Overview

During IKE negotiation, some vendor IDs cannot be identified, causing negotiation failure. You can configure this function to disable the peer ID check.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Disable the function of checking the peer ID during IKE negotiation.

crypto isakmp id-check-disable

By default, the peer ID is checked during IKE negotiation.

2.7.12 Configuring Interoperability with the Standby Link

1. Overview

In scenarios with active and standby links or multi-links, IPsec monitors the status of the active link. When the active link is Up, IPsec automatically deletes the IPsec tunnel of the standby link, deleting the reverse route and enabling normal forwarding of service data. The active link is monitored through Track and DLDP.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure interoperability between IPsec tunnel and the standby link.

crypto isakmp link-redundency backup *backup-interface* { **intf-down** *master-interface* | **track** *track-id* }

No interoperability between IPsec tunnel and the standby link is configured by default.

2.7.13 Configuring Phase 1 Negotiation Only for Standby Link Detection

1. Overview

In a scenario with active and standby links or a scenario with multiple links, traffic is normally transmitted only through the IPsec tunnel on the active link, and no IPsec tunnel is established on the standby link. To ensure that the standby link is also available and can be switched to when the active link fails, the standby link needs to be periodically detected. This function is used in combination with the SNC. That is, the SNC periodically initiates standby link detection, the router performs the detection, and reports the detection result to the SNC. By default, the router detects that an IPsec tunnel is fully established on the standby link. However, if the full establishment of an IPsec tunnel causes network flapping, you can configure Phase 1 negotiation only for the standby link detection.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure only Phase 1 in standby link detection.

crypto isakmp link-redundency detect ike

By default, a complete IPsec tunnel is established for standby link detection.

2.7.14 Configuring Compatibility with OpenWRT and Sangfor Devices

3. Overview

This function is used to configure compatibility with OpenWrt and Sangfor devices, where after the expiration of the Phase 1, Phase 2 will not initiate its own expiration process but is directly reattached to a new Phase 1.

4. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure compatibility with OpenWRT and Sangfor devices.

crypto isakmp owt-compatibility

By default, the compatibility with OpenWrt and Sangfor devices is disabled.

2.8 Monitoring

Run the **show** command to check the configuration.

Run the **clear** command to clear information.

Note

- Debugging occupies system resources, so disable it immediately if not required.
- During device operation, running the **clear** command may cause service interruption due to key information loss.

Table 2-2 IKE Monitoring

Command	Purpose
show crypto isakmp ippool	Displays IKE address pool information.
show crypto isakmp neg-counter	Displays statistics on the exception process during IPsec negotiation.
show crypto isakmp policy	Displays all parameters of an IKE policy.
show ipsec manual key-id	Display the IPsec manual key ID.
show ipsec manual sa [*sa-name*]	Display information about a configured SA.
show ipsec manual spi	Display the security parameter index (SPI) of IPsec.
show crypto isakmp sa [**ipv6-peer** *ipv6-peer-address* | **peer** *peer-address*]	Displays all current IKE SAs.

Command	Purpose
clear crypto isakmp [*connection-id*]	Clears an IKE connection.
clear crypto sa [peer *ipv4-address* | **map** *map-name* | **spi** *destination-address* { **ah** | **esp** } *spi* **]**	Clears an IPsec SA.

Contents

1 Configuring VPDN

1.1 Introduction

1.1.1 Overview

The virtual private dial-up network (VPDN) is a type of virtual private network (VPN) services based on dial-up users. It connects to the Internet in dial-up access mode and utilizes the bearer function of the IP network, in combination with authentication and authorization mechanisms, to establish a secure VPN. The VPDN technology develops with the Internet development. VPDN is applicable to scenarios in which customers have branches distributed scatteredly as well as many mobile users, for example, enterprise users and remote learning users. The device supports Layer 2 Tunneling Protocol (L2TP) tunnels.

- L2TP

  L2TP is a standard Internet tunneling protocol in industry. It has similar functions to the PPTP protocol, for example, encrypting network data flows. Their differences are as follows: PPTP requires an IP network while L2TP requires data packet-oriented point-to-point connections; PPTP uses a single tunnel while L2TP uses multiple tunnels; L2TP provides packet header compression and tunnel authentication while PPTP does not.

  L2TP is proposed by the IETF by integrating two existing tunneling protocols: Layer 2 Forwarding (L2F) protocol of Cisco and PPTP protocol of Microsoft. It is documented in RFC 2661.

  L2TP is an extension to PPP. It utilizes PPP to implement user identity authentication and data transmission. Different from PPTP, which uses the Transmission Control Protocol (TCP) for data transmission, L2TP uses the User Datagram Protocol (UDP) to transmit control messages and data messages.

  L2TP is also an important and effective method for implementing VPN. VPN enables both dial-up users and network access users to conveniently and securely access internal networks of enterprises.

1.1.2 Basic Concepts

- Local address pool

  The server accepts VPDN connection requests initiated by remote clients. If a remote client has no IP address to be used in a VPN, such an IP address needs to be assigned to it. In general, the server allocates an idle IP address in a specified address pool to a client.

- User information

  User information is the basis for the local device to conduct identity authentication on remote access clients. The server locally maintains a database that stores the names and passwords of users who are allowed to access (dial in to) the server.

- VPDN global parameters

  VPDN global parameters include the following:

  - Enabling/Disabling the VPDN function: If a device is required to accept the access from remote clients and establish tunnels and sessions, the VPDN function must be enabled on the device.
  - VPDN source address: After the VPDN source address is set, the tunnel destination address configured on a remote client must be consistent with the VPDN source address so that a tunnel is established

successfully.

- Maximum number of VPDN sessions: After the maximum number of VPDN sessions is configured, an access request is rejected when the number of existing VPDN sessions reaches the maximum limit.
- VPDN rate limit: Users can limit the global VPDN tunnel establishment rate, so as to limit the number of VPDN tunnels that can be established at a time.

- Virtual-VPDN interface

  A virtual-VPDN includes the following:

  - Virtual-VPDN interface (mandatory): The created virtual-VPDN will be used a configuration template of the virtual-access interface that binds and bears sessions.
  - Local IP address (mandatory)
  - Peer IP address (optional)

- VPDN-group

  A VPDN-group includes the following:

  - VPDN-group (mandatory): The destination address in a tunnel establishment request sent from a remote client must match the VPDN-group local address so that the VPDN-group is applied.
  - Tunnel mode (mandatory): You can configure whether the device accepts the dial-in from remote clients. If the local device is required to provide the server function, the device must be configured to accept dial-in from remote clients.
  - Tunnel protocol (mandatory). L2TP is supported.
  - Remote hostname (optional): If a remote hostname is configured, the VPDN-group is effective only for a remote client that matches the remote hostname. If no remote hostname is configured, the VPDN-group becomes the default VPDN-group of the system and can provide the VPDN service for any remote client.
  - Local hostname (optional)
  - Tunnel parameters (optional)

- Maximum number of VPDN sessions

  Set the maximum number of sessions that are allowed by the VPDN server.

- L2TP-class interface

  An L2TP-class interface includes the following:

  - L2TP-class unit: It is used to set parameters related to L2TP control connections. A created L2TP-class interface can be referenced by a pseudowire-class interface by name.
  - Maintenance update parameter for L2TP control connections: You can set the hello message transmission interval.

- Pseudowire-class interface

  A pseudowire-class interface includes the following:

  - Pseudowire-class unit: It can be referenced by name in pseudowire rules of a virtual-ppp interface.
  - Encapsulation mode for L2TP data transmission: The encapsulation mode for data transmission over an L2TP tunnel cannot be changed after it is set. If you need to set L2TP data transmission parameters for a pseudowire-class interface, the encapsulation mode for L2TP data transmission must be configured first.

- L2TP control connection parameter. The control parameter can be set to L2TPv2, indicating that a control connection is created according to the L2TP protocol documented in RFC 2661. The *L2TP-class-name* parameter is used to reference an existing L2TP-class interface to limit the value of the control connection parameter. If this parameter is not set, the default L2TP control connection parameter of the system is used.

- Virtual-ppp interface

  A virtual-ppp interface includes the following:

  - Virtual-ppp interface: It is used to create and bind a specified L2TP session.
  - IP address
  - Identity authentication parameters: They are used to set the username and password.
  - Pseudowire rule: Users can set pseudowire rules for establishing an L2TP session on a virtual-ppp interface. Once pseudowire rules are set on a virtual-PPP interface, the virtual-ppp interface automatically attempts to establish an L2TP session with a specified L2TP network server (LNS). If the session fails to established, the virtual-ppp interface attempts to establish an L2TP session 10 seconds later again.

1.1.3 Protocols and Standards

- RFC 2661: Layer Two Tunneling Protocol "L2TP"

1.2 Configuring Parameters to Make the Device Initiate an L2TP Connection Request as a Local Client

1.2.1 Overview

When the device needs to serve as an L2TP client and actively initiate negotiation to establish a tunnel with the remote L2TP server, configure this function.

