Implementing MPLS Layer 3 VPNs

A Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) consists of a set of sites that are interconnected by means of an MPLS provider core network. At each customer site, one or more customer edge (CE) routers attach to one or more provider edge (PE) routers.

This module provides the conceptual and configuration information for MPLS Layer 3 VPNs on Cisco NCS 5500 Series Routers.

Note You must acquire an evaluation or permanent license in order to use MPLS Layer 3 VPN functionality. For more information about licenses, see the module in the *System Management Configuration Guide for Cisco NCS 5500 Series Routers*.

For a complete description of the commands listed in this module, refer these command references:

- BGP
- MPLS
- Routing
- VPN and Ethernet Services

This chapter includes topics on:

MPLS L3VPN Overview

Before defining an MPLS VPN, VPN in general must be defined. A VPN is:

- An IP-based network delivering private network services over a public infrastructure
- A set of sites that are allowed to communicate with each other privately over the Internet or other public or private networks

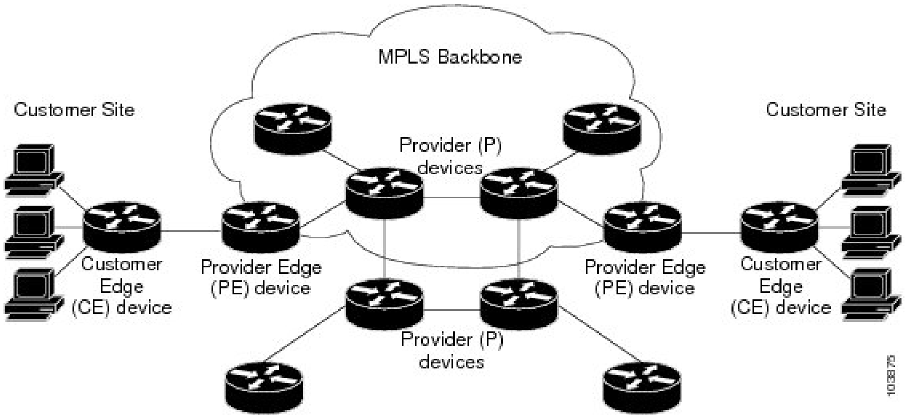
Conventional VPNs are created by configuring a full mesh of tunnels or permanent virtual circuits (PVCs) to all sites in a VPN. This type of VPN is not easy to maintain or expand, as adding a new site requires changing each edge device in the VPN.

MPLS-based VPNs are created in Layer 3 and are based on the peer model. The peer model enables the service provider and the customer to exchange Layer 3 routing information. The service provider relays the data between the customer sites without customer involvement.

MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the edge router of the service provider that provides services to the customer site needs to be updated.

The following figure depicts a basic MPLS VPN topology.

Figure 1: Basic MPLS VPN Topology


MPLS Backbone
Customer Site
Customer Site
Provider (P) devices
Customer Edge (CE) device
Provider Edge (PE) device
Provider Edge (PE) device
Customer Edge (CE) device
Provider (P) devices
103875

These are the basic components of MPLS VPN:

- Provider (P) router—Router in the core of the provider network. P routers run MPLS switching and do not attach VPN labels to routed packets. VPN labels are used to direct data packets to the correct private network or customer edge router.
- PE router—Router that attaches the VPN label to incoming packets based on the interface or sub-interface on which they are received, and also attaches the MPLS core labels. A PE router attaches directly to a CE router.
- Customer (C) router—Router in the Internet service provider (ISP) or enterprise network.
- Customer edge (CE) router—Edge router on the network of the ISP that connects to the PE router on the network. A CE router must interface with a PE router.

How MPLS L3VPN Works

MPLS VPN functionality is enabled at the edge of an MPLS network. The PE router performs the following tasks:

- Exchanges routing updates with the CE router
- Translates the CE routing information into VPN version 4 (VPNv4) routes
- Exchanges VPNv4 routes with other PE routers through the Multiprotocol Border Gateway Protocol (MP-BGP)

Major Components of MPLS L3VPN

An MPLS-based VPN network has three major components:

- VPN route target communities—A VPN route target community is a list of all members of a VPN community. VPN route targets need to be configured for each VPN community member.
- Multiprotocol BGP (MP-BGP) peering of the VPN community PE routers—MP-BGP propagates VRF reachability information to all members of a VPN community. MP-BGP peering needs to be configured in all PE routers within a VPN community.
- MPLS forwarding—MPLS transports all traffic between all VPN community members across a VPN service-provider network.

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A given site can be a member of multiple VPNs. However, a site can associate with only one VRF. A customer-site VRF contains all the routes available to the site from the VPNs of which it is a member.

Read more at Major Components of MPLS L3VPN—Details, on page 80.

Restrictions for MPLS L3VPN

Implementing MPLS L3VPN in Cisco NCS 5500 Series Routers is subjected to these restrictions:

- L3VPN prefix lookup always yields a single path. In case of multiple paths at IGP or BGP level, path selection at each level is done using the prefix hash in control plane. The selected path is programmed in the data plane.
- L3VPN over Generic Routing Encapsulation (GRE) is not supported.
- BGP-Prefix Independent Convergence (PIC) is not supported for Layer 3 VPN routes learnt over BGP-LU.
- PIC over RSVP-TE is not supported.
- When paths of different technologies are resolved over ECMP, it results in *heterogeneous* ECMP, leading to severe network traffic issues. Don’t use ECMP for any combination of the following technologies:
  - LDP
  - BGP-LU, including services over BGP-LU loopback peering or recursive services at Level-3
  - VPNv4

- 6PE and 6VPE
- EVPN
- Recursive static routing

Apart from the specific ones mentioned above, these generic restrictions for implementing MPLS L3VPNs also apply for Cisco NCS 5500 Series Routers:

The following restrictions apply when configuring MPLS VPN Inter-AS with ASBRs exchanging IPv4 routes and MPLS labels:

- For networks configured with eBGP multihop, a label switched path (LSP) must be configured between non adjacent routers.
- Layer 3 VPN over SR-TE is not supported.

Note The physical interfaces that connect the BGP speakers must support FIB and MPLS.

Hardware Module Profiles

Hardware module profile is used to modify router resources for the specific needs during the router boot up time. You can configure the hardware module profile or you can view the default profile.

The following table describes the hardware module profile commands:

Table 1: Hardware Module Commands

Hardware Module Commands	Description	Supported Platforms
hw-module fib mpls label lsr-optimized	Use this command to store the outgoing MPLS label with a prefix in largest exact match (LEM) memory in the hardware. For host routes with /32 IPv4 prefixes, this optimization saves the following hardware resources: • One Egress Encapsulation Data Base (EEDB) entry. • One regular Forward Equivalence Class (FEC) per ECMP path per prefix. • One ECMP FEC per prefix, as all the prefixes share the same set of ECMP path point to one shared ECMP FEC. The command is used for LSR roles. **Note** Layer 3 VPN services do not work when the command is configured.	• NCS 5500 fixed port routers • NCS 5700 fixed port routers • NCS 5500 modular routers • NCS 5500 line cards • NCS 5700 line cards [Mode: Compatibility]
hw-module fib mpls bgp-sr lsr-optimized	Use this command to optimize the ECMP FEC resources for BGP SR prefixes when the out label is the same for all the LU paths, by pushing the label into the leaf. **Note** This command cannot co-exist with the **hw-module fib mpls label lsr-optimized** command.	• NCS 5500 fixed port routers • NCS 5700 fixed port routers • NCS 5500 modular routers • NCS 5500 line cards • NCS 5700 line cards [Mode: Compatibility]
hw-module fib mpls ldp lsr-optimized	Enables the Push or Swap shared MPLS encapsulation, which can be used for label push or label swap. If the label consists of IPv4 packets, then it is pushed and if it consists if MPLS packets, then it is swapped. **Note** The optimization does not work on Layer 2, Layer 3, and EVPN services.	• NCS 5500 fixed port routers • NCS 5700 fixed port routers • NCS 5500 modular routers • NCS 5500 line cards • NCS 5700 line cards [Mode: Compatibility]

Hardware Module Commands	Description	Supported Platforms
hw-module fib recycle service-over-rsvpte	Use this command to support the LU services on LDP over RSVP-TE. **Note** Bandwidth is limited as the command uses the recycle approach.	• NCS 5500 fixed port routers • NCS 5500 modular routers • NCS 5500 line cards
hw-module fib bgp-mp-pic auto-protect	Use this command to enable the BGP MP PIC loop-back peering auto protection. By default, the BGP MP PIC loop-back peering auto protection is disabled.	• NCS 5500 fixed port routers • NCS 5700 fixed port routers • NCS 5500 modular routers • NCS 5500 line cards • NCS 5700 line cards [Mode: Compatibility; Native]
hw-module fib bgp-pic multipath-core enable	Use this command to save ECMP FEC resources by enabling the BGP PIC multipath core and BGP PIC multipath edge interface peering.	• NCS 5500 fixed port routers • NCS 5700 fixed port routers • NCS 5500 modular routers • NCS 5500 line cards • NCS 5700 line cards [Mode: Compatibility; Native]

Inter-AS Support for L3VPN

This section contains the following topics:

Inter-AS Support: Overview

An autonomous system (AS) is a single network or group of networks that is controlled by a common system administration group and uses a single, clearly defined routing protocol.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. In addition, some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of the complexity and location of the VPNs, the connection between autonomous systems must be seamless.

An MPLS VPN Inter-AS provides the following benefits:

- Allows a VPN to cross more than one service provider backbone.

Service providers, running separate autonomous systems, can jointly offer MPLS VPN services to the same end customer. A VPN can begin at one customer site and traverse different VPN service provider backbones before arriving at another site of the same customer. Previously, MPLS VPN could traverse only a single BGP autonomous system service provider backbone. This feature lets multiple autonomous systems form a continuous, seamless network between customer sites of a service provider.

- Allows a VPN to exist in different areas.

  A service provider can create a VPN in different geographic areas. Having all VPN traffic flow through one point (between the areas) allows for better rate control of network traffic between the areas.

- Allows confederations to optimize iBGP meshing.

  Internal Border Gateway Protocol (iBGP) meshing in an autonomous system is more organized and manageable. You can divide an autonomous system into multiple, separate subautonomous systems and then classify them into a single confederation. This capability lets a service provider offer MPLS VPNs across the confederation, as it supports the exchange of labeled VPN-IPv4/IPv6 Network Layer Reachability Information (NLRI) between the subautonomous systems that form the confederation.

Inter-AS and ASBRs

Separate autonomous systems from different service providers can communicate by exchanging IPv4 NLRI and IPv6 in the form of VPN-IPv4/IPv6 addresses. The ASBRs use eBGP to exchange that information. Then an Interior Gateway Protocol (IGP) distributes the network layer information for VPN-IPV4/IPv6 prefixes throughout each VPN and each autonomous system. The following protocols are used for sharing routing information:

- Within an autonomous system, routing information is shared using an IGP.

- Between autonomous systems, routing information is shared using an eBGP. An eBGP lets service providers set up an interdomain routing system that guarantees the loop-free exchange of routing information between separate autonomous systems.

  The primary function of an eBGP is to exchange network reachability information between autonomous systems, including information about the list of autonomous system routes. The autonomous systems use EBGP border edge routers to distribute the routes, which include label switching information. Each border edge router rewrites the next-hop and MPLS labels.

  Inter-AS configurations supported in an MPLS VPN can include:

  - Interprovider VPN—MPLS VPNs that include two or more autonomous systems, connected by separate border edge routers. The autonomous systems exchange routes using eBGP. No IGP or routing information is exchanged between the autonomous systems.

  - BGP Confederations—MPLS VPNs that divide a single autonomous system into multiple subautonomous systems and classify them as a single, designated confederation. The network recognizes the confederation as a single autonomous system. The peers in the different autonomous systems communicate over eBGP sessions; however, they can exchange route information as if they were iBGP peers.