1.2.2 Restrictions and Guidelines

- The device establishes an L2TP session by using a specified LNS name but the precondition is that the domain name system (DNS) service must be enabled. The device provides only the DNS client service and the specified LNS name must have been registered with the DNS server.

1.2.3 Configuration Tasks

Configuring parameters to make the device initiate an L2TP connection request as a local client includes the following tasks:

1. (Optional) Configuring an L2TP-Class Interface
2. (Optional) Configuring a Pseudowire-Class Interface
3. Configuring a Virtual-PPP Interface

1.2.4 Configuring an L2TP-Class Interface

1. Overview

You can configure an L2TP-class interface to set L2TP control connection parameters. The created L2TP-class interface can be referenced by a pseudowire-class interface.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create an L2TP-class interface and enter the L2TP-class interface configuration mode.

l2tp-class *l2tp-class-name*

No L2TP-class interface is configured by default.

(4) (Optional) Configure time parameters for L2TP control connections. Configure at least one of the tasks.

- Configure the size of the receive window for control connections.

  receive-window *size*

  The default size of the control message receive window is **8**.

- Configure control connection retransmission parameters.

  retransmit { **initial** { **retries** *initial-retries* | **timeout** { **max** *initial-timeout* | **min** *initial-timeout* } } | **retries** *retries* | **timeout** { **max** *timeout* | **min** *timeout* } }

  By default, the retransmission count of SCCRQ messages is **2**, the retransmission count of other control messages is **5**, the minimum retransmission interval of control messages is **1** second, and the maximum retransmission interval of control messages is **8** seconds.

- Configure the maximum allowable time for establishing a control connection.

  timeout setup *max-time*

  The default maximum allowable time for establishing a control connection is **120** seconds.

(5) (Optional) Configure authentication parameters for L2TP control connections.

a Enable tunnel authentication.

authentication

Tunnel authentication is disabled and the device name is used as the local hostname by default.

If tunnel authentication is enabled, you need to configure the following two commands:

b Configure the local hostname for an L2TP tunnel.

hostname *host-name*

The device name is used as the local hostname of an L2TP tunnel by default.

c Configure the tunnel authentication password.

password [**0** | **7**] *pass-words*

No tunnel authentication password is configured by default.

The same tunnel authentication password must be used at both ends.

5. Configure the maintenance update parameter for L2TP control connections.

   hello *interval*

   The default transmission interval of hello messages is **60** seconds.

1.2.5 Configuring a Pseudowire-Class Interface

1. Overview

A created pseudowire-class can be referenced in pseudowire rules of a virtual-ppp interface.

2. Procedure

1. Enter the privileged EXEC mode.

   enable

2. Enter the global configuration mode.

   configure terminal

3. Create a pseudowire-class interface and enter the pseudowire-class interface configuration mode.

   pseudowire-class *pseudowire-class-name*

   No pseudowire-class interface is configured by default.

4. (Optional) Configure the encapsulation mode for data transmission.

   encapsulation l2tpv2

   No data encapsulation mode is configured for a tunnel by default.

   The encapsulation mode for data transmission over an L2TP tunnel cannot be changed once set. If you need to configure L2TP data transmission parameters for a pseudowire-class interface, you must set the encapsulation mode for L2TP data transmission first.

5. (Optional) Disable tunnel data fragmentation.

   ip dfbit set

   Tunnel data can be fragmented for transmission by default.

6. (Optional) Configure the TTL in IP headers for a tunnel.

   ip ttl *ttl-value*

   The default value of the TTL field in the IP headers of tunnel data is 255.

7. (Optional) Configure the local interface for a tunnel.

   ip local interface *interface-type interface-number*

   No local interface is configured for a tunnel by default.

8. (Optional) Configure the L2TP control connection parameter.

   protocol l2tpv2 [*l2tp-class-name*]

   No L2TP control connection parameter is configured by default.

1.2.6 Configuring a Virtual-PPP Interface

1. Overview

After a pseudowire rule is configured on a virtual-ppp interface, the virtual-ppp interface automatically attempts to establish an L2TP session with a specified LNS. If an L2TP session fails to be established, the virtual-ppp interface makes another attempt 10 seconds later.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a virtual-ppp interface and enter the virtual-ppp interface configuration mode.

interface virtual-ppp *number*

The created virtual-ppp interface is used to create and bind an L2TP session.

(4) Configure an IP address.

ip address negotiate

(5) Configure identity authentication parameters. The following uses PAP as an example.

ppp pap sent-username username password *password*

(6) Configure a pseudowire rule.

pseudowire { *peer-ipv4-address* | **hostname** *peer-hostname* } *vcid* { **encapsulation l2tpv2** [**pw-class** *pw-class-name*] | **pw-class** *pw-class-name* }

No pseudowire-class interface is configured by default.

1.3 Configuring Parameters to Make the Device Accept the L2TP Connection Request from a Remote Client

1.3.1 Overview

When the device needs to accept a connection request from an L2TP remote client and negotiate to establish a tunnel, you can configure this function.

1.3.2 Restrictions and Guidelines

- The VPDN function of Ruijie devices becomes available/unavailable immediately after it is enabled/disabled. If VPDN is disabled, all existing L2TP tunnels and sessions will be released.

1.3.3 Configuration Tasks

Configuring parameters to make the device accept the L2TP connection request from a remote client includes the following tasks:

(1) (Optional) Configuring a Local Address Pool

(2) (Optional) Configuring User Information

3. Configuring VPDN Global Parameters
4. Configuring a Virtual-VPDN Interface
5. Configuring a VPDN-Group

1.3.4 Configuring a Local Address Pool

1. Overview

The LNS accepts L2TP connection requests initiated by remote clients. If a remote client has no IP address to be used in a VPN, such an IP address needs to be assigned to it. In general, an idle IP address in a specified address pool is assigned to the client.

2. Procedure

1. Enter the privileged EXEC mode.

   enable

2. Enter the global configuration mode.

   configure terminal

3. Create a local address pool.

   ip local pool *poolname first-ip* [*last-ip*]

1.3.5 Configuring User Information

1. Overview

User information is used to conduct user identity authentication on clients that remotely access the local device via L2TP. The device locally maintains a database that stores the names and passwords of users who are allowed to access (dial in to) the device.

2. Procedure

1. Enter the privileged EXEC mode.

   enable

2. Enter the global configuration mode.

   configure terminal

3. Configure user information.

   username *user-name* **password** *password*

1.3.6 Configuring VPDN Global Parameters

1. Overview

VPDN global parameters must be configured for the establishment of an L2TP tunnel initiated by a remote client.

2. Procedure

1. Enter the privileged EXEC mode.

   enable

2. Enter the global configuration mode.

configure terminal

(3) Enable the VPDN function.

vpdn enable

The VPDN function is disabled by default.

If a device is required to accept L2TP access requests from remote clients and establish L2TP tunnels and sessions, the VPDN function must be enabled on the device.

The VPDN function of Ruijie products becomes available/unavailable immediately after it is enabled/disabled. Disabling the VPDN function will cause the release of all existing L2TP tunnels and sessions.

(4) (Optional) Enable VPDN congestion control.

vpdn congestion_avoidanc

The VPDN congestion control function is disabled by default.

(5) (Optional) Configure the VPDN source address.

vpdn source-ip *ipv4-address*

No local (source) address for providing the VPDN function is configured by default.

After the VPDN source address is configured, the tunnel destination address configured for a remote client must be consistent with the VPDN source address so that an L2TP tunnel is successfully established. The system does not check whether the destination address in a received tunnel establishment request is a specific value by default.

(6) (Optional) Configure the maximum number of VPDN sessions.

vpdn session-limit *sessions*

The maximum number of VPDN sessions is 256 by default.

(7) (Optional) Configure the function of ignoring the VPDN source address check.

vpdn ignore_source

The function of ignoring the VPDN source address check is disabled by default, that is, the source addresses of tunnel packets are checked by default.

(8) (Optional) Configure the VPDN tunnel establishment rate limit.

vpdn limit-rate *limit-number*

The default maximum number of VPDN tunnels that can be established concurrently is **15**.

1.3.7 Configuring a Virtual-VPDN Interface

1. Overview

Create a virtual-VPDN for configuring a virtual-access interface, and bind the virtual-access interface to an L2TP session to bear session traffic.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Create a virtual-VPDN interface and enter the virtual VPDN interface configuration mode.

interface virtual-vpdn *interface-number*

(4) Configure the local IP address.

ip address *ipv4-address mask*

No IP address is configured by default.

(5) (Optional) Configure the peer IP address.

peer default ip address pool [*pool-name*]

An interface does not assign an IP address to the peer by default.

1.3.8 Configuring a VPDN-Group

1. Overview

A VPDN-group must be configured for the establishment of an L2TP tunnel to be initiated by a remote client. The created VPDN-group interface allows a client to access the interface and establish a tunnel.

2. Procedure

1. Enter the privileged EXEC mode.

   enable

2. Enter the global configuration mode.

   configure terminal

3. Create a VPDN-group interface and enter the VPDN-group interface configuration mode.

   vpdn-group *name*

   No VPDN-group interface is configured by default.

4. Set the tunnel work mode to acceptance of remote client dial-in.

   accept-dialin

   No tunnel work mode is configured by default.

   If the local device needs to provide the LNS function, the device must be configured to accept the dial-in from remote clients.

5. Configure the tunnel protocol.

   protocol l2tp

   No L2TP control connection parameter is configured by default.

   The tunnel mode must be set before the tunnel protocol is configured. If the local device needs to function as an LNS, **protocol l2tp** must be configured.

6. Configure a virtual-VPDN to be used.

   virtual-vpdn *number*

   No virtual-VPDN interface is bound to a VPDN-group by default.

   Before a virtual-VPDN is configured for a VPDN-group, the tunnel mode must be set.

7. Return to the VPDN-group interface configuration mode.

exit

[8] (Optional) Configure the remote hostname.

terminate-from hostname *name*

No remote hostname is configured for a tunnel by default.

If the remote hostname is configured, the VPDN-group is effective only for the remote client that matches the remote hostname. If no remote hostname is configured, the VPDN-group becomes the default VPDN-group of the system and can provide the VPDN service for any remote client.

[9] (Optional) Configure the local hostname.

local name *name*

The device name is used as the local hostname of a tunnel by default.

[10] (Optional) Configure the VPDN-group local address.

source-ip *ipv4-address*

No local address is configured for a VPDN-group tunnel by default.

The destination address in the tunnel establishment request sent from a remote client must match the VPDN-group local address so that the VPDN-group is applied.

[11] (Optional) Configure L2TP parameters as required.

- Configure the checksum field for UDP packets.

  l2tp ip udp checksum

  The default value of the checksum field in UDP packets that carry tunnel data is null (that is, 0).

- Enable tunnel authentication.

  l2tp tunnel authentication

  Tunnel authentication is disabled by default.

- Configure an interval for sending hello packets.

  l2tp tunnel hello *interval*

  The default interval for sending hello packets to keep a tunnel alive is 60 seconds.

- Configure the tunnel authentication password.

  l2tp tunnel password [0 | 7] *password*

  No tunnel authentication password is configured by default.

  If L2TP tunnel authentication is required, the same tunnel password must be configured at both ends of an L2TP tunnel.

- Configure the size of the control message receive window for a tunnel.

  l2tp tunnel receive-window *size*

  The default size of the control message receive window of a tunnel is 4.

- Configure control message retransmission parameters for a tunnel.

  l2tp tunnel retransmit { **retries** *number* | **timeout** { **min** *seconds* | **max** *seconds* } }

  By default, the maximum retransmission count of control messages is 5, and the minimum and maximum retransmission intervals of control messages are 1 second and 8 seconds, respectively.

- Configure the maximum time for tunnel setup with no session or for control connection setup.

  l2tp tunnel timeout { **no-session** *interval* | **setup** *interval* }

  By default, the maximum time for tunnel setup with no session is 600 seconds, and the maximum time for control connection setup (tunnel setup time) is 120 seconds.

- Enable forcible packet encryption.

  l2tp tunnel force_ipsec

  Forcible packet encryption is disabled by default.

  Run this command when external encryption is used. After this command is configured, only encrypted packets can pass through VPDN tunnels.

- Configure support for the RFC 2661-compliant hidden attribute-value pair (AVP) parsing algorithm.

  l2tp tunnel avp-hidden-compatible

  The Cisco standard-compliant hidden AVP parsing algorithm is supported by default.

  After the device is configured to support the RFC 2661-compliant hidden AVP parsing algorithm, the device parses the hidden AVPs according to RFC 2661.

- Configure the interval for clearing expired sessions.

  l2tp tunnel clear timeout *time*

  The software clears expired sessions immediately by default.

- Enable RFC compatibility. After this function is enabled, the device does not send a stop packet when receiving duplicate SCCRQ packets.

  l2tp tunnel none-rfc-compatible send-stop-pkt

  RFC compatibility is disabled by default.

- Configure the function of ignoring errors in L2TP control packets from the peer device.

  lcp renegotiation always

  Received L2TP control packets must strictly comply with specifications by default.

- Configure the type of service (ToS) field for IP headers.

  ip tos *tos-value*

  The default value of the ToS field in the IP headers of tunnel data packets is 0, indicating normal.

- Configure the precedence field for IP headers.

  ip precedence *value*

  The default value of the precedence field in IP headers of tunnel data packets is 0, indicating routine.

1.4 Monitoring

Run the **show** command to check the configuration.

Run the **debug** command to output debugging information.

Run the **clear** command to clear information.

Caution

- The output debugging information occupies system resources. Therefore, disable the debugging function immediately after use.

- Running the **clear** command may lose vital information and thus interrupt services.

Table 1-1 VPDN Monitoring

Command	Purpose
show vpdn [**session** [**l2tp** [**interface** *interface-type interface-number*]] | **tunnel** [**l2tp** *locid*]]	Displays information about the current VPDN session, tunnels, or a tunnel with a specified ID.
show vpdn log [**user** *username*]	Displays the login and logout information of all users or a specified user in the current log file.
show l2tp-class [*l2tp-class-name*]	Displays detailed configuration of all L2TP-class interfaces or a specified L2TP-class interface configured in the system.
show pseudowire-class [*pseudowire-class-name*]	Displays detailed configuration of all pseudowire-class interfaces or a specified pseudowire-class interface configured in the system.
clear vpdn tunnel [**l2tp** [**id** [*locid*] | *remote-host-name*]]	Clears a specified tunnel.
debug vpdn error	Debugs VPDN errors.
debug vpdn packet	Debugs VPDN packets.
debug vpdn l2x-errors	Debugs VPDN l2x-errors.
debug vpdn l2x-packets	Debugs VPDN l2x-packets.
debug vpdn event	Debugs VPDN events.

1.5 Configuration Examples

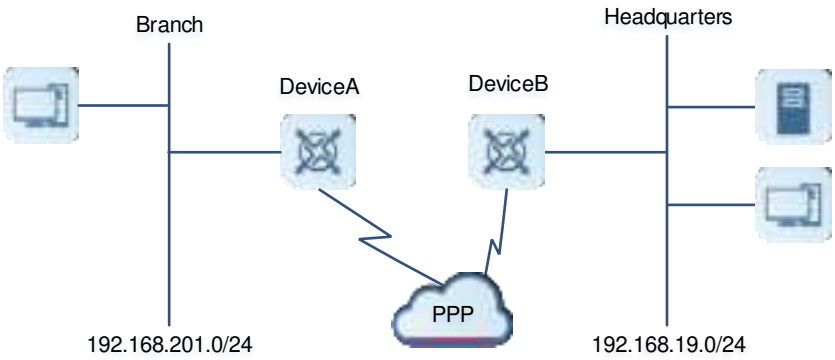
1.5.1 Configuring L2TP Parameters

1. Requirements

Device A serves as the gateway of the branch network and an L2TP client. Device B serves as the gateway of the headquarters (HQ) network and L2TP server. The client needs to access the HQ network.

2. Topology

Figure 1-1 L2TP Topology



3. Notes

- Configure Device A as an L2TP client.
- Configure Device B as the L2TP server to allow the client to access through L2TP.

4. Procedure

(1) Configure Device A as an L2TP client.

Configure the L2TP data transmission and encapsulation modes for the client.

```
Device> enable
Device# configure terminal
Device(config)# l2tp-class 1
Device(config-l2tp-class)# exit
Device(config)# pseudowire-class 1
Device(config-pw-class)# encapsulation l2tpv2
Device(config-pw-class)# exit
```

Configure a tunnel interface for the client, and set a pseudowire rule to specify the server address to be connected.

```
Device(config)# interface virtual-ppp 1
Device(config-if-virtual-ppp 1)# pseudowire hostname mm.hxs.meibu.com 1
encapsulation l2tpv2
```

Configure the PAP authentication mode for the client and set the username to **user** and password to **password@123**.

```
Device(config-if-virtual-ppp 1)# ppp pap sent-username user password
password@123
```

Configure the negotiation mode for the client to obtain a tunnel address so that the server dynamically assigns an address to the client after the client passes authentication.

```
Device(config-if-virtual-ppp 1)# ip address negotiate
Device(config-if-virtual-ppp 1)# exit
```

Configure the route from the branch network of the client to the HQ network of the server to pass through the virtual-ppp interface.

```
Device(config)# ip route 192.168.19.0 255.255.255.0 virtual-ppp 1
```

(2) Configure Device B as the L2TP server.