Confederations

A confederation is multiple subautonomous systems grouped together. A confederation reduces the total number of peer devices in an autonomous system. A confederation divides an autonomous system into subautonomous systems and assigns a confederation identifier to the autonomous systems. A VPN can span service providers running in separate autonomous systems or multiple subautonomous systems that form a confederation.

In a confederation, each subautonomous system is fully meshed with other subautonomous systems. The subautonomous systems communicate using an IGP, such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Each subautonomous system also has an eBGP connection to the other subautonomous systems. The confederation eBGP (CEBGP) border edge routers forward next-hop-self addresses between the specified subautonomous systems. The next-hop-self address forces the BGP to use a specified address as the next hop rather than letting the protocol choose the next hop.

You can configure a confederation with separate subautonomous systems two ways:

- Configure a router to forward next-hop-self addresses between only the CEBGP border edge routers (both directions). The subautonomous systems (iBGP peers) at the subautonomous system border do not forward the next-hop-self address. Each subautonomous system runs as a single IGP domain. However, the CEBGP border edge router addresses are known in the IGP domains.

- Configure a router to forward next-hop-self addresses between the CEBGP border edge routers (both directions) and within the iBGP peers at the subautonomous system border. Each subautonomous system runs as a single IGP domain but also forwards next-hop-self addresses between the PE routers in the domain. The CEBGP border edge router addresses are known in the IGP domains.

Note eBGP Connection Between Two Subautonomous Systems in a Confederation figure illustrates how two autonomous systems exchange routes and forward packets. Subautonomous systems in a confederation use a similar method of exchanging routes and forwarding packets.

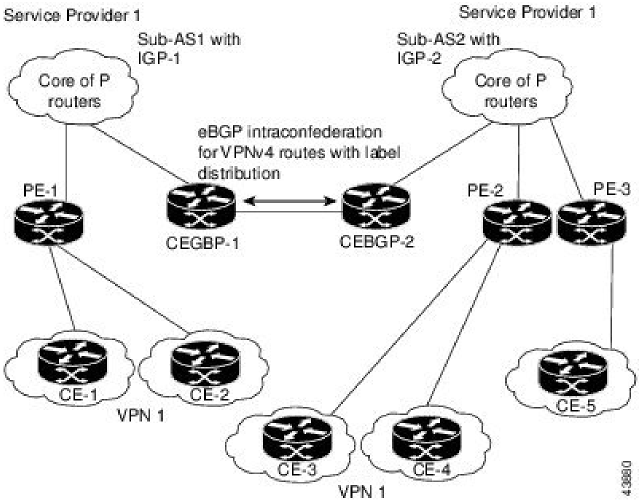
The figure below illustrates a typical MPLS VPN confederation configuration. In this configuration:

- The two CEBGP border edge routers exchange VPN-IPv4 addresses with labels between the two autonomous systems.

- The distributing router changes the next-hop addresses and labels and uses a next-hop-self address.

- IGP-1 and IGP-2 know the addresses of CEBGP-1 and CEBGP-2.

Figure 2: eBGP Connection Between Two Subautonomous Systems in a Confederation



In this confederation configuration:

- CEBGP border edge routers function as neighboring peers between the subautonomous systems. The subautonomous systems use eBGP to exchange route information.
- Each CEBGP border edge router (CEBGP-1 and CEBGP-2) assigns a label for the router before distributing the route to the next subautonomous system. The CEBGP border edge router distributes the route as a VPN-IPv4 address by using the multiprotocol extensions of BGP. The label and the VPN identifier are encoded as part of the NLRI.
- Each PE and CEBGP border edge router assigns its own label to each VPN-IPv4 address prefix before redistributing the routes. The CEBGP border edge routers exchange IPV-IPv4 addresses with the labels. The next-hop-self address is included in the label (as the value of the eBGP next-hop attribute). Within the subautonomous systems, the CEBGP border edge router address is distributed throughout the iBGP neighbors, and the two CEBGP border edge routers are known to both confederations.
- For more information about how to configure confederations, see the “Configuring MPLS Forwarding for ASBR Confederations, on page 68”.

MPLS VPN Inter-AS BGP Label Distribution

Note This section is not applicable to Inter-AS over IP tunnels.

You can set up the MPLS VPN Inter-AS network so that the ASBRs exchange IPv4 routes with MPLS labels of the provider edge (PE) routers. Route reflectors (RRs) exchange VPN-IPv4 routes by using multihop, multiprotocol external Border Gateway Protocol (eBGP). This method of configuring the Inter-AS system is often called MPLS VPN Inter-AS BGP Label Distribution.

Configuring the Inter-AS system so that the ASBRs exchange the IPv4 routes and MPLS labels has the following benefits:

- Saves the ASBRs from having to store all the VPN-IPv4 routes. Using the route reflectors to store the VPN-IPv4 routes and distributes them to the PE routers results in improved scalability compared with configurations in which the ASBR holds all the VPN-IPv4 routes and distributes the routes based on VPN-IPv4 labels.
- Having the route reflectors hold the VPN-IPv4 routes also simplifies the configuration at the border of the network.
- Enables a non-VPN core network to act as a transit network for VPN traffic. You can transport IPv4 routes with MPLS labels over a non-MPLS VPN service provider.
- Eliminates the need for any other label distribution protocol between adjacent label switch routers (LSRs). If two adjacent LSRs are also BGP peers, BGP can handle the distribution of the MPLS labels. No other label distribution protocol is needed between the two LSRs.

Exchanging IPv4 Routes with MPLS labels

Note This section is not applicable to Inter-AS over IP tunnels.

You can set up a VPN service provider network to exchange IPv4 routes with MPLS labels. You can configure the VPN service provider network as follows:

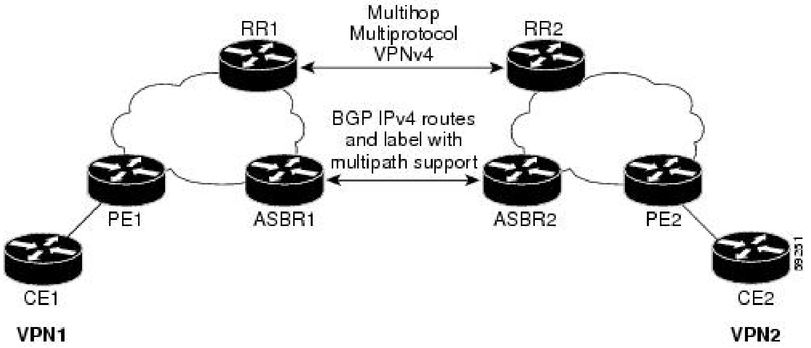
- Route reflectors exchange VPN-IPv4 routes by using multihop, multiprotocol eBGP. This configuration also preserves the next-hop information and the VPN labels across the autonomous systems.
- A local PE router (for example, PE1 in the figure below) needs to know the routes and label information for the remote PE router (PE2).

  This information can be exchanged between the PE routers and ASBRs in one of two ways:

  - Internal Gateway Protocol (IGP) and Label Distribution Protocol (LDP): The ASBR can redistribute the IPv4 routes and MPLS labels it learned from eBGP into IGP and LDP and from IGP and LDP into eBGP.
  - Internal Border Gateway Protocol (iBGP) IPv4 label distribution: The ASBR and PE router can use direct iBGP sessions to exchange VPN-IPv4 and IPv4 routes and MPLS labels.

Alternatively, the route reflector can reflect the IPv4 routes and MPLS labels learned from the ASBR to the PE routers in the VPN. This reflecting of learned IPv4 routes and MPLS labels is accomplished by enabling the ASBR to exchange IPv4 routes and MPLS labels with the route reflector. The route reflector also reflects the VPN-IPv4 routes to the PE routers in the VPN. For example, in VPN1, RR1 reflects to PE1 the VPN-IPv4 routes it learned and IPv4 routes and MPLS labels learned from ASBR1. Using the route reflectors to store the VPN-IPv4 routes and forward them through the PE routers and ASBRs allows for a scalable configuration.

Figure 3: VPNs Using eBGP and iBGP to Distribute Routes and MPLS Labels



BGP Routing Information

BGP routing information includes the following items:

- Network number (prefix), which is the IP address of the destination.
- Autonomous system (AS) path, which is a list of the other ASs through which a route passes on the way to the local router. The first AS in the list is closest to the local router; the last AS in the list is farthest from the local router and usually the AS where the route began.
- Path attributes, which provide other information about the AS path, for example, the next hop.

BGP Messages and MPLS Labels

MPLS labels are included in the update messages that a router sends. Routers exchange the following types of BGP messages:

- Open messages—After a router establishes a TCP connection with a neighboring router, the routers exchange open messages. This message contains the number of the autonomous system to which the router belongs and the IP address of the router that sent the message.
- Update messages—When a router has a new, changed, or broken route, it sends an update message to the neighboring router. This message contains the NLRI, which lists the IP addresses of the usable routes. The update message includes any routes that are no longer usable. The update message also includes path attributes and the lengths of both the usable and unusable paths. Labels for VPN-IPv4 routes are encoded in the update message, as specified in RFC 2858. The labels for the IPv4 routes are encoded in the update message, as specified in RFC 3107.
- Keepalive messages—Routers exchange keepalive messages to determine if a neighboring router is still available to exchange routing information. The router sends these messages at regular intervals. (Sixty seconds is the default for Cisco routers.) The keepalive message does not contain routing data; it contains only a message header.
- Notification messages—When a router detects an error, it sends a notification message.

Sending MPLS Labels with Routes

When BGP (eBGP and iBGP) distributes a route, it can also distribute an MPLS label that is mapped to that route. The MPLS label mapping information for the route is carried in the BGP update message that contains the information about the route. If the next hop is not changed, the label is preserved.

When you issue the **show bgp neighbors** *ip-address* command on both BGP routers, the routers advertise to each other that they can then send MPLS labels with the routes. If the routers successfully negotiate their ability to send MPLS labels, the routers add MPLS labels to all outgoing BGP updates.

Inter-AS Option B for L3VPN

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Resource Optimization for MPLS Inter-AS Option B	Release 7.9.1	You can now preserve MPLS encapsulation ID resources for Inter-AS option B local labels on routers that have Cisco NC57 line cards installed, with Embedded Ternary Content-Addressable Memory (eTCAM) cards without the need to allocate any additional resources for these IDs. This feature is enabled by default and you cannot disable it. Previously, these encapsulation IDs were allocated but left unused.
Inter-AS Option B for L3VPN	Release 7.4.1	This feature allows ISPs to provide MPLS Layer 3 VPN services to their end customers where the routing boundaries for a customer are spread across different geographical locations. Separate autonomous systems with autonomous system boundary routers (ASBRs) from different service providers can communicate by exchanging VPN-IPv4 addresses or IPv4 routes and MPLS labels. This feature provides better scalability as it requires only one BGP session to exchange all VPN prefixes between the ASBRs.

A Multiprotocol Label Switching (MPLS) Layer 3 VPN consists of a set of sites that are interconnected using an MPLS provider core network. At each customer site, one or more customer edge (CE) routers attach to one or more provider edge (PE) routers. L3VPN Inter-AS Option B is one of the ways the VPN sites share the routes.

When you configure Inter-AS Option B, subinterfaces of the router enable the ASBR ports to receive the MPLS traffic. ASBR uses MP-BGP sessions to distribute labeled VPN prefixes between the ASBRs. A VPN label is assigned whenever the BGP next hop is changed.