Enable the VPDN function.

```
Device> enable
Device# configure terminal
Device(config)# vpdn enable
```

Configure a VPDN-group unit on the server to allow the dial-in from the remote client, and set the tunnel protocol to L2TP.

```
Device(config)# vpdn-group 1
Device(config-vpdn)# accept-dialin
Device(config-vpdn-acc-in)# protocol l2tp
Device(config-vpdn-acc-in)# virtual-vpdn 1
Device(config-vpdn-acc-in)# exit
```

Configure an address pool for the server to dynamically assign a tunnel IP address to the client after the client passes authentication.

```
Device(config-vpdn)# ip local pool l2tp 1.1.1.2 1.1.1.254
Device(config-vpdn)# exit
```

Configure the username and password.

```
Device(config)# username user password 0 password@123
```

Configure a virtual-VPDN for the server and set the PPP authentication mode to PAP.

```
Device(config)# interface virtual-vpdn 1
Device(config-if-virtual-vpdn 1)# ppp authentication pap
```

Associate the virtual-VPDN with the local address pool.

```
Device(config-if-virtual-vpdn 1)# ip address 1.1.1.1 255.255.255.0
Device(config-if-virtual-vpdn 1)# peer default ip address pool l2tp
```

5. Verification

L2TP configuration is not required for a PC on the branch network. Instead, the network administrator only needs to assign an internal network address (for example, 192.168.201.213) to the PC and set the gateway to 192.168.201.1. Check whether the PC on the branch network can access the HQ network segment 192.168.19.0 of the server.

6. Configuration Files

(1) Device A configuration file

```
hostname DeviceA
!
l2tp-class 1
!
pseudowire-class 1
 encapsulation l2tpv2
```

```
!
interface Virtual-ppp 1
 ppp pap sent-username user password password@123
 ip address negotiate
 pseudowire hostname mm.hxs.meibu.com 1 encapsulation l2tpv2
!
ip route 192.168.19.0 255.255.255.0 Virtual-ppp 1
!
end
```

(2) Device B configuration file

```
hostname DeviceB
!
ip local pool l2tp 1.1.1.2 1.1.1.254
!
username user password password@123
!
vpdn enable
!
vpdn-group 1
 accept-dialin
  protocol l2tp
  virtual-vpdn 1
!
interface Virtual-vpdn 1
 ppp authentication pap
 ip address 1.1.1.1 255.255.255.0
 peer default ip address pool l2tp
!
end
```

Contents

1 Configuring the PPPoE Client

1.1 Introduction

1.1.1 Overview

Point-to-Point Protocol over Ethernet (PPPoE) enables Ethernet hosts to be connected to a remote access concentrator through a simple bridging device. With PPPoE enabled, the remote access device can carry out control and accounting for access users. Compared to traditional access methods, PPPoE is cost-effective and widely used in various applications, including community network construction. Moreover, the commonly used Asymmetric Digital Subscriber Line (ADSL) also utilizes the PPPoE protocol. The PPPoE protocol includes a PPPoE client and a PPPoE server, wherein the PPPoE client is responsible for dial-up and initiating PPPoE connection requests.

Note

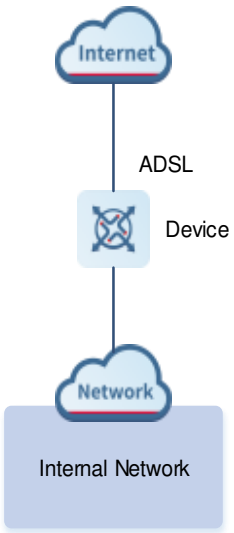
The following sections introduce only the PPPoE client.

1.1.2 Principles

1. Internet Access Through Dial-up

In a scenario where ADSL is used to access the Internet, the device provides dial-up and packet forwarding functions. In Figure 1-1, after the device has finished dial-up, it can access the Internet, and hosts connected to the device on the intranet also gain access to the Internet.

Figure 1-1 Internet Access Through Dial-up



Dial-up and Internet access correspond to negotiation and message forwarding, respectively. Negotiation can be further classified into protocol negotiation, protocol keepalive, and protocol termination.

- Protocol negotiation

  Protocol negotiation includes PPPoE negotiation and PPP negotiation.

  In PPPoE negotiation, both parties involved in the negotiation record each other's MAC address to uniquely identify each other and establish a unique session ID. During this process, PPPoE goes through the following five statues sequentially:

  a SENT_IDLE: Idle state.

  b SENT_PADI: The PPPoE client broadcasts a PPPoE Active Discovery Initial (PADI) packet.

  c RECEIVED_PADO: The PPPoE server in the network responds to the PADI packet by sending a PPPoE Active Discovery Offer (PADO) packet. When the PPPoE client receives the first PADO packet, its PPPoE status changes to RECEIVED_PADO.

  d SENT_PADR: The PPPoE client unicasts a PPPoE Active Discovery Request (PADR) packet. Upon receiving the PADR from the client, the PPPoE server generates a session ID and records it in a PPPoE Active Discovery Session-confirmation (PADS) message, which is then sent back to the PPPoE client.

  e SESSION: After receiving the PADS packet, the PPPoE client changes its status to SESSION. In the subsequent interaction process, both sides exchange packets carrying this session ID. At this point, the PPPoE session is established, and the PPP negotiation process will take place.

  In PPP negotiation, the server verifies the client's authentication information. If the verification is successful, the server assigns an IP address to the client. The server agrees to assign the IP address as the client's designated IP address if the client meets the server's requirements and is assigned with an IP address.

  After both negotiations are completed, the device gains access to the Internet and encapsulates data packets with the Layer 2 header.

- Protocol keepalive

  After PPP negotiation is completed, both parties will regularly exchange heartbeat packets. If the local end does not receive a heartbeat packet from the remote end within a certain period of time, the local end will proactively terminate the connection.

- Protocol termination

  The active party initiating connection termination will first send a PPP termination packet to end the PPP session, and then send a PPPoE termination packet to end the PPPoE session.

  The passive party receiving the PPP termination packet will send an acknowledgment packet to terminate the PPP session. Upon receiving the PPPoE termination packet, it will send an acknowledgment packet to terminate the PPPoE session. The PPPoE status is set to TERMINATED.

  Once both parties receive the PPPoE termination packet, even if no PPP termination packet is received, both the PPP and PPPoE sessions will immediately be terminated.

2. Dial-up Mode

- Automatic dial-up

  No Dail-on-Demand Routing (no-DDR): Dial-up is initiated automatically when the device is powered on or disconnected.

1.1.3 Protocols and Standards

- RFC 2516: A Method for Transmitting PPP Over Ethernet (PPPoE)
- RFC 1661: The Point-to-Point Protocol (PPP)

1.2 Configuration Task Summary

The PPPoE client configuration includes the following tasks:

(1) Configuring a Dialer Interface

(2) Configuring PPP Parameters.

(3) Configuring the Primary Interface

1.3 Configuring the PPPoE Client

1.3.1 Overview

The PPPoE client program on the device starts dial-up to connect to a remote ISP and gain access to the Internet through an ADSL line. The device also forwards Internet traffic from intranet PCs.

1.3.2 Configuration Task

(1) Configuring a Dialer Interface

(2) Configuring PPP Parameters

(3) Configuring the Primary Interface

1.3.3 Configuring a Dialer Interface

1. Overview

The dialer interface is the logical interface used by the PPPoE client for dial-up.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Add a dialer interface and enter the dialer interface configuration mode.

interface dialer *dialer-number*

(4) Configure a mode for the dialer interface to obtain an IPv4 address.

ip address { **negotiate** | *ipv4-address mask* }

No IPv4 address is configured for a dialer interface by default.

If you manually specify the IPv4 address of a dialer interface, the local device must obtain approval of the remote end during negotiation.

(5) Associate a dialing pool with the dialer interface.

dialer pool *pool-number*

A dialer interface is not associated with any dialing pool by default.

1.3.4 Configuring PPP Parameters

1. Overview

The encapsulation protocol must be configured as PPP for the dialer interface so that the PPPoE client can work properly.

To ensure proper authentication, you are advised to configure usernames and passwords for both CHAP (Challenge Handshake Authentication Protocol) and PAP (Password Authentication Protocol) authentication. This is because it may not be possible to determine which mode an ISP is using for authentication.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the dialer interface configuration mode.

interface dialer *dialer-number*

(4) Configure the MTU of the dialer interface.

mtu *value*

The MTU is **1500** by default.

PPPoE is used for Internet access, resulting in an additional 12 bytes of link-layer information compared to a typical Ethernet packet. Thus, it is recommended that the MTU be set to 1488.

(5) Configure the username for CHAP authentication.

ppp chap hostname *username*

No username for CHAP authentication is configured by default.

(6) Configure the password for CHAP authentication.

ppp chap password *password*

No password for CHAP authentication is configured by default.

(7) Configure the username and password for PAP authentication.

ppp pap sent-username *username* **password** *password*

The username and password for PAP authentication are not configured by default.

1.3.5 Configuring the Primary Interface

1. Overview

To enable the dialing function for the dialer interface of the PPPoE client, you must enable the PPPoE client on the primary interface (either the Layer 3 Ethernet interface or sub-interface) and associate the primary interface and dialer interface with the dialing pool.

2. Restrictions and Guidelines

- Only one dialing pool can be associated with a dialer interface.

- When the multi-dialing function is disabled, a primary interface is associated with a single dialing pool. However, when the multi-dialing function is enabled, a primary interface can be linked to multiple dialing pools.
- A dialing pool can only have one primary interface associated with it. When a dialer interface initiates dialing, the primary interface is selected from the dialing pool for dialing.
- When a Layer 3 Ethernet sub-interface is used as the primary interface, you need to configure the MAC address of the PPPoE session on the sub-interface for packet exchange.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the interface configuration mode.

- Enter the Layer 3 Ethernet interface configuration mode when the PPPoE client function is enabled on the Layer 3 Ethernet interface.

  interface *interface-type interface-number*

- Enter the Layer 3 Ethernet sub-interface configuration mode when the PPPoE client function is enabled on the Layer 3 Ethernet sub-interface.

  interface *interface-type interface-number.subnumber*

(4) Enable the PPPoE client function.

pppoe enable

The PPPoE client function is disabled by default.

(5) (Optional) Configure the MAC address of the PPPoE session.

pppoe session mac-address *mac-address*

When the PPPoE client function is enabled on a Layer 3 Ethernet sub-interface, you need to configure the MAC address of the PPPoE session.

By default, the MAC address of a PPPoE session is not configured.

(6) Add the interface to the dialing pool and specify the dialing mode.

pppoe-client dial-pool-number *pool-number* **no-ddr**

No interface joins any dialing pool by default.

1.4 Monitoring

Run the **show** command to check the configuration.

Run the **debug** command to output debugging information.

Caution

The output debugging information occupies system resources. Therefore, disable the debugging function immediately after use.

Run the **clear** command to clear the information.

Caution

Running the **clear** commands during device operation may cause service interruption due to the loss of important information.

Table 1-1 Monitoring the PPPoE Client

Command	Purpose
clear pppoe session	Clears PPPoE client information.
debug pppoe { **datas** | **errors** | **events** | **packets** }	Enables PPPoE session debugging.
show pppoe { **ref** | **session** }	Displays PPPoE status information.

1.5 Configuration Examples

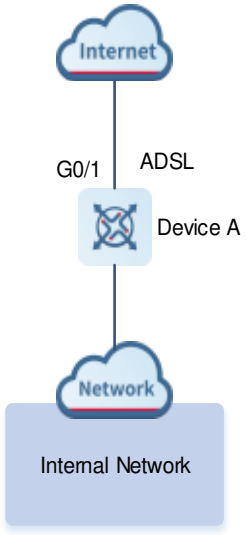
1.5.1 Configuring Automatic Dial-up of the PPPoE Client

1. Requirements

In the ADSL scenario, the PPPoE client function is enabled and the ADSL line is used for Internet access.

2. Topology

Figure 1-2 Configuring Basic Functions of the PPPoE Client



3. Note

- Enable the PPPoE client function on the primary interface of the device and add the primary interface to the dialing pool.
- Configure a dialer interface on the device, and configure encapsulation and negotiation parameters.
- Configure the route of the dialer interface.

4. Procedure

(1) Configure the dialer interface.

```
DeviceA(config)# interface dialer 1
DeviceA(config-if-dialer 1)# ip address negotiate
DeviceA(config-if-dialer 1)# mtu 1488
DeviceA(config-if-dialer 1)# ip nat outside
DeviceA(config-if-dialer 1)# dialer pool 1
DeviceA(config-if-dialer 1)# ppp chap hostname pppoe
DeviceA(config-if-dialer 1) # ppp chap password pppoe
DeviceA(config-if-dialer 1)# ppp pap sent-username pppoe password pppoe
DeviceA(config-if-dialer 1)# exit
```

(2) Enable the PPPoE client function on GigabitEthernet 0/1.

```
DeviceA(config)# interface GigabitEthernet 0/1
DeviceA(config-if-GigabitEthernet 0/1)# pppoe enable
DeviceA(config-if-GigabitEthernet 0/1)# pppoe-client dial-pool-number 1 no-ddr
DeviceA(config-if-GigabitEthernet 0/1)# exit
```

(3) Configure NAT and route information.

```
DeviceA(config)# access-list 1 permit 10.10.3.0 0.0.0.255
DeviceA(config)# ip nat inside source list 1 interface dialer 1
DeviceA(config)# ip route 0.0.0.0 0.0.0.0 dialer 1
DeviceA(config)# end
```

5. Verification

Run the **show ip interface brief | include dialer 1** command to check whether the dialer interface obtains an IP address.

```
DeviceA# show ip interface brief | include dialer 1
dialer 1                    49.1.1.127/32       YES      UP
```

Run the **show ip route** command to check the routing entry of the dialer interface.

```
DeviceA# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is 0.0.0.0 to network 0.0.0.0
S*   0.0.0.0/0 is directly connected, dialer 1
C    10.10.3.0/24 is directly connected, GigabitEthernet 0/0
C    10.10.3.1/32 is local host.
C    10.202.172.1/32 is directly connected, dialer 1
C    49.1.1.127/32 is local host.
```

6. Configuration Files

Device A configuration file

```
!
interface GigabitEthernet 0/1
no switchport
pppoe enable
 pppoe-client dial-pool-number 1 no-ddr
!
interface dialer 1
 ip address negotiate
 ip nat outside
 ppp chap hostname pppoe
 ppp chap password pppoe
 ppp pap sent-username pppoe password pppoe
 dialer pool 1
!
access-list 1 permit 10.10.3.0 0.0.0.255
!
ip nat inside source list 1 interface dialer 1
!
ip route 0.0.0.0 0.0.0.0 dialer 1 100
!
```

7. Common Misconfigurations

- Intranet hosts cannot access the Internet through dial-up.
  - Negotiation fails due to incorrect user name and password.
  - NAT configuration is incorrect.
  - Routing configuration is incorrect.

Contents

1 Configuring PKI

1.1 Introduction

1.1.1 Overview

Public Key Infrastructure (PKI) is a certificate management platform that uses the public key technology to provide network security services. It binds the identity of a person or entity to a public key using the digital certificate technology and issues certificates through a certificate authority (CA) to ensure the validity and security of certificate-holding entities.

A digital certificate is an electronic file issued by a CA. It contains entity identity information, public key information, and CA signature. A public key and a private key form a key pair in the public key cryptography system. Both communication parties verify the validity of the certificate through the CA signature in the digital certificate. They then compare the public key contained in the digital certificate with the digital signature generated based on the other party's private key to implement authentication.

The PKI feature can be used to implement certificate management for IP Security (IPSec) and Secure Sockets Layer (SSL).

1.1.2 Basic Concepts

1. CA

A CA is an authoritative, trustworthy, and fair third-party organization responsible for issuing and managing digital certificates for all entities involved in online transactions. The purpose of a CA is to manage keys, issue certificates to prove the validity of keys, and bind public keys with entities. A root CA is a CA at the top of the CA hierarchy.

2. Digital Certificate

A digital certificate is also known as a certificate. In this document, it refers to an X.509 certificate that binds an entity and its public key to identify the entity. A simple certificate contains a public key, name, and CA digital signature. A typical certificate also contains the validity period of the key, name of the license issuing authority (LIA), and certificate serial number. The certificate format complies with the ITUT X.509 international standard. The CA root certificate is a self-signed certificate issued by the root CA. The root certificate is used to sign other certificates issued by the CA.

3. Privacy-enhanced Mail (PEM)

PEM is a Base64-encoded text format defined in RFC 1421 to RFC 1424, which is commonly used in emails and certificate import and export.

4. Public Key Cryptography Standards (PKCS)

PKCS is a set of file formats based on public key encryption defined by RSA Laboratories in cooperation with multiple security system developers, industry, academia, and government representatives.

- PKCS#1: Defines the RSA encryption and signature algorithms.
- PKCS#7: Defines a syntax for encrypting information.
- PKCS#12: Defines the method of packing a security package. PKCS12 contains several security packages,

such as a certificate and a private key. It is a commonly used format for issuing certificates.

Files output based on the PKCS standard are DER-encoded binary files, which are sometimes converted to PEM-encoded text files.

5. Simple Certificate Enrollment Protocol (SCEP)

SCEP is a part of the PKI protocol system and is a certificate acquisition protocol that ensures certificate security and reliability. SCEP-based digital certificate acquisition has the following advantages:

- The signature private key does not leave the device, ensuring higher security.
- The SCEP uses the PKCS7 digital envelope to ensure the security of communication.
- If the CA supports automatic certificate update, SCEP can automatically update certificates.