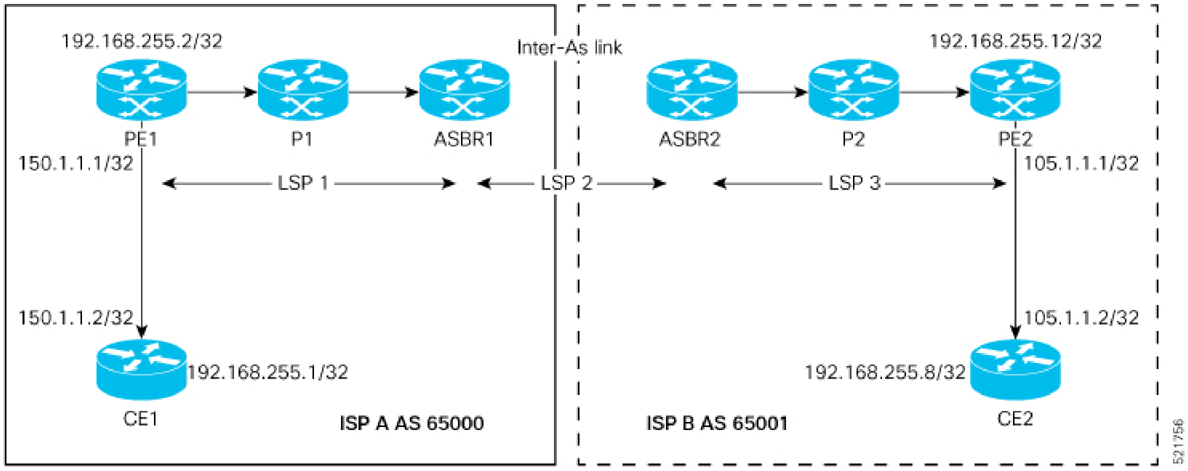
Functions of Inter-AS Option B

- This feature allows to have an iBGP VPNv4 session between the routers within an AS and also an eBGP VPNv4 session between edge routers and WAN routers.
- BGP distributes the label between ASBRs.

  The label mapping information for a particular route is added to the same BGP update message that is used to distribute the route itself.
- When BGP is used to distribute a particular route, it also distributes an MPLS label which is mapped to that route. Many ISPs prefer this method of configuration since it ensures complete IGP isolation between different sites.

Topology

Figure 4: L3VPN Inter-AS Option B



In this topology:

- eBGP is configured as the routing protocol between CE and PE devices.
- ISIS is used as the IGP for the ISP core. On all the physical links of both ISPs, LDP and IGP are configured.
- LDP and IGP are not configured on the Inter-As link between ASBR1 and ASBR2.
- MP iBGP is used to exchange routes between PE and ASBR within a customer network.
- On the ASBRs, the eBGP VPNv4 peering is configured. MPLS is not enabled with LDP on the link connecting the ASBRs.
- When the eBGP VPNv4 peering comes up on the ASBR1, the MPLS BGP forwarding happens automatically on the Inter-As link. Exchange of the labels with ASBR2 is accomplished through BGP-Labeled Unicast (LU), and not through LDP or IGP.
- No VRF is required on ASBRs.
- Use the **retain route-target all** command on the ASBR to refrain from dropping the updates from those VRFs which do not have RT configured in them. The default behavior is that the ASBR drops the update for those VRFs with RT which are not locally configured.
- Configure a static /32 route to remote ASBR next-hop interface address, so that the MPLS label is bound for a /32 prefix. If you do not configure a static /32 route, the control plane comes up, without the traffic not being forwarded.
- Cisco IOS XR does not send or receive routing updates with eBGP peers unless a route policy is configured. A route policy is configured with pass-all which enables sending and receiving all updates.

Note L3VPN Inter-AS Option B does not support BGP-LU as an underlay.

Configure Inter-AS Option B for L3VPN

Perform the following task on PE1, P1, ASBR1, ASBR2, P2, and PE2 to configure Inter-AS Option B for L3VPN.

Configuration Example

PE1 Configuration

```
Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 192.168.255.2/32
Router(config-if)# ipv6 address 50:50:50::50/128
Router(config-if)# exit
Router(config)# interface Bundle-Ether25
Router(config-if)# description interface to R4
Router(config-if)# ipv4 address 172.16.0.1 255.240.0.0
Router(config-if)# exit
Router(config)# router isis access
Router(config-isis)# is-type level-1
Router(config-isis)# net 49.0001.0000.0000.0050.00
Router(config-isis)# nsr
Router(config-isis)# nsf ietf
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether25
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# interface Bundle-Ether25
Router(config-ldp-if)# exit
Router(config-ldp)# exit
Router(config)# vrf vrf1
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target 100:1
Router(config-vrf-af)# export route-target 100:1
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# import route-target 100:1
Router(config-vrf-af)# export route-target 100:1
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# router bgp 65000
Router(config-bgp)# nsr
Router(config-bgp)# bgp router-id 192.168.255.2
Router(config-bgp)# bgp graceful-restart
Router(config-bgp)# address-family vpnv4 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# address-family vpnv6 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 10.10.10.10
Router(config-bgp-nbr)# remote-as 65000
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# vrf vrf1
Router(config-bgp-vrf)# rd 100:1
Router(config-bgp-vrf)# bgp router-id 192.168.255.2
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# address-family ipv6 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# neighbor 150.1.1.2
Router(config-bgp-vrf-nbr)# remote-as 501
Router(config-bgp-vrf-nbr)# address-family ipv4 unicast
Router(config-bgp-vrf-nbr-af)# route-policy pass-all in
Router(config-bgp-vrf-nbr-af)# route-policy pass-all out
Router(config-bgp-vrf-nbr-af)# exit
Router(config-bgp-vrf-nbr)# exit
Router(config-bgp-vrf)# neighbor 150:1:1::2
Router(config-bgp-vrf-nbr)# remote-as 501
Router(config-bgp-vrf-nbr)# address-family ipv6 unicast
Router(config-bgp-vrf-nbr-af)# route-policy pass-all in
Router(config-bgp-vrf-nbr-af)# route-policy pass-all out
Router(config-bgp-vrf-nbr-af)# commit
```

P1 Configuration

```
Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 172.16.0.1 255.255.255.255
Router(config-if)# exit
Router(config)# interface Bundle-Ether12
Router(config-if)# description interface to ASBR1
Router(config-if)# ipv4 address 10.20.1.2 255.240.0.0
Router(config-if)# exit
Router(config)# interface Bundle-Ether25
Router(config-if)# description interface to R50
Router(config-if)# ipv4 address 20.50.1.1 255.240.0.0
Router(config-if)# exit
Router(config)# router isis core
Router(config-isis)# is-type level-1
Router(config-isis)# net 49.0001.0000.0000.0020.00
Router(config-isis)# nsr
Router(config-isis)# nsf ietf
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether12
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Bundle-Ether25
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# nsr
Router(config-ldp)# interface Bundle-Ether12
Router(config-ldp-if)# exit
Router(config-ldp)# interface Bundle-Ether25
Router(config-ldp-if)# commit
```

ASBR1 Configuration

```
Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 10.10.10.10 255.255.255.255
Router(config-if)# ipv6 address 10:10:10::10/128
Router(config-if)# exit
Router(config)# interface Bundle-Ether12
Router(config-if)# description interface to 172.16.0.1
Router(config-if)# ipv4 address 10.20.1.1 255.240.0.0
Router(config-if)# monitor-session Test ethernet port-level
Router(config-if-mon)# exit
Router(config-if)# exit
Router(config)# interface Bundle-Ether11
Router(config-if)# description interface to ASBR2
Router(config-if)# ipv4 address 10.0.0.1 255.240.0.0
Router(config-if)# exit
Router(config)# router isis core
Router(config-isis)# is-type level-1
Router(config-isis)# net 49.0001.0000.0000.0010.00
Router(config-isis)# nsr
Router(config-isis)# nsf ietf
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether12
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# nsr
Router(config-ldp)# interface Bundle-Ether12
Router(config-ldp-if)# exit
Router(config-ldp)# exit
Router(config)# router bgp 65000
Router(config-bgp)# nsr
Router(config-bgp)# bgp router-id 10.10.10.10
Router(config-bgp)# bgp graceful-restart
Router(config-bgp)# address-family vpnv4 unicast
Router(config-bgp-af)# retain route-target all
Router(config-bgp-af)# exit
Router(config-bgp)# address-family vpnv6 unicast
Router(config-bgp-af)# retain route-target all
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 10.0.0.1
Router(config-bgp-nbr)# remote-as 65001
Router(config-bgp-nbr)# ebgp-multihop 2
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# neighbor 192.168.255.2
Router(config-bgp-nbr)# remote-as 65000
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# exit
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 10.0.0.1/8 172.16.0.1
Router(config-static-afi)# commit
```

ASBR2 Configuration

```
Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 10.0.0.1 255.0.0.0
Router(config-if)# ipv6 address 1:1:1::1/128
Router(config-if)# exit
Router(config)# interface Bundle-Ether12
Router(config-if)# description interface to P2
Router(config-if)# ipv4 address 1.2.1.1 255.240.0.0
Router(config-if)# monitor-session Test ethernet port-level
Router(config-if-mon)# exit
Router(config-if)# exit
Router(config)# interface Bundle-Ether11
Router(config-if)# description interface to ASBR1
Router(config-if)# ipv4 address 1.10.1.2 255.240.0.0
Router(config-if)# exit
Router(config)# router isis core
Router(config-isis)# is-type level-1
Router(config-isis)# net 49.0001.0000.0000.0010.00
Router(config-isis)# nsr
Router(config-isis)# nsf ietf
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether12
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# nsr
Router(config-ldp)# interface Bundle-Ether12
Router(config-ldp-if)# exit
Router(config-ldp)# exit
Router(config)# router bgp 65001
Router(config-bgp)# nsr
Router(config-bgp)# bgp router-id 10.0.0.1
Router(config-bgp)# bgp graceful-restart
Router(config-bgp)# address-family vpnv4 unicast
Router(config-bgp-af)# retain route-target all
Router(config-bgp-af)# exit
Router(config-bgp)# address-family vpnv6 unicast
Router(config-bgp-af)# retain route-target all
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 5.5.5.5
Router(config-bgp-nbr)# remote-as 65001
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# neighbor 10.10.10.10
Router(config-bgp-nbr)# remote-as 65000
Router(config-bgp-nbr)# ebgp-multihop 2
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# exit
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 10.10.10.10/32 10.0.0.1
Router(config-static-afi)# commit
```

P2 Configuration

```
Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 2.2.2.2 255.255.255.255
Router(config-if)# exit
Router(config)# interface Bundle-Ether12
Router(config-if)# description interface towards ASBR2
Router(config-if)# ipv4 address 1.2.1.2 255.240.0.0
Router(config-if)# exit
Router(config)# interface Bundle-Ether25
Router(config-if)# description interface towards PE2
Router(config-if)# ipv4 address ipv4 address 2.5.1.1 255.240.0.0
Router(config-if)# exit
Router(config)# router isis core
Router(config-isis)# is-type level-1
Router(config-isis)# net 49.0001.0000.0000.0020.00
Router(config-isis)# nsr
Router(config-isis)# nsf ietf
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether12
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Bundle-Ether25
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# interface Bundle-Ether12
Router(config-ldp-if)# exit
Router(config-ldp)# interface Bundle-Ether25
Router(config-ldp-if)# commit
```

PE2 Configuration

```
Router(config)# router bgp 65001
Router(config-bgp)# nsr
Router(config-bgp)# bgp router-id 192.168.255.12
Router(config-bgp)# bgp graceful-restart
Router(config-bgp)# address-family vpnv4 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# address-family vpnv6 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 10.0.0.1
Router(config-bgp-nbr)# remote-as 65001
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# vrf vrf1
Router(config-bgp-vrf)# rd 100:1
Router(config-bgp-vrf)# bgp router-id 192.168.255.12
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# address-family ipv6 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# neighbor 150.1.1.2
Router(config-bgp-vrf-nbr)# remote-as 501
Router(config-bgp-vrf-nbr)# address-family ipv4 unicast
Router(config-bgp-vrf-nbr-af)# route-policy pass-all in
Router(config-bgp-vrf-nbr-af)# route-policy pass-all out
Router(config-bgp-vrf-nbr-af)# exit
Router(config-bgp-vrf-nbr)# exit
Router(config-bgp-vrf)# neighbor 150:1:1::2
Router(config-bgp-vrf-nbr)# remote-as 501
Router(config-bgp-vrf-nbr)# address-family ipv6 unicast
Router(config-bgp-vrf-nbr-af)# route-policy pass-all in
Router(config-bgp-vrf-nbr-af)# route-policy pass-all out
Router(config-bgp-vrf-nbr-af)# exit
Router(config-bgp-vrf-nbr)# exit
Router(config-bgp-vrf)# exit
Router(config-bgp)# exit
Router(config-bgp)# exit
Router(config)# interface TenGigE0/0/0/30.1
Router(config-subif)# vrf vrf1
Router(config-subif)# ipv4 address 172.16.0.1 255.240.0.0
Router(config-subif)# ipv6 address 105:1:1::1/96
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# exit
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 192.168.255.12 255.255.255.224
Router(config-if)# ipv6 address 50:50:50::50/128
Router(config-if)# exit
Router(config)# router isis access
Router(config-isis)# is-type level-1
Router(config-isis)# net 49.0001.0000.0000.0050.00
Router(config-isis)# nsr
Router(config-isis)# nsf ietf
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether25
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# interface Bundle-Ether25
Router(config-ldp-if)# exit
Router(config-ldp)# exit
Router(config)# vrf vrf1
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target 100:1
Router(config-vrf-af)# export route-target 100:1
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# import route-target 100:1
Router(config-vrf-af)# export route-target 100:1
Router(config-vrf-af)# commit
```

Running Configuration

This section shows the running configuration of Inter-AS Option B for L3VPN.