6. Certificate Revocation List (CRL)

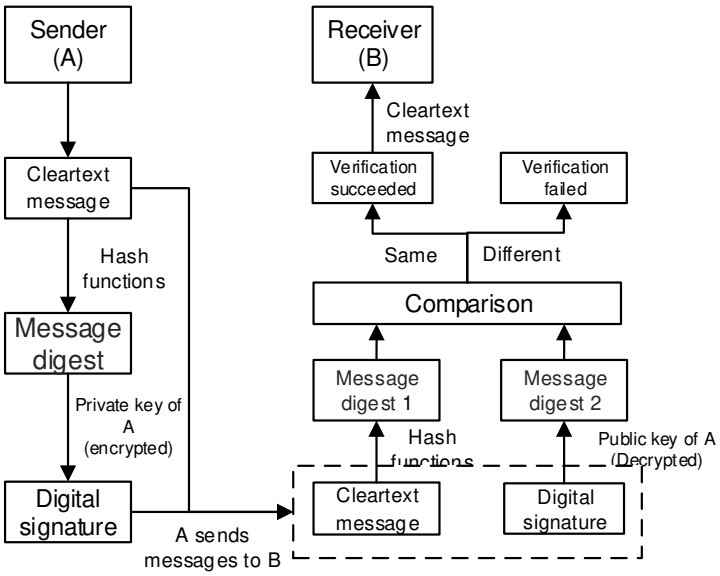
A CRL is a timestamped list of certificates that have been revoked by a CA and can be obtained freely from a public storage repository. Each certificate in the CRL is identified by its serial number. The two communication parties can query and compare the recently released CRLs to verify the validity of the certificates. In PKI, when a certificate needs to be revoked due to a change in the user name or service suspension, the certificate revocation information can be published through a CRL.

1.1.3 Principles

1. Digital Signature

Digital signature is a technology used to verify the integrity and authenticity of data. It ensures that data is not tampered with during transmission, and can confirm the identity of the sender. Asymmetric encryption algorithms can be used to generate digital signatures. The implementation process is shown in Figure 1-1.

Figure 1-1 Digital Signature Process Using Asymmetric Encryption Algorithm



The process and principles are as follows:

1. Generate a key pair. The sender of a digital signature needs to generate a pair of asymmetric keys, including a private key and a corresponding public key. The private key is used to sign data, while the public key is used to verify the signature.

(2) Generate message digest. The sender uses the hash algorithm to calculate the message digest of the message to be sent.

(3) The sender generates a digital signature. The sender uses its private key to encrypt the message digest, which is the signature process. Common encryption algorithms include RSA, DSA, and ECDSA.

(4) Send messages and digital signatures. The sender sends the generated digital signature and the original cleartext message to the receiver.

(5) The receiver verifies the data.

a Upon receiving the message and the digital signature, the receiver decrypts the digital signature using the sender's public key to obtain the original message digest.

b The receiver uses the hash algorithm to calculate the digest of the original cleartext message, obtaining a message digest.

c The receiver uses the public key to decrypt the message digest and compares it with the message digest calculated using the hash algorithm. If the two message digests are the same, the verification succeeds. Otherwise, the verification fails.

Through this process, the receiver can confirm the integrity and authenticity of the data because only the sender has the corresponding private key to generate the correct digital signature, and the public key can be used by anyone to verify the authenticity of the digital signature.

2. Public Key Exchange

Public key exchange usually involves the application of PKI and digital certificates. The basic process and principles of certificate-based secure public key exchange are as follows:

(1) Obtain a digital certificate. The two communication parties need to obtain their own digital certificates, which are typically applied from a CA. When applying for a certificate, an applicant need to provide some personal or organizational identity information so that the CA can authenticate the applicant and issue a digital certificate.

(2) Exchange digital certificates. Before establishing secure communication, the two parties need to exchange their digital certificates. Typically, the digital certificate is sent to the peer device at the initial communication stage in a secure manner, such as through a secure network connection or physical medium.

(3) Extract the public key. After receiving the digital certificate from the peer device, each communication entity can extract the peer device's public key from the digital certificate.

(4) Verify the public key. Before using the public key of the peer device for encryption or digital signature verification, you need to verify the digital certificate of the peer device, including the validity period of the certificate, the validity of the issuer's signature, and whether the certificate is revoked.

(5) Exchange the public key. After the digital certificate of the peer device is verified, the two devices can use the public key of the peer device to encrypt communication or verify digital signatures.

With the preceding steps, the two communication parties can obtain each other's public keys securely and ensure the authenticity of the other party's identity and certificate. This method provides a secure mechanism for exchange of public keys through digital certificates and PKI, ensuring the security and reliability of communication.

1.1.4 Protocols and Standards

- RFC5280: Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
- RFC 4210: Public Key Infrastructure Certificate Management

1.2 Configuration Task Summary

PKI configuration includes the following tasks:

(1) Configuring Basic

a Configuring the Certificate Enrollment Type

b Importing a Certificate

c Exporting a Certificate

d Obtaining a Certificate Through SCEP

e Obtaining a Certificate in Offline Mode

f Obtaining the CRL File

(2) (Optional)Using the SM2 Digital Certificate

(3) (Optional)Configuring the Certificate Update Function

(4) (Optional)Disabling Self-signed Certificate Verification of the CA Root Certificate

(5) (Optional)Disabling the Certificate Validity Period Check

(6) (Optional)Configuring the Interface for Interacting with the CA Server

1.3 Configuring Basic Features

1.3.1 Overview

PKI is a system used to create, manage, and issue digital certificates. When performing secure communication or connection, for example, SSL/TLS connection or VPN connection, users can use PKI for authentication and communication encryption to ensure the security and integrity of communication. PKI can be used to generate and verify digital signatures for files or data that require digital signatures. PKI not only provides a secure authentication mechanism to ensure that the identities of the communicating parties are valid, but also implement encrypted communication to protect data from being stolen or tampered with during transmission. Digital signatures ensure the integrity of files and the credibility of their sources. Configuring PKI enhances network security, but may increase the computation and communication overheads. Therefore, when configuring the PKI function, you are advised to set the certificate expiry date, key length, and encryption algorithm properly, and periodically update and maintain the certificate.

1.3.2 Configuration Tasks

The PKI basic function configuration includes the following tasks:

(1) Configuring the Certificate Enrollment Type

(2) Importing a Certificate

(3) Exporting a Certificate

(4) Obtaining a Certificate Through SCEP

(5) Obtaining a Certificate in Offline Mode

(6) Obtaining the CRL File

1.3.3 Configuring the Certificate Enrollment Type

1. Overview

You can run this command to configure the enrollment type for obtaining a certificate. The certificate enrollment type cannot be modified after configuration.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Configure the certificate enrollment type.

enrollment type { **import** | **offline** | **scep** }

No certificate enrollment type is configured by default.

1.3.4 Importing a Certificate

1. Overview

When a digital certificate is imported, the validity period of the certificate will be checked. The expired or not-yet-valid certificate cannot be imported.

2. Restrictions and Guidelines

- The device system time must be in the time zone of the CA.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Import a digital certificate file in PEM format.

crypto pki import *trustpoint_name* **pem terminal** *password*

(4) Import only the CA root certificate.

crypto pki import *trustpoint_name* **ca**

(5) Import a digital certificate in PKCS12 format.

crypto pki import *trustpoint_name* **pkcs12** { **flash**:*cert_path* | **tftp**:*tftp_url* } [*password*]

(6) Import the digital certificate of the peer device.

crypto pki certificate peer address *ipv4_address*

No digital certificate of the peer device is imported by default.

(7) Import the digital certificate of a trustpoint to the device.

crypto pki certificate chain *trustpoint-name*

No digital certificate is configured for any trustpoint by default.

1.3.5 Exporting a Certificate

1. Overview

This function is used to configure exporting digital certificate in PEM format, including the CA digital certificate and device digital certificate.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Export the digital certificate configuration of a specified trustpoint.

crypto pki export *trustpoint_name* **pem terminal**

1.3.6 Obtaining a Certificate Through SCEP

1. Overview

This function is used to trigger the device to obtain the digital certificate from the CA. The digital certificate is saved in the device storage. You do not need to obtain the digital certificate again in the case of a device restart.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure a trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Configure the URL of the certificate.

enrollment url *url_string* [**auto-up**]

No URL is configured for obtaining a device certificate through SCEP by default.

(5) (Optional) Configure the number of retries for obtaining a device certificate using SCEP in the case of failure.

enrollment retry count *number*

The number of retries for obtaining the device certificate is 60 by default.

(6) (Optional) Configure the polling interval for obtaining a certificate through SCEP.

enrollment retry period *number*

The default interval for obtaining a device certificate through SCEP is 1 second.

(7) (Optional) Configure a distinguishable name for the local device.

subject-name [*sub_name*]

No distinguishable name is configured for the local device by default.

(8) Configure the device to generate a self-signed certificate.

enrollment selfsigned

The device is not configured to generate a self-signed certificate by default.

(9) Exit the trustpoint configuration mode.

exit

(10) Obtain the CA root certificate.

crypto pki authenticate *trustpoint_name*

The CRL distribution point address in the CA certificate or device certificate is used to obtain the CA root certificate by default.

(11) Perform device certificate enrollment.

crypto pki enroll *ca_name*

No digital certificate is enrolled for the trustpoint by default.

1.3.7 Obtaining a Certificate in Offline Mode

1. Overview

The digital certificate can be obtained when no network connection is available.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure a trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Configure a distinguishable name for the local device in offline mode.

enrollment offline subject

No distinguishable name is configured for the local device in offline mode by default.

(5) Configure the add-on option of the certificate.

enrollment extend { **authenticate** | **enroll** } *extend_string*

The add-on option for downloading the root certificate or device certificate is not configured by default.

(6) Exit the trustpoint configuration mode.

exit

(7) Perform device certificate enrollment.

crypto pki enroll *ca_name*

No digital certificate is enrolled for the trustpoint by default.

1.3.8 Obtaining the CRL File

1. Overview

This function is used to obtain the CRL of PKI to list revoked certificates.

2. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure a trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Configure the address for downloading the CRL.

crl query *url_string*

The CRL distribution point address in the CA certificate or device certificate is used by default.

(5) (Optional) Disable CRL verification.

revocation-check none

The self-signing check function of the CA root certificate is enabled by default.

(6) Exit the trustpoint configuration mode.

exit

(7) Import a CRL file manually.

crypto pki import *trustpoint_name* **crl** { **flash:***crl_path* | **tftp:**//*tftp_url* }

(8) Download the CRL file manually.

crypto pki crl request *trustpoint_name*

The trustpoint is not imported to the CRL by default.

1.4 Using the SM2 Digital Certificate

1.4.1 Overview

This function is used to configure using the SM2 digital certificate. The SM2 certificate can be used only for digital envelope V2.

1.4.2 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure a trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Configure the trustpoint to use the SM2 digital certificate.

asymmetric sm2

The RSA digital certificate is used by default.

1.5 Configuring the Certificate Update Function

1.5.1 Overview

This function is used to configure automatic certificate update.

1.5.2 Restrictions and Guidelines

The certificate auto-update function is supported only when the certificate enrollment type is SCEP.

1.5.3 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Configure the CA server to support certificate update.

enrollment renewable

(5) Configure the certificate update interval.

enrollment auto-enroll *percentage*

The certificate update function is disabled on the CA server by default.

1.6 Disabling Self-signed Certificate Verification of the CA Root Certificate

1.6.1 Overview

This function is used to disable the self-signed certificate verification for the CA root certificate. Typically, the CA root certificate is self-signed. Disabling self-signed certification verification means that the self-signed CA root certificate is not verified, and the validity of the CA root certificate is trusted by default. It should be noted that, in actual scenarios, disabling the self-signed certificate verification for the CA root certificate may compromise security. Therefore, exercise caution when configuring this function. You are advised to configure this function only when there are clear security requirements and proper risk assessment.

1.6.2 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Disable self-signed certificate verification for the CA root certificate.

recursion-check none

The self-signed certification verification function of the CA root certificate is enabled by default.

1.7 Disabling the Certificate Validity Period Check

1.7.1 Overview

This function is used to disable the certificate validity period check function. Certificate validity period check is performed to ensure that the certificate is valid within the specified time range. Disabling the certificate validity period check function means that the system no longer verifies the expiry date of the certificate, which may be because the certificate has been revoked or expired. However, the system still accepts the security verification of the certificate. In this case, you can configure this function to disable the certificate validity period check. Disabling the certificate validity period check may cause security risks because expired or revoked certificates may be maliciously exploited. Therefore, it is recommended that the certificate validity period check function be enabled in the PKI system to ensure the security and validity of the certificate.

1.7.2 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Disable the certificate validity period check.

time-check none

The certificate validity period check function is enabled by default.

1.8 Configuring the Interface for Interacting with the CA Server

1.8.1 Overview

This function is used to configure the CA server to communicate with other systems or entities. These functions include certificate issuance, certificate update, certificate revocation, key management, and certificate verification. These functions enable the PKI CA server to effectively manage digital certificates and keys, ensuring secure communication and authentication.

1.8.2 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the trustpoint and enter the trustpoint configuration mode.

crypto pki trustpoint *trustpoint_name*

(4) Configure the interface for interacting with the CA server.

source interface *interface-type interface-name*

No interface is specified for interacting with the CA server by default.

1.9 Monitoring

Run the **show** command to check the configuration.

Run the **debug** command to output debugging information

Note

Debugging occupies system resources, so disable it immediately if not required.

Table 1-1 PKI Monitoring

Command	Purpose
show crypto pki certificates [*trustpoint_name* [**detail**]]	Displays the current certificate information.
show crypto pki crls [*trustpoint_name* [**detail**]]	Displays the CRL.
show crypto pki trustpoints [*trustpoint_name*]	Displays the configuration of the system trustpoint.
debug crypto pki error	Enables the PKI error commission switch.
debug crypto pki event	Enable the PKI event commission switch.

1.10 Configuration Examples

1.10.1 Obtaining a Certificate Through SCEP

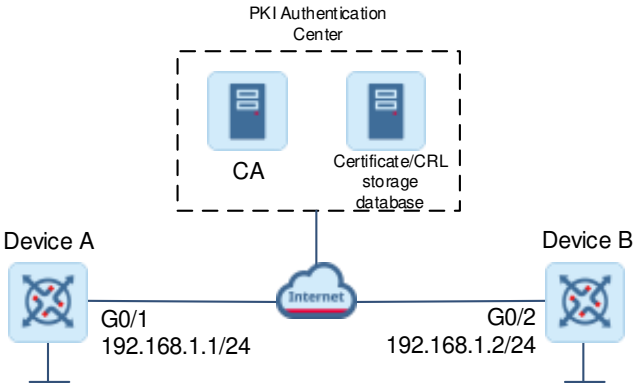
1. Requirements

Device A and Device B are directly connected and both devices are connected to the PKI authentication center.

Device A is configured to apply for a local certificate from the CA server on the public network using SCEP. After successful application, the CA digital certificate is saved in the device storage.

2. Topology

Figure 1-2 Obtaining a Certificate Through SCEP



3. Notes

- Configure the URL of the certificate.
- Configure Device A to obtain a certificate through SCEP.

4. Procedure

(1) Configure the URL of the certificate.

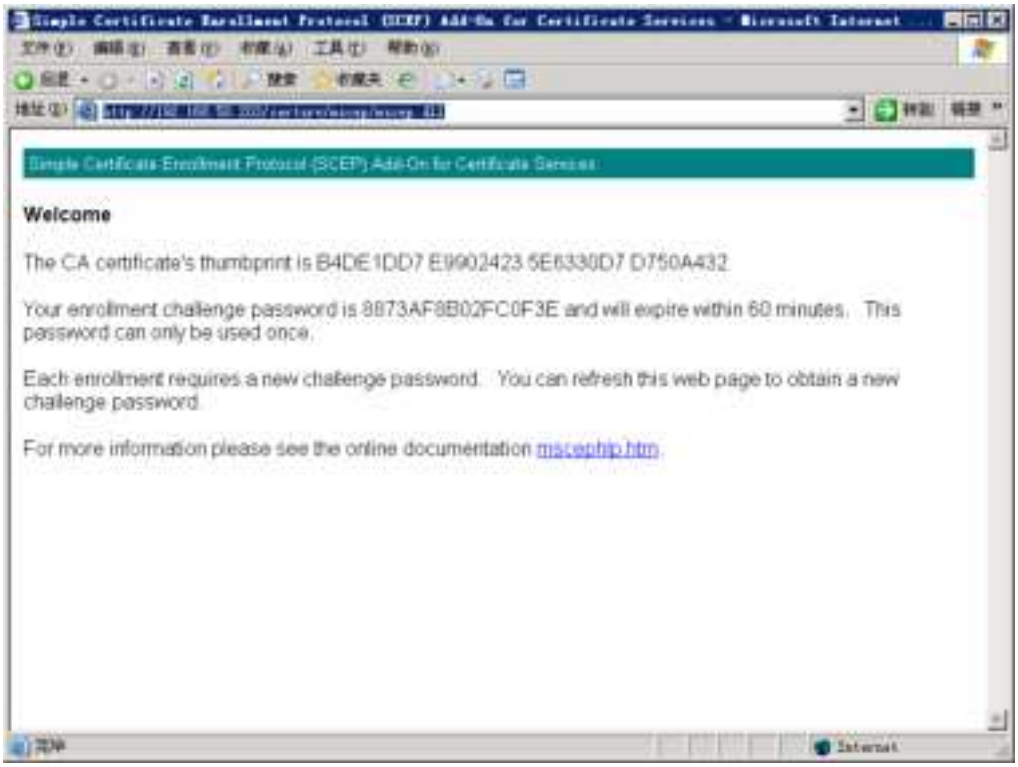
```
DeviceA> enable
DeviceA# configure terminal
DeviceA(config)# crypto pki trustpoint CA
DeviceA(ca-trustpoint)# enrollment url
http://192.168.50.203/certsrv/mscep/mscep.dll
```

(2) Configure Device A to obtain and authenticate the CA root certificate.

```
DeviceA(config)# crypto pki authenticate CA
Certificate has the following attributes:
MD5 fingerprint: B4DE1DD7 E9902423 5E6330D7 D750A432
SHA1 fingerprint: AD070162 672A7C57 BD5EE522 A95AAFA1 351524D0
% Do you accept this certificate?[yes/no]:yes
```

(3) To obtain the certificate fingerprint and challenge password, visit http://ca-ip-address/certsrv/mscep/mscep.dll Enter the authentication code on the website, as shown in Figure 1-3. Ensure that administrator credentials are provided to access this site.