The following is the running configuration of PE1:

```
interface Loopback0
 ipv4 address 192.168.255.2/32
 ipv6 address 50:50:50::50/128
!
interface Bundle-Ether25
 description interface to R4
 ipv4 address 172.16.0.1 255.240.0.0
 !
!
router isis access
 is-type level-1
 net 49.0001.0000.0000.0050.00
 nsr
 nsf ietf
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
 !
 interface Bundle-Ether25
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface Loopback0
  point-to-point
  address-family ipv4 unicast
  !
 !
!
mpls ldp
 interface Bundle-Ether25
 !
!

vrf vrf1
 address-family ipv4 unicast
  import route-target 100:1
  !
  export route-target 100:1
  !
 !
 address-family ipv6 unicast
  import route-target 100:1
  !
  export route-target 100:1
  !
 !
!
router bgp 65000
 nsr
 bgp router-id 192.168.255.2
 bgp graceful-restart
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
 neighbor 10.10.10.10
  remote-as 65000
  update-source Loopback0
  address-family vpnv4 unicast
   maximum-prefix 4500000 90
  !
  address-family vpnv6 unicast
   maximum-prefix 4500000 90
  !
 !
 vrf vrf1
  rd 100:1
  bgp router-id 192.168.255.2
  address-family ipv4 unicast
   label mode per-vrf
   redistribute connected
  !
  address-family ipv6 unicast
   label mode per-vrf
   redistribute connected
  !
  neighbor 150.1.1.2
   remote-as 501
   address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
  neighbor 150:1:1::2
   remote-as 501
   address-family ipv6 unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
 !
```

The following is the running configuration of P1:

```
interface Loopback0
 ipv4 address 172.16.0.1 255.255.255.255
!
interface Bundle-Ether12
 description interface to ASBR1
 ipv4 address 10.20.1.2 255.240.0.0
!
interface Bundle-Ether25
```

```
 description interface to R50
 ipv4 address 20.50.1.1 255.240.0.0
!
router isis core
 is-type level-1
 net 49.0001.0000.0000.0020.00
 nsr
 nsf ietf
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
 !
 interface Bundle-Ether12
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface Bundle-Ether25
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface Loopback0
  point-to-point
  address-family ipv4 unicast
  !
 !
!

mpls ldp
 nsr
 interface Bundle-Ether12
 !
 interface Bundle-Ether25
 !
!
```

The following is the running configuration of ASBR1:

```
interface Loopback0
 ipv4 address 10.10.10.10 255.255.255.255
 ipv6 address 10:10:10::10/128
!
interface Bundle-Ether12
 description interface to 172.16.0.1
 ipv4 address 10.20.1.1 255.240.0.0
 monitor-session Test ethernet port-level
 !
!
interface Bundle-Ether11
 description interface to ASBR2
 ipv4 address 10.0.0.1 255.240.0.0
!
router isis core
 is-type level-1
 net 49.0001.0000.0000.0010.00
 nsr
 nsf ietf
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
 !
 interface Bundle-Ether12
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface Loopback0
  point-to-point
  address-family ipv4 unicast
  !
 !
!

mpls ldp
 nsr
 interface Bundle-Ether12
 !
!

router bgp 65000
 nsr
 bgp router-id 10.10.10.10
 bgp graceful-restart
 address-family vpnv4 unicast
  retain route-target all
 !
  address-family vpnv6 unicast
  retain route-target all
 !
 neighbor 10.0.0.1
  remote-as 65001
  ebgp-multihop 2
  update-source Loopback0
  address-family vpnv4 unicast
   route-policy pass-all in
   maximum-prefix 4500000 90
   route-policy pass-all out
  !
  address-family vpnv6 unicast
   route-policy pass-all in
   maximum-prefix 4500000 90
   route-policy pass-all out
  !
 !
  neighbor 192.168.255.2
  remote-as 65000
  update-source Loopback0
  address-family vpnv4 unicast
   maximum-prefix 4500000 90
   next-hop-self
  !
  address-family vpnv6 unicast
   maximum-prefix 4500000 90
   next-hop-self
  !
 !
!
router static
 address-family ipv4 unicast
  10.0.0.1/8 1.10.1.2
 !
!
```

The following is the running configuration of ASBR2:

```
interface Loopback0
 ipv4 address 10.0.0.1 255.0.0.0
 ipv6 address 1:1:1::1/128
!
interface Bundle-Ether12
 description interface to P2
 ipv4 address 1.2.1.1 255.240.0.0
!
interface Bundle-Ether11
 description interface to ASBR1
 ipv4 address 1.10.1.2 255.240.0.0
!
router isis core
 is-type level-1
 net 49.0001.0000.0000.0001.00
 nsr
 distribute link-state
 nsf ietf
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
 !
 interface Bundle-Ether12
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface Loopback0
  point-to-point
  address-family ipv4 unicast
  !
 !
!
mpls ldp
 interface Bundle-Ether12
 !
!

router bgp 65001
 nsr
 bgp router-id 10.0.0.1
 bgp graceful-restart
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
  retain route-target all
 !
 address-family vpnv6 unicast
  retain route-target all
 !
 neighbor 5.5.5.5
  remote-as 65001
  update-source Loopback0
  address-family vpnv4 unicast
   next-hop-self
  !
  address-family vpnv6 unicast
   next-hop-self
  !
 !
  neighbor 10.10.10.10
  remote-as 65000
  ebgp-multihop 2
  update-source Loopback0
  address-family vpnv4 unicast
   route-policy pass-all in
   maximum-prefix 4500000 90
   route-policy pass-all out
  !
  address-family vpnv6 unicast
   route-policy pass-all in
   maximum-prefix 4500000 90
   route-policy pass-all out
  !
 !

router static
 address-family ipv4 unicast
  10.10.10.10/32 10.0.0.1
 !
```

The following is the running configuration of P2:

```
interface Loopback0
 ipv4 address 2.2.2.2 255.255.255.255
!
interface Bundle-Ether12
 description interface towards ASBR2
 ipv4 address 1.2.1.2 255.240.0.0
!
interface Bundle-Ether25
 description interface towards PE2
 ipv4 address 2.5.1.1 255.240.0.0
!

router isis core
 is-type level-1
 net 49.0001.0000.0000.0002.00
 nsr
 nsf ietf
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
 !
 interface Bundle-Ether12
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface Bundle-Ether25
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface Loopback0
  point-to-point
  address-family ipv4 unicast
  !
 !
!

mpls ldp
 interface Bundle-Ether12
 !
 interface Bundle-Ether25
 !
!
```

The following is the running configuration of PE2:

```
router bgp 65001
 nsr
 bgp router-id 192.168.255.12
 bgp graceful-restart
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
 neighbor 10.0.0.1
  remote-as 65001
  update-source Loopback0
  address-family vpnv4 unicast
   maximum-prefix 4500000 90
  !
  address-family vpnv6 unicast
   maximum-prefix 4500000 90
  !
!
 vrf vrf1
  rd 100:1
  bgp router-id 192.168.255.12
  address-family ipv4 unicast
   label mode per-vrf
   redistribute connected
  !
  address-family ipv6 unicast
   label mode per-vrf
   redistribute connected
  !
  neighbor 105.1.1.2
   remote-as 501
   address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
  neighbor 105:1:1::2
   remote-as 501
   address-family ipv6 unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
 !

interface TenGigE0/0/0/30.1
 vrf vrf1
 ipv4 address 105.1.1.1 255.240.0.0
 ipv6 address 105:1:1::1/96
 encapsulation dot1q 1
!

interface Loopback0
 ipv4 address 5.5.5.5 255.255.255.255
 ipv6 address 5:5:5::5/128
!

router isis access
 is-type level-1
 net 49.0001.0000.0000.0005.00
 nsr
 nsf ietf
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
 !
 interface Bundle-Ether25
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface Loopback0
  point-to-point
  address-family ipv4 unicast
  !
 !
!

mpls ldp
 interface Bundle-Ether25
 !
!

vrf vrf1
 address-family ipv4 unicast
  import route-target
   100:1
  !
  export route-target
   100:1
  !
 !
 address-family ipv6 unicast
  import route-target
   100:1
  !
  export route-target
   100:1
```

Verification

Verification on PE1.

L3VPN route 202.1.0.0/24 is learned through iBGP from ASBR1 on PE1 over address family VPNv4 unicast.

```
Router:PE1# show route vrf vrf1
Sun Jun  6 23:08:38.433 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, 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, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is not set

B    105.1.1.0/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:04:43
C    150.1.1.0/24 is directly connected, 01:14:27, TenGigE0/0/0/22/0.1
```

```
L    150.1.1.1/32 is directly connected, 01:14:27, TenGigE0/0/0/22/0.1
B    202.1.0.0/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:00:08
B    202.1.1.0/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:00:08
```

The following output shows that you can reach 202.1.0.0/24 using a VPN label of 24521. The next hop for the VPNv4 prefix decides the transport label as well as the label switched path.

```
Router:PE1# show bgp vpnv4 unicast rd 100:1 202.1.0.0/24
Sun Jun  6 23:12:12.140 UTC
BGP routing table entry for 202.1.0.0/24, Route Distinguisher: 100:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               2844        2844
Last Modified: Jun  6 23:08:30.194 for 00:03:42
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  200 501
    10.10.10.10 (metric 30) from 10.10.10.10 (10.10.10.10)
      Received Label 24521
      Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported
      Received Path ID 0, Local Path ID 1, version 2844
      Extended community: RT:100:1
      Source AFI: VPNv4 Unicast, Source VRF: vrf1, Source Route Distinguisher: 100:1

Router:PE1# show cef vrf vrf1 202.1.0.0
Mon Jun  7 02:07:39.841 UTC
202.1.0.0/24, version 513583, internal 0x5000001 0x30 (ptr 0xa3f8bac8) [1], 0x0 (0x0), 0x208
 (0x8f505928)
 Updated Jun  7 01:50:33.710
 Prefix Len 24, traffic index 0, precedence n/a, priority 3
  gateway array (0x8f2d20e8) reference count 252, flags 0x2038, source rib (7), 0 backups
                [1 type 1 flags 0x48441 (0x8ad86708) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Jun  6 23:20:45.951
 LDI Update time Jun  6 23:20:45.951
   via 10.10.10.10/32, 5 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa25ff9d8 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 10.10.10.10/32 via 24003/0/21
     next hop 20.50.1.1/32 BE25         labels imposed {24004 24521}

    Load distribution: 0 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   recursive                 24003/0
```

The following output shows the transport label information to reach 202.1.0.0/24.

```
Router:PE1# show mpls forwarding prefix 10.10.10.10/32
Mon Jun  7 02:06:40.845 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24003  24004       10.10.10.10/32     BE25         20.50.1.1       141107
-------------------------------------------------------------------------------

Router:PE1# show cef vrf vrf1 202.1.0.0
Mon Jun  7 02:07:39.841 UTC
```

```
202.1.0.0/24, version 513583, internal 0x5000001 0x30 (ptr 0xa3f8bac8) [1], 0x0 (0x0), 0x208
 (0x8f505928)
 Updated Jun  7 01:50:33.710
 Prefix Len 24, traffic index 0, precedence n/a, priority 3
  gateway array (0x8f2d20e8) reference count 252, flags 0x2038, source rib (7), 0 backups
                [1 type 1 flags 0x48441 (0x8ad86708) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Jun  6 23:20:45.951
 LDI Update time Jun  6 23:20:45.951
   via 10.10.10.10/32, 5 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa25ff9d8 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 10.10.10.10/32 via 24003/0/21
     next hop 20.50.1.1/32 BE25         labels imposed {24004 24521}

    Load distribution: 0 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   recursive                 24003/0
```

Verification on P1.