Figure 1-3 Obtaining the Certificate Fingerprint and Challenge Password



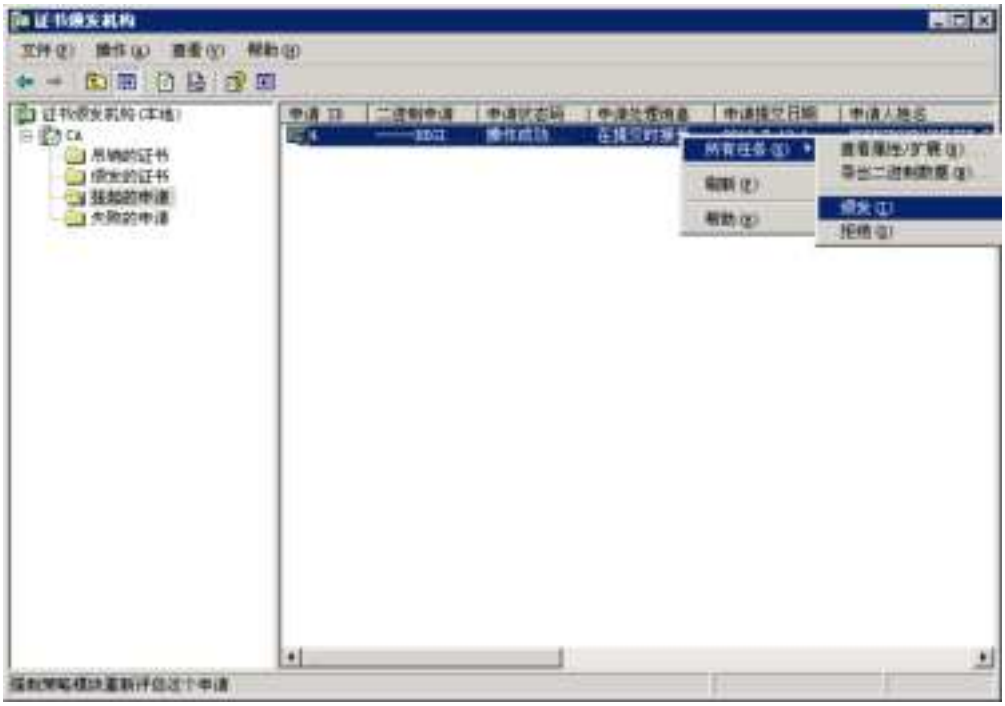
(4) Register the device certificate on Device A.

```
DeviceA(config)# crypto pki enroll CA
%
%Start certificate enrollment ..
%Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

Password:F4EEE4FEB3766007     //Enter the challenge password obtained from the CA.
Re-enter password:F4EEE4FEB3766007
%The subject name in the certificate will include: router
```

(5) Issue a certificate. On the CA, choose **Pending Requests** from the list, click **All Tasks**, and then click **Issue**, as shown inFigure 1-4.

Figure 1-4 Configuring Issuing a Certificate



5. Verification

Check the certificate information configured on Device A.

```
DeviceA# show crypto pki certificates
% CA certificate info:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            77:e0:c4:e3:2f:6e:29:bc:45:bc:8f:89:5a:15:af:47
        Issuer: CN=vpnca
        Validity
            Not Before: Feb 13 03:12:39 2019 GMT
            Not After : Feb 13 03:22:15 2024 GMT
        Subject: CN=vpnca
Associated Trustpoints: sm2

% Router certificate info:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            61:03:0d:7e:00:00:00:00:00:6f
        Issuer: CN=vpnca
        Validity
            Not Before: Mar 29 02:17:47 2019 GMT
```

```
          Not After : Mar 29 02:27:47 2020 GMT
       Subject: C=CN, ST=fj, L=fz, O=rj, OU=test, CN=rj
Associated Trustpoints: sm2

% CA certificate info:
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number:
          77:e0:c4:e3:2f:6e:29:bc:45:bc:8f:89:5a:15:af:47
       Issuer: CN=vpnca
       Validity
          Not Before: Feb 13 03:12:39 2019 GMT
          Not After : Feb 13 03:22:15 2024 GMT
       Subject: CN=vpnca
Associated Trustpoints: rsa

% Router certificate info:
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number:
          11:38:cf:f4:00:00:00:00:00:80
       Issuer: CN=vpnca
       Validity
          Not Before: Apr  9 12:57:20 2019 GMT
          Not After : Apr  9 13:07:20 2020 GMT
       Subject: C=CN, ST=fj, L=fz, O=rj, OU=test, CN=rj
Associated Trustpoints: rsa

% CA certificate info:
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number:
          55:b8:3b:79:72:15:b1:9e:40:14:87:69:61:a6:dd:b6
       Issuer: CN=vpnca
       Validity
          Not Before: Apr 15 11:39:59 2019 GMT
          Not After : Apr 15 11:49:28 2024 GMT
       Subject: CN=vpnca
Associated Trustpoints: test

% Router certificate info:
Certificate:
   Data:
```

```
      Version: 3 (0x2)
      Serial Number:
         61:5b:1f:16:00:00:00:00:00:09
      Issuer: CN=vpnca
      Validity
         Not Before: Apr 15 13:04:40 2019 GMT
         Not After : Apr 15 13:14:40 2020 GMT
      Subject: unstructuredName=Ruijie
Associated Trustpoints: test

% CA certificate info:
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number:
         02:37:fc:7b:d9:be:f0:b1:44:4e:14:98:a5:12:e4:31
      Issuer: CN=vpnca
      Validity
         Not Before: Apr 28 01:47:27 2019 GMT
         Not After : Apr 28 01:56:24 2029 GMT
      Subject: CN=vpnca
Associated Trustpoints: testcore
```

Check the CRL information on Device A.

```
DeviceA# show crypto pki crls
Certificate Revocation List (CRL):
Version 2 (0x1)
Signature Algorithm: sha1WithRSAEncryption
Issuer: /emailAddress=wlcpyjwb@star-net.cn/C=CN/ST=fj/L=fuzhou/O=Red
Giant/OU=Department 5/CN=CA Server
Last Update: Jun 22 06:10:27 2005 GMT
Next Update: Jun 29 18:30:27 2005 GMT
CRL extensions:
X509v3 Authority Key Identifier:
keyid:64:46:12:C0:27:A4:9E:01:0C:65:DA:F8:6E:E7:FE:C6:56:EC:AD:D4
1.3.6.1.4.1.311.21.1:...
Revoked Certificates:
Serial Number: 162A7A1D000000000002
Revocation Date: Jun 22 06:19:53 2005 GMT
CRL entry extensions:
X509v3 CRL Reason Code:
Key Compromise
Serial Number: 1635E5E3000000000003
Revocation Date: Jun 22 06:19:53 2005 GMT
CRL entry extensions:
X509v3 CRL Reason Code:
Key Compromise
```

```
Signature Algorithm: sha1WithRSAEncryption
5d:a2:ab:07:ff:7e:0e:9a:af:b2:25:11:7f:31:86:aa:21:48:
37:e7:22:99:e3:b2:15:e0:f9:80:63:66:5e:2f:f2:d6:c0:ea:
ef:46:7e:d1:c1:b2:66:0e:0b:d3:74:d1:55:bc:5c:13:46:e8:
56:ec:40:83:7b:1b:75:f2:68:87
```

Check the configuration of the trustpoint on Device A.

```
DeviceA# show crypto pki trustpoints
Trustpoint rsa
    Subject Name: cn=rj,ou=test,o=rj,l=fz,st=fj,c=CN
    Certificate configured.
  enrollment url http://192.168.50.203/certsrv/mscep/mscep.dll
  enrollment extend authenticate:mess
  enrollment retry period 1
  enrollment retry count 60
  renew percentage:90
```

6. Configuration Files

- Device A configuration file

```
!
crypto pki trustpoint CA
 enrollment url http://192.168.50.203/certsrv/mscep/mscep.dll
!
crypto pki authenticate CA
crypto pki enroll CA
!
```

7. Common Errors

N/A