P1 performs a PHP operation for transport label and exposes the VPN label before forwarding the traffic to next-hop 10.10.10.10.

```
Router:P1#d show mpls forwarding prefix 10.10.10.10/32
Mon Jun  7 02:34:55.293 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Pop         10.10.10.10/32     BE12         10.20.1.1       28804
-------------------------------------------------------------------------------
```

Verification on ASBR1.

ASBR1 learns the remote route 202.1.0.0/24 from ASBR2 through address-family VPNv4 unicast. The next hop is the ASBR2 loopback0.

After receiving this update, it is advertised to the local PE1 through iBGP address-family VPNv4 unicast. The next-hop-self configuration is used on ASBR1 since it is reachable through IGP from PE1, so the next hop is changed to itself. The traffic arrives from PE1 with a label 24521 and is swapped with label 25516 before forwarding it to ASBR2.

```
Router:ASBR1# show bgp vpnv4 unicast rd 100:1 202.1.0.0
Sun Jun  6 19:28:09.018 EDT
BGP routing table entry for 202.1.0.0/24, Route Distinguisher: 100:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker            1002022     1002022
    Local Label: 24521

Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.3
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.3
  200 501
    10.0.0.1 from 10.0.0.1 (10.0.0.1)
      Received Label 25516
      Origin IGP, localpref 100, valid, external, best, group-best, import-candidate,
not-in-vrf
      Received Path ID 0, Local Path ID 1, version 1002022
      Extended community: RT:100:1


Router:ASBR1# show bgp vpnv4 unicast rd 100:1 advertised neighbor 192.168.255.2 summary

Network            Next Hop        From            AS Path
Route Distinguisher: 100:1
105.1.1.0/24       10.10.10.10     10.0.0.1         200?
202.1.0.0/24       10.10.10.10     10.0.0.1         200 501i
202.1.1.0/24       10.10.10.10     10.0.0.1         200 501i

Processed 3 prefixes, 3 paths


Router:ASBR1# show mpls forwarding labels 24521
Sun Jun  6 23:05:49.323 EDT
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24521  25516       100:1:202.1.0.0/24              10.0.0.1         0
-------------------------------------------------------------------------------
```

Verification on ASBR2.

The prefix 202.1.0.0/24 is received through iBGP address-family VPNv4 unicast from PE2 with a label of 24002. ASBR2 assigns it a local label of 25516 and advertises it to ASBR1 through eBGP vpnv4 address-family changing the next hop to itself. This local label of 25516 is used by the ASBR1 to forward traffic to ASBR2, which in turn swaps it with a VPN label of 24002 before forwarding it to the next hop.

```
Router:ASBR2# show bgp vpnv4 unicast rd 100:1 202.1.0.0
Sun Jun  6 23:06:32.812 EDT
BGP routing table entry for 202.1.0.0/24, Route Distinguisher: 100:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker           16194881    16194881
    Local Label: 25516
    Gateway Array ID: 21940, Resilient per-PE nexthop set ID: 19598

Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.3
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.3
  501
    5.5.5.5 (metric 30) from 5.5.5.5 (5.5.5.5)
      Received Label 24002
      Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
not-in-vrf
      Received Path ID 0, Local Path ID 1, version 16194881
      Extended community: RT:100:1

Router:ASBR2# show bgp vpnv4 unicast rd 100:1 advertised neighbor 10.10.10.10 summary
Sun Jun  6 23:07:05.617 EDT
Network            Next Hop        From            AS Path
Route Distinguisher: 100:1
105.1.1.0/24       10.0.0.1         5.5.5.5         200?
150.1.1.0/24       10.0.0.1         10.10.10.10     200 100?
202.1.0.0/24       10.0.0.1         5.5.5.5         200 501i
202.1.1.0/24       10.0.0.1         5.5.5.5         200 501i
```

```
Processed 4 prefixes, 4 paths

Router:ASBR2# show mpls forwarding labels 25516
Sun Jun  6 23:07:32.394 EDT
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
25516  24002       No ID                           5.5.5.5         654
-------------------------------------------------------------------------------
```

Verification on P2.

P2 is along the transit path of the traffic. It label switches or pop the transport label. In this example, PHP operation is performed and exposes the VPN label before forwarding the traffic.

```
Router:P2# show mpls forwarding prefix 5.5.5.5/32
Mon Jun  7 03:09:11.532 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24005  Pop         5.5.5.5/32         BE25         2.5.1.2         11921958
-------------------------------------------------------------------------------
```

Verification on PE2.

L3VPN route 202.1.0.0/24 is learned from eBGP neighbor 105.1.1.2 (CE2 interface towards PE2) in vrf1.

```
Router:PE2# show route vrf vrf1

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, 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, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is not set

C    105.1.1.0/24 is directly connected, 6w4d, TenGigE0/0/0/30.1
L    105.1.1.1/32 is directly connected, 6w4d, TenGigE0/0/0/30.1
B    150.1.1.0/24 [200/0] via 10.0.0.1 (nexthop in vrf default), 03:58:16
B    160.1.0.0/24 [200/0] via 10.0.0.1 (nexthop in vrf default), 04:30:07
B    202.1.0.0/24 [20/0] via 105.1.1.2, 01:30:05
B    202.1.1.0/24 [20/0] via 105.1.1.2, 01:30:05
```

The route 202.1.0.0/24 gets installed in VRF1 with a local label of 24002 and then advertised through iBGP address-family VPNv4 unicast to ASBR2 changing the next hop to itself. ASBR2 adds this VPN label before forwarding it to PE2.

```
Router:PE2# show bgp vpnv4 unicast rd 100:1 202.1.0.0

BGP routing table entry for 202.1.0.0/24, Route Distinguisher: 100:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker            1070062     1070062
    Local Label: 24002
Last Modified: Jun  7 01:30:56.657 for 01:31:29
```

```
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    10.0.0.1
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    10.0.0.1
  501
    105.1.1.2 from 105.1.1.2 (194.0.0.1)
      Origin IGP, localpref 100, valid, external, best, group-best, import-candidate
      Received Path ID 0, Local Path ID 1, version 1070062
      Extended community: RT:100:1
```

The traffic that arrives from PE2 with a VPN label of 24002 is assigned an outgoing label ‘aggregate’, which means that the lookup is to be performed in vrf1 RIB to forward it to the next hop on 150.1.1.2.

```
Router:PE2# show mpls forwarding labels 24002
Mon Jun  7 03:02:53.255 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24002  Aggregate   vrf1: Per-VRF Aggr[V]   \
                                      vrf1                         138
-------------------------------------------------------------------------------

Router:PE2# show cef vrf vrf1 202.1.0.0
Mon Jun  7 03:04:08.268 UTC
202.1.0.0/24, version 3477, internal 0x1000001 0x30 (ptr 0x97f75328) [1], 0x0 (0x0), 0x0
(0x0)
 Updated Jun  7 01:30:57.120
 Prefix Len 24, traffic index 0, precedence n/a, priority 3
  gateway array (0x8c820f38) reference count 2, flags 0x2010, source rib (7), 0 backups
                [1 type 3 flags 0x48441 (0x8a79cd88) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Jun  7 01:30:57.120
 LDI Update time Jun  7 01:30:57.120

  Level 1 - Load distribution: 0
  [0] via 105.1.1.2/32, recursive

   via 105.1.1.2/32, 3 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0x8d575b80 0x0]
    next hop 105.1.1.2/32 via 105.1.1.2/32

    Load distribution: 0 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   TenGigE0/0/0/30.1         105.1.1.2
```

How to Implement MPLS Layer 3 VPNs

Implementing MPLS L3VPNs involves these main tasks:

Prerequisites for Implementing MPLS L3VPN

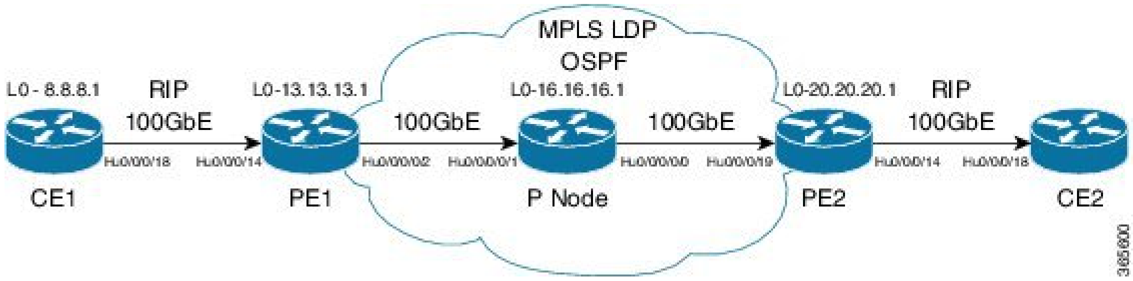
These are the prerequisites to configure MPLS L3VPN:

- You must be in a user group associated with a task group that includes the proper task IDs for these commands:
    - BGP
    - IGP
    - MPLS
    - MPLS Layer 3 VPN
- If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- To configure MPLS Layer 3 VPNs, routers must support MPLS forwarding and Forwarding Information Base (FIB).

Configure the Core Network

Consider a network topology where MPLS L3VPN services are transported over MPLS LDP core.

Figure 5: L3VPN over MPLS LDP



Configuring the core network involves these main tasks:

Assess the Needs of MPLS VPN Customers

Before configuring an MPLS VPN, the core network topology must be identified so that it can best serve MPLS VPN customers. The tasks listed below helps to identify the core network topology.

- Identify the size of the network:

  Identify the following to determine the number of routers and ports required:

    - How many customers to be supported?
    - How many VPNs are required for each customer?
    - How many virtual routing and forwarding (VRF) instances are there for each VPN?
- Determine the routing protocols required in the core.
- Determine if BGP load sharing and redundant paths in the MPLS VPN core are required.

Configure Routing Protocols in the Core

You can use RIP, OSPF or IS-IS as the routing protocol in the core.

Configuration Example

This example lists the steps to configure OSPF as the routing protocol in the core.

```
Router-PE1#configure
Router-PE1(config)#router ospf dc-core
Router-PE1(config-ospf)#address-family ipv4 unicast
Router-PE1(config-ospf)#area 1
Router-PE1(config-ospf-ar)#interface HundredGigE0/0/0/2
Router-PE1(config-ospf-ar-if)#commit
```

Running Configuration

```
router ospf dc-core
 router-id 13.13.13.1
 address-family ipv4 unicast
 area 1
  interface HundredGigE0/0/0/2
  !
 !
!
```

Verification

- Verify the OSPF neighbor and ensure that the *State* is displayed as 'FULL'.

```
Router-PE1# show ospf neighbor
Neighbors for OSPF dc-core

Neighbor ID     Pri   State           Dead Time   Address         Interface
16.16.16.1      1     FULL/-  00:00:34    191.22.1.2      HundredGigE0/0/0/2
    Neighbor is up for 1d18h

Total neighbor count: 1
```

Related Topics

- How to Implement MPLS Layer 3 VPNs, on page 33

For more details on configuring the routing protocol, see *Routing Configuration Guide for Cisco NCS 5500 Series Routers* and *BGP Configuration Guide for Cisco NCS 5500 Series Routers*.

Associated Commands

- router-id
- router ospf

Configure MPLS in the Core

To enable MPLS on all routers in the core, you must configure a Label Distribution Protocol (LDP).

You can also transport MPLS L3VPN services using segment routing in the core. For details, see Configure Segment Routing in MPLS Core, on page 75.

Configuration Example

This example lists the steps to configure LDP in MPLS core.

```
Router-PE1#configure
Router-PE1(config)#mpls ldp
Router-PE1(config-ldp)#router-id 13.13.13.1
Router-PE1(config-ldp)#address-family ipv4
Router-PE1(config-ldp-af)#exit
Router-PE1(config-ldp)#interface HundredGigE0/0/0/2
Router-PE1(config-ldp-if)#commit
```

Repeat this configuration in PE2 and P routers as well.

Running Configuration

```
mpls ldp
 router-id 13.13.13.1
 address-family ipv4
 !
 interface HundredGigE0/0/0/2
 !
!
```

Verification

- Verify that the neighbor (16.16.16.1) is UP through the core interface:

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
  TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
  LDP Discovery Sources:
    IPv4: (1)
      HundredGigE0/0/0/2HundredGigE 0/9/0/0
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.64.98.32    87.0.0.2       88.88.88.14    50.50.50.50
      178.0.0.1      192.1.1.1
    IPv6: (0)
```

Related Topics

- How to Implement MPLS Layer 3 VPNs, on page 33

For more details on configuring MPLS LDP, see the *Implementing MPLS Label Distribution Protocol* chapter in the *MPLS Configuration Guide for Cisco NCS 5500 Series Routers*.

Associated Commands

- mpls ldp
- show mpls ldp neighbor

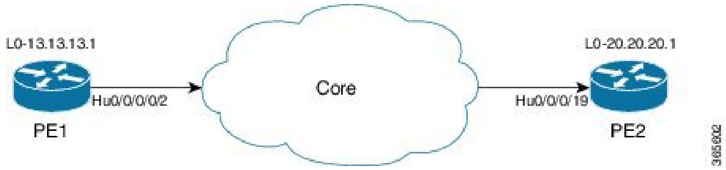
Determine if FIB is Enabled in the Core

Forwarding Information Base (FIB) must be enabled on all routers in the core, including the provider edge (PE) routers. For information on how to determine if FIB is enabled, see the *Implementing Cisco Express Forwarding* module in the *IP Addresses and Services Configuration Guide for Cisco NCS 5500 Series Routers*.

Configure Multiprotocol BGP on the PE Routers and Route Reflectors

Multiprotocol BGP (MP-BGP) propagates VRF reachability information to all members of a VPN community. You must configure MP-BGP peering in all the PE routers within a VPN community.

Figure 6: Multiprotocol BGP on PE Routers



Configuration Example

This example shows how to configure MP-BGP on PE1. The loopback address (20.20.20.1) of PE2 is specified as the neighbor of PE1. Similarly, you must perform this configuration on PE2 node as well, with the loopback address (13.13.13.1) of PE1 specified as the neighbor of PE2.

```
Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 13.13.13.1
Router-PE1(config-bgp)#address-family ipv4 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#address-family vpnv4 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#neighbor 20.20.20.1
Router-PE1(config-bgp-nbr)#remote-as 2001
Router-PE1(config-bgp-nbr)#update-source loopback 0
Router-PE1(config-bgp-nbr)#address-family ipv4 unicast
Router-PE1(config-bgp-nbr-af)#exit
Router-PE1(config-bgp-nbr)#address-family vpnv4 unicast
Router-PE1(config-bgp-nbr-af)#exit
Router-PE1(config-bgp-nbr)#exit
/* VRF configuration */
Router(config-bgp)# vrf vrf1601
Router-PE1(config-bgp-vrf)#rd 2001:1601
Router-PE1(config-bgp-vrf)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-af)#label mode per-vrf
Router-PE1(config-bgp-vrf-af)#redistribute connected
Router-PE1(config-bgp-vrf-af)#commit
```

Running Configuration

```
router bgp 2001
 bgp router-id 13.13.13.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 20.20.20.1
  remote-as 2001
  update-source Loopback0
  address-family vpnv4 unicast
  !
  address-family ipv4 unicast
  !
 !
 vrf vrf1601
  rd 2001:1601
  address-family ipv4 unicast
   label mode per-vrf
   redistribute connected
  !
 !
```

Verification

- Verify if the BGP state is established, and if the Remote AS and local AS displays the same value (2001 in this example):

```
Router-PE1#show bgp neighbor

BGP neighbor is 20.20.20.1
 Remote AS 2001, local AS 2001, internal link
 Remote router ID 20.20.20.1
  BGP state = Established, up for 1d19h
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00
  Hold time is 60, keepalive interval is 20 seconds
  Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
  Last write 00:00:16, attempted 19, written 19
  Second last write 00:00:36, attempted 19, written 19
  Last write before reset 00:00:00, attempted 0, written 0
  Second last write before reset 00:00:00, attempted 0, written 0
  Last write pulse rcvd  Apr 12 10:31:20.739 last full not set pulse count 27939
  Last write pulse rcvd before reset 00:00:00
  Socket not armed for io, armed for read, armed for write
  Last write thread event before reset 00:00:00, second last 00:00:00
  Last KA expiry before reset 00:00:00, second last 00:00:00
  Last KA error before reset 00:00:00, KA not sent 00:00:00
  Last KA start before reset 00:00:00, second last 00:00:00
  Precedence: internet
  Non-stop routing is enabled
  Multi-protocol capability received
  Neighbor capabilities:
    Route refresh: advertised (old + new) and received (old + new)
    Graceful Restart (GR Awareness): received
    4-byte AS: advertised and received
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
  Received 25595 messages, 0 notifications, 0 in queue
  Sent 8247 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 0 secs
  Inbound message logging enabled, 3 messages buffered
  Outbound message logging enabled, 3 messages buffered

 For Address Family: IPv4 Unicast
  BGP neighbor version 484413
  Update group: 0.4 Filter-group: 0.3  No Refresh request being processed
  Inbound soft reconfiguration allowed
  NEXT_HOP is always this router
  AF-dependent capabilities:
    Outbound Route Filter (ORF) type (128) Prefix:
      Send-mode: advertised, received
      Receive-mode: advertised, received
    Graceful Restart capability received
      Remote Restart time is 120 seconds
      Neighbor did not preserve the forwarding state during latest restart
    Additional-paths Send: advertised and received
    Additional-paths Receive: advertised and received
  Route refresh request: received 1, sent 1
  Policy for incoming advertisements is pass-all
  Policy for outgoing advertisements is pass-all
  24260 accepted prefixes, 24260 are bestpaths
  Cumulative no. of prefixes denied: 0.
  Prefix advertised 2000, suppressed 0, withdrawn 0
  Maximum prefixes allowed 1048576
  Threshold for warning message 75%, restart interval 0 min
  AIGP is enabled
  An EoR was received during read-only mode
  Last ack version 484413, Last synced ack version 0
  Outstanding version objects: current 0, max 1
  Additional-paths operation: Send and Receive
  Send Multicast Attributes
  Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with stitching-RT
 option

 For Address Family: VPNv4 Unicast
  BGP neighbor version 798487
  Update group: 0.2 Filter-group: 0.1  No Refresh request being processed
  AF-dependent capabilities:
    Graceful Restart capability received
      Remote Restart time is 120 seconds
      Neighbor did not preserve the forwarding state during latest restart
    Additional-paths Send: advertised and received
    Additional-paths Receive: advertised and received
  Route refresh request: received 0, sent 0
  29150 accepted prefixes, 29150 are bestpaths
  Cumulative no. of prefixes denied: 0.
  Prefix advertised 7200, suppressed 0, withdrawn 0
  Maximum prefixes allowed 2097152
  Threshold for warning message 75%, restart interval 0 min
  AIGP is enabled
  An EoR was received during read-only mode
  Last ack version 798487, Last synced ack version 0
  Outstanding version objects: current 0, max 1
  Additional-paths operation: Send and Receive
  Send Multicast Attributes
  Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with stitching-RT
 option

  Connections established 1; dropped 0
  Local host: 13.13.13.1, Local port: 35018, IF Handle: 0x00000000
  Foreign host: 20.20.20.1, Foreign port: 179
  Last reset 00:00:00
```

- Verify if all the IP addresses are learnt on PE1 from PE2:

```
Router-PE1#show bgp vpnv4 unicast

BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 798487
BGP NSR Initial initsync version 15151 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2001:1601 (default for vrf vrf1601)
*> 20.13.1.1/32       192.13.26.5                            0 7501 i
*> 20.13.1.2/32       192.13.26.5                            0 7501 i
*> 20.13.1.3/32       192.13.26.5                            0 7501 i
*> 20.13.1.4/32       192.13.26.5                            0 7501 i
*> 20.13.1.5/32       192.13.26.5                            0 7501 i
*>i20.14.1.1/3214.14.14.1                    100      0 8501 i
*>i20.14.1.2/3214.14.14.1                    100      0 8501 i
*>i20.14.1.3/3214.14.14.1                    100      0 8501 i
*>i20.14.1.4/3214.14.14.1                    100      0 8501 i
*>i20.14.1.5/3214.14.14.1                    100      0 8501 i
```

Related Topics

- Configure the Core Network, on page 34
- Define VRFs on PE Routers to Enable Customer Connectivity, on page 41

For more details on Multiprotocol BGP, see *BGP Configuration Guide for Cisco NCS 5500 Series Routers*.

Associated Commands

- neighbor
- router-bgp
- update-source
- vrf
- show bgp

Connect MPLS VPN Customers

Connecting MPLS VPN customers involves these main tasks:

- Define VRFs on PE Routers to Enable Customer Connectivity, on page 41
- Configure VRF Interfaces on PE Routers for Each VPN Customer, on page 42

- Configure the Routing Protocol between the PE and CE Routers

  Use any of these options:

  - Configure BGP as the Routing Protocol Between the PE and CE Routers, on page 43
  - Configure Static Routes Between the PE and CE Routers, on page 49
  - Configure OSPF as the Routing Protocol Between the PE and CE Routers, on page 50

Define VRFs on PE Routers to Enable Customer Connectivity

VPN routing and forwarding (VRF) defines the VPN membership of a customer site attached to a PE router. A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A VRF contains all the routes available to the site from the VPNs of which it is a member. The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by BGP extended communities.

Configuration Example

This example configures a VRF instance (vrf1601) and specifies the import and export route-targets (2001:1601). The import route policy is the one that can be imported into the local VPN. The export route policy is the one that can be exported from the local VPN. The import route-target configuration allows exported VPN routes to be imported into the VPN if one of the route targets of the exported route matches one of the local VPN import route targets. When the route is advertised to other PE routers, the export route target is sent along with the route as an extended community.

```
Router-PE1#configure
Router-PE1(config)#vrf vrf1601
Router-PE1(config-vrf)#address-family ipv4 unicast
Router-PE1(config-vrf-af)#import route-target
Router-PE1(config-vrf-af-import-rt)#2001:1601
Router-PE1(config-vrf-af-import-rt)#exit
Router-PE1(config-vrf-af)#export route-target
Router-PE1(config-vrf-af-export-rt)#2001:1601
Router-PE1(config-vrf-af-export-rt)#commit
```

This VRF instance is then associated with the respective BGP instance.

Running Configuration

```
vrf vrf1601
 address-family ipv4 unicast
  import route-target
   2001:1601
  !
  export route-target
   2001:1601
  !
 !
!
```

Verification

Verify the import and export route targets.

```
Router-PE1#show vrf vrf1601
VRF                     RD            RT                   AFI    SAFI
vrf1601                 2001:1601
                                      import   2001:1601    IPV4   Unicast
                                      export   2001:1601    IPV4   Unicast
```

Related Topics

- Configure VRF Interfaces on PE Routers for Each VPN Customer, on page 42
- Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 37

Associated Commands

- import route-policy
- import route-target
- export route-policy
- export route-target
- vrf

Configure VRF Interfaces on PE Routers for Each VPN Customer

After a VRF instance is created, you must associate that VRF instance with an interface or a sub-interface on the PE routers.

Note You must remove the IPv4 or IPv6 addresses from an interface prior to assigning, removing, or changing an interface's VRF. If this is not done in advance, any attempt to change the VRF on an IP interface is rejected.

Configuration Example

This example assigns an IP address *192.13.26.6* to the interface (*HundredGigE0/0/0/14.1601*) on PE1 router and associates the VRF instance *vrf1601*, to that interface.

```
Router-PE1#configure
Router-PE1(config)#interface HundredGigE0/0/0/14.1601
Router-PE1(config-if)#vrf vrf1601
Router-PE1(config-if)#ipv4 address 192.13.26.6 255.255.255.252
Router-PE1(config-if)#encapsulation dot1q 1601
Router-PE1(config)#commit
```

Running Configuration

```
interface HundredGigE0/0/0/14.1601
```

```
 vrf vrf1601
 ipv4 address 192.13.26.6 255.255.255.252
 encapsulation dot1q 1601
!
```

Verification

- Verify that the interface with which the VRF is associated, is UP.

```
Router-PE1#show ipv4 vrf vrf1601 interface
interface HundredGigE0/0/0/14.1601 is Up, ipv4 protocol is Up
  Vrf is vrf1601 (vrfid 0x60000001)
  Internet address is 192.13.26.6/30
  MTU is 1518 (1500 is available to IP)
  Helper address is not set
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000001
```

Related Topics

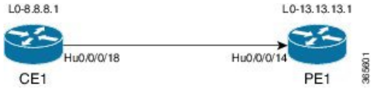
- Define VRFs on PE Routers to Enable Customer Connectivity, on page 41

Configure Routing Protocol Between the PE and CE Routers

Configure BGP as the Routing Protocol Between the PE and CE Routers

BGP distributes reachability information for VPN-IPv4 prefixes for each VPN. PE to PE or PE to route reflector (RR) sessions are iBGP sessions, and PE to CE sessions are eBGP sessions. PE to CE eBGP sessions can be directly or indirectly connected (eBGP multihop).

Figure 7: BGP as the Routing Protocol between PE and CE Routers



Configuration Example

This example lists the steps to configure BGP as the routing protocol between the PE and CE routers. The route policy, *pass-all* in this example, must be configured before it can be attached.

PE1:

```
Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 13.13.13.1
Router-PE1(config-bgp)#address-family ipv4 unicast
Router-PE1(config-bgp-af)#exit
```

```
Router-PE1(config-bgp)#address-family vpnv4 unicast
Router-PE1(config-bgp-af)#exit
/* VRF configuration */
Router-PE1(config-bgp)#vrf vrf1601
Router-PE1(config-bgp-vrf)#rd 2001:1601
Router-PE1(config-bgp-vrf)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-af)#label mode per-vrf
Router-PE1(config-bgp-vrf-af)#redistribute connected
Router-PE1(config-bgp-vrf-af)#exit
Router-PE1(config-bgp-vrf)#neighbor 192.13.26.5
Router-PE1(config-bgp-vrf-nbr)#remote-as 7501
Router-PE1(config-bgp-vrf-nbr)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-nbr-af)#route-policy pass-all in
Router-PE1(config-bgp-vrf-nbr-af)#route-policy pass-all out
Router-PE1(config-bgp-vrf-nbr-af)#commit
```

CE1:

```
Router-CE1#configure
Router-CE1(config)#router bgp 2001
Router-CE1(config-bgp)#bgp router-id 8.8.8.1
Router-CE1(config-bgp)#address-family ipv4 unicast
Router-CE1(config-bgp-af)#exit
Router-CE1(config-bgp)#address-family vpnv4 unicast
Router-CE1(config-bgp-af)#exit
Router-CE1(config-bgp)#neighbor 192.13.26.6
Router-CE1(config-bgp-nbr)#remote-as 2001
Router-CE1(config-bgp-nbr)#address-family ipv4 unicast
Router-CE1(config-bgp-nbr-af)#route-policy pass-all in
Router-CE1(config-bgp-nbr-af)#route-policy pass-all out
Router-CE1(config-bgp-nbr-af)#commit
```

Running Configuration

PE1:

```
router bgp 2001
 bgp router-id 13.13.13.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 vrf vrf1601
  rd 2001:1601
  address-family ipv4 unicast
   label mode per-vrf
   redistribute connected
  !
  neighbor 192.13.26.5
   remote-as 7501
   address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
 !
```

CE1:

```
router bgp 7501
 bgp router-id 8.8.8.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 192.13.26.6
  remote-as 2001
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
 !
!
```

Verification

- **PE1:**

```
Router-PE1#show bgp neighbor
BGP neighbor is 192.13.26.5
 Remote AS 6553700, local AS 2001, external link
 Administratively shut down
 Remote router ID 192.13.26.5
  BGP state = Established
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00
  Hold time is 60, keepalive interval is 20 seconds
  Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
  Last write 00:00:16, attempted 19, written 19
  Second last write 00:00:36, attempted 19, written 19
  Last write before reset 00:00:00, attempted 0, written 0
  Second last write before reset 00:00:00, attempted 0, written 0
  Last write pulse rcvd  Apr 12 10:31:20.739 last full not set pulse count 27939
  Last write pulse rcvd before reset 00:00:00
  Socket not armed for io, armed for read, armed for write
  Last write thread event before reset 00:00:00, second last 00:00:00
  Last KA expiry before reset 00:00:00, second last 00:00:00
  Last KA error before reset 00:00:00, KA not sent 00:00:00
  Last KA start before reset 00:00:00, second last 00:00:00
  Precedence: internet
  Non-stop routing is enabled
  Graceful restart is enabled
  Restart time is 120 seconds
  Stale path timeout time is 360 seconds
  Enforcing first AS is enabled
  Multi-protocol capability not received
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 30 secs
  Inbound message logging enabled, 3 messages buffered
  Outbound message logging enabled, 3 messages buffered

 For Address Family: IPv4 Unicast
  BGP neighbor version 0
  Update group: 0.2 Filter-group: 0.0  No Refresh request being processed
  Inbound soft reconfiguration allowed
  AF-dependent capabilities:
    Outbound Route Filter (ORF) type (128) Prefix:
      Send-mode: advertised
      Receive-mode: advertised
    Graceful Restart capability advertised
```

```
      Local restart time is 120, RIB purge time is 600 seconds
      Maximum stalepath time is 360 seconds
  Route refresh request: received 0, sent 0
  Policy for incoming advertisements is pass-all
  Policy for outgoing advertisements is pass-all
  0 accepted prefixes, 0 are bestpaths
  Cumulative no. of prefixes denied: 0.
  Prefix advertised 0, suppressed 0, withdrawn 0
  Maximum prefixes allowed 1048576
  Threshold for warning message 75%, restart interval 0 min
  An EoR was not received during read-only mode
  Last ack version 1, Last synced ack version 0
  Outstanding version objects: current 0, max 0
  Additional-paths operation: None
  Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with stitching-RT
 option
  Advertise VPNv6 routes is enabled with default option


  Connections established 1; dropped 0
  Local host: 192.13.26.6, Local port: 23456, IF Handle: 0x00000000
  Foreign host: 192.13.26.5, Foreign port: 179
  Last reset 03:12:58, due to Admin. shutdown (CEASE notification sent - administrative
 shutdown)
  Time since last notification sent to neighbor: 03:12:58
  Notification data sent:
    None
  External BGP neighbor not directly connected.
```

- **CE1:**

```
Router-CE1#show bgp neighbor
BGP neighbor is 192.13.26.6
 Remote AS 2001, local AS 6553700, external link
 Remote router ID 192.13.26.6
  BGP state = Established
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00
  Hold time is 60, keepalive interval is 20 seconds
  Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
  Last write 00:00:16, attempted 19, written 19
  Second last write 00:00:36, attempted 19, written 19
  Last write before reset 00:00:00, attempted 0, written 0
  Second last write before reset 00:00:00, attempted 0, written 0
  Last write pulse rcvd  Apr 12 10:31:20.739 last full not set pulse count 27939
  Last write pulse rcvd before reset 00:00:00
  Socket not armed for io, armed for read, armed for write
  Last write thread event before reset 00:00:00, second last 00:00:00
  Last KA expiry before reset 00:00:00, second last 00:00:00
  Last KA error before reset 00:00:00, KA not sent 00:00:00
  Last KA start before reset 00:00:00, second last 00:00:00
  Precedence: internet
  Non-stop routing is enabled
  Graceful restart is enabled
  Restart time is 120 seconds
  Stale path timeout time is 360 seconds
  Enforcing first AS is enabled
  Multi-protocol capability not received
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 30 secs
  Inbound message logging enabled, 3 messages buffered
  Outbound message logging enabled, 3 messages buffered
```

```
For Address Family: IPv4 Unicast
 BGP neighbor version 0
 Update group: 0.1 Filter-group: 0.0  No Refresh request being processed
 Inbound soft reconfiguration allowed
 AF-dependent capabilities:
   Outbound Route Filter (ORF) type (128) Prefix:
     Send-mode: advertised
     Receive-mode: advertised
   Graceful Restart capability advertised
     Local restart time is 120, RIB purge time is 600 seconds
     Maximum stalepath time is 360 seconds
 Route refresh request: received 0, sent 0
 Policy for incoming advertisements is pass-all
 Policy for outgoing advertisements is pass-all
 0 accepted prefixes, 0 are bestpaths
 Cumulative no. of prefixes denied: 0.
 Prefix advertised 0, suppressed 0, withdrawn 0
 Maximum prefixes allowed 1048576
 Threshold for warning message 75%, restart interval 0 min
 An EoR was not received during read-only mode
 Last ack version 1, Last synced ack version 0
 Outstanding version objects: current 0, max 0
 Additional-paths operation: None

 Connections established 0; dropped 0
 Local host: 192.13.26.5, Local port: 179, IF Handle: 0x00000000
 Foreign host: 192.13.26.6, Foreign port: 23456
 Last reset 00:00:00
 External BGP neighbor not directly connected.
```

Related Topics

- Connect MPLS VPN Customers, on page 40
- Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 37

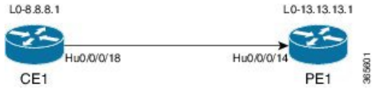
For more details on BGP, see *BGP Configuration Guide for Cisco NCS 5500 Series Routers*.

Associated Commands

- label mode
- neighbor
- rd
- redistribute
- remote-as
- route-policy
- router bgp

Configure RIPv2 as the Routing Protocol Between the PE and CE Routers

Figure 8: RIP as the Routing Protocol between PE and CE Routers



Configuration Example

This example lists the steps to configure RIPv2 as the routing protocol between the PE and CE routers. The VRF instance vrf1601 is configured in the router rip configuration mode and the respective interface (TenGigE0/0/0/14.1601 on PE1 and TenGigE0/0/0/18.1601 on CE1) is associated with that VRF. The **redistribute** option specifies routes to be redistributed into RIP.

PE1:

```
Router-PE1#configure
Router-PE1(config)#router rip
Router-PE1(config-rip)#vrf vrf1601
Router-PE1(config-rip-vrf)#interface TenGigE0/0/0/14.1601
Router-PE1(config-rip-vrf-if)#exit
Router-PE1(config-bgp-vrf)#redistribute bgp 2001
Router-PE1(config-bgp-vrf)#redistribute connected
Router-PE1(config-bgp-vrf)#commit
```

CE1:

```
Router-CE1#configure
Router-CE1(config)#router rip
Router-CE1(config-rip)#vrf vrf1601
Router-CE1(config-rip-vrf)#interface TenGigE0/0/0/14.1601
Router-CE1(config-rip-if)#exit
Router-CE1(config-rip)#redistribute connected
Router-CE1(config-rip)#commit
```

Running Configuration

PE1:

```
Router-PE1#show running-config router rip
router rip
 vrf vrf1601
  interface TenGigE0/0/0/14.1601
  !
  redistribute bgp 2001
  redistribute connected
 !
!
```

CE1:

```
Router-CE1#show running-config router rip
router rip
 vrf vrf1601
  interface TenGigE0/0/0/18.1601
  !
  redistribute connected
 !
!
```

Related Topics

- Connect MPLS VPN Customers, on page 40

Associated Commands

- redistribute
- router rip

Configure Static Routes Between the PE and CE Routers

Configuration Example

In this example, the static route is assigned to VRF, vrf1601.

```
Router-PE1#configure
Router-PE1(config)#router static
Router-PE1(config-static)#vrf vrf1601
Router-PE1(config-static-vrf)#address-family ipv4 unicast
Router-PE1(config-static-vrf-afi)#23.13.1.1/32 TenGigE0/0/0/14.1601 192.13.3.93
Router-PE1(config-static-vrf-afi)#commit
```

Repeat the configuration in CE1, with the respective interface values.

Running Configuration

PE1:

```
router static
 vrf vrf1601
  address-family ipv4 unicast
   23.13.1.1/32 TenGigE0/0/0/14.1601 192.13.3.93
  !
 !
!
```

CE1:

```
router static
 vrf vrf1601
  address-family ipv4 unicast
   23.8.1.2/32 TenGigE0/0/0/18.1601 192.8.3.94
  !
 !
!
```

Related Topics

- Connect MPLS VPN Customers, on page 40

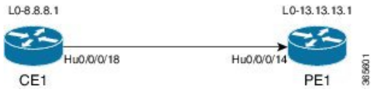
Associated Commands

- router static

Configure OSPF as the Routing Protocol Between the PE and CE Routers

You can use RIP, OSPF or ISIS as the routing protocol between the PE and CE routers.

Figure 9: OSPF as the Routing Protocol between PE and CE Routers



Configuration Example

This example lists the steps to configure PE-CE routing sessions that use OSPF routing protocol. A VRF instance *vrf1601* is configured in the **router ospf** configuration mode. The router-id for the OSPF process is 13.13.13.1. The **redistribute** option specifies routes to be redistributed into OSPF. The OSPF area is configured to be *1* and interface TenGigE0/0/0/14.1601 is associated with that area to enable routing on it.

PE1:

```
Router-PE1#configure
Router-PE1(config)#router ospf pe-ce-ospf-vrf
Router-PE1(config-ospf)#router-id 13.13.13.1
Router-PE1(config-ospf)#vrf vrf1601
Router-PE1(config-ospf-vrf)#redistribute connected
Router-PE1(config-ospf-vrf)#redistribute bgp 2001
Router-PE1(config-ospf-vrf)#area 1
Router-PE1(config-ospf-vrf-ar)#interface TenGigE0/0/0/14.1601
Router-PE1(config-ospf-vrf-ar)# commit
```

Repeat this configuration at PE2 node as well.

CE1:

```
Router-CE1#configure
Router-CE1(config)#router ospf ospf pe-ce-1
Router-CE1(config-ospf)#router-id 8.8.8.1
Router-CE1(config-ospf)#vrf vrf1601
Router-CE1(config-ospf-vrf)#area 1
Router-CE1(config-ospf-vrf-ar)#interface TenGigE0/0/0/18.1601
Router-CE1(config-ospf-vrf-ar)#commit
```

Running Configuration

PE1:

```
router ospf pe-ce-ospf-vrf
 router-id 13.13.13.1
 vrf vrf1601
  redistribute connected
  redistribute bgp 2001
  area 1
   interface TenGigE0/0/0/14.1601
   !
  !
 !
!
```

CE1:

```
router ospf pe-ce-1
 router-id 8.8.8.1
 vrf vrf1601
  area 1
   interface TenGigE0/0/0/18.1601
   !
  !
 !
!
```

Related Topics

- Connect MPLS VPN Customers, on page 40

Associated Commands

- router ospf

Verify MPLS L3VPN Configuration

You must verify these to ensure the successful configuration of MPLS L3VPN:

Verify the L3VPN Traffic Flow

- Verify the number of bytes switched for the label associated with the VRF (vrf1601):

  P node:

```
Router-P#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24119  Pop         20.20.20.1/32      Hu0/0/0/0    191.31.1.90      2170204180148
```

  PE2:

```
Router#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24031  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                                       vrf1601                      11124125835
```

Verify the Underlay (transport)

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
MPLS layer 3 VPNs CLI enhancements	Release 24.2.1	This feature enables precise hardware-level verification and troubleshooting of MPLS forwarding entries. You can now verify how Cisco Express Forwarding (CEF) entries are handled by the egress Network Processing Unit (NPU) on a specific line card, view the forwarding path, and check hardware resource usage for a given MPLS label and location. The **npu** keyword is introduced in the **show mpls forwarding labels** command to display the hardware forwarding details processed by the NPU.

- Verify if the LDP neighbor connection is established with the respective neighbor:

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
  TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
  LDP Discovery Sources:
    IPv4: (1)

    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.64.98.32    87.0.0.2       88.88.88.14    50.50.50.50
      178.0.0.1      192.1.1.1
    IPv6: (0)
```

- Verify if the label update is received by the FIB:

```
Router-PE1#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24036  Pop         16.16.16.1/32      Hu0/0/0/2    191.22.1.2      293294
```

```
24037  24165       18.18.18.1/32      Hu0/0/0/2    191.22.1.2      500

24039  24167       20.20.20.1/32      Hu0/0/0/2    191.22.1.2      17872433
       24167       20.20.20.1/32      Hu0/0/0/2.1  191.22.3.2      6345
24041  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                      vrf1601                      7950400999
```

- Verify if label is updated in the hardware:

Note Starting with Cisco IOS XR Release 24.2.1, the `npu` keyword is introduced in the **show mpls forwarding labels** command to verify Cisco Express Forwarding (CEF) entry information that are processed by the Network Processing Unit (NPU). The NPU handles packet forwarding and switching in hardware. Using the npu keyword focuses the command output on how the NPU processes these entries.

The following sample output is from the **show mpls forwarding labels** command without using **npu** keyword:

```
Router#show mpls forwarding labels 24000 hardware egress location 0/0/CPU0
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24000  24002       3.3.3.3/32         Hu0/0/0/10   10.1.1.2        N/A


 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
    PI:0x308e342928 PD:0x308e3429d0 rev:2014 type: MPLS (2) TBL: 0
    LEAF location: LEM
    FEC key: 0x1440000f7d0
    label action: MPLS_SWAP, dpa handle: 0x308e4672d8

    LWLDI:
        PI:0x308d252a08 PD:0x308d252a50 rev:2011 p-rev:2010  ldi type:IMP_EOS0_EOS1
        FEC key: 0x1540000f7d0 fec index: 0x2001ffd5(131029) num paths: 1 bkup paths:
0
        IMP pattern:3
        PI:0x308d252a08 PD:0x308d252a50 rev:2011 p-rev:2010  dpa-rev:4269924
        FEC key: 0x1540000f7d0 fec index: 0x2001ffd5(131029) num paths: 1 bkup paths:
0
        Path:0  fec index: 0x2001ffd5(131029) DSP: 0x9f
                MPLS encap key: 0xf1b0000040011841 MPLS encap id: 0x40011841 Remote: 0

                Label Stack(Top -> Bottom): { 24002 } dpa-rev: 4269923

        SHLDI:
            PI:0x308bec3698 PD:0x308bec37a8 rev:2010 dpa-rev:4269922 cbf_enabled:0
pbts_enabled:0 surpf_enable:0 flag:0x0
            FEC key: 0x1440000f7d0 fec index: 0x2001ffd6(131030) num paths: 1 bkup
paths: 0
            p-rev:1973
            Path:0 fec index: 0x2001ffd6(131030) DSP:0x9f Dest fec index: 0x0(0)
            TX-NHINFO:
                PI: 0x308ea81340 PD: 0x308ea813d0 rev:1973 dpa-rev:4269323 Encap hdl:
0x308eb21298
                Encap id: 0x40010001 Remote: 0 L3 int: 17 flags: 0x3
transport_encap_id:0x0
                npu_mask: 0x2 DMAC: ba:d6:11:de:61:d8
```

The following sample output is from the **show mpls forwarding labels** command with **npu** keyword:

```
Router#show mpls forwarding labels 24000 hardware egress npu location 0/0/CPU0
<<<<<<<<<<<< Show output with NPU keyword showing additional data from NPU hardware

Using NPU option is resource intensive and may result in system
instability and possibly result in traffic loss.
Do you really want to continue[confirm with only 'y' or 'n'] [y/n] :Local  Outgoing
 Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24000  24002       3.3.3.3/32         Hu0/0/0/10   10.1.1.2        N/A


 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
    PI:0x308e342928 PD:0x308e3429d0 rev:2014 type: MPLS (2) TBL: 0
    LEAF location: LEM
    FEC key: 0x1440000f7d0
    label action: MPLS_SWAP, dpa handle: 0x308e4672d8
    mplslabel HW:
    npu:0x0 out_label:0x5dc2 fec:0x2001ffd6  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
    npu:0x1 out_label:0x5dc2 fec:0x2001ffd6
    npu:0x2 out_label:0x5dc2 fec:0x2001ffd6
    npu:0x3 out_label:0x5dc2 fec:0x2001ffd6

    LWLDI:
        PI:0x308d252a08 PD:0x308d252a50 rev:2011 p-rev:2010  ldi type:IMP_EOS0_EOS1
        FEC key: 0x1540000f7d0 fec index: 0x2001ffd5(131029) num paths: 1 bkup paths:
0
        IMP pattern:3
        PI:0x308d252a08 PD:0x308d252a50 rev:2011 p-rev:2010  dpa-rev:4269924
        FEC key: 0x1540000f7d0 fec index: 0x2001ffd5(131029) num paths: 1 bkup paths:
0
        Path:0  fec index: 0x2001ffd5(131029) DSP: 0x9f
                MPLS encap key: 0xf1b0000040011841 MPLS encap id: 0x40011841 Remote: 0

                Label Stack(Top -> Bottom): { 24002 } dpa-rev: 4269923
        FEC:
        npu:0x0 fec:0x2001ffd5 port:0x8001811 encap:0x40011841
        npu:0x1 fec:0x2001ffd5 port:0x8001811 encap:0x40011841
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
        npu:0x2 fec:0x2001ffd5 port:0x8001811 encap:0x40011841
        npu:0x3 fec:0x2001ffd5 port:0x8001811 encap:0x40011841
        mplsnh HW:
            npu:0x1 mpls_encap_id:0x40011841 label1:0x5dc2 action:0x2

        SHLDI:
            PI:0x308bec3698 PD:0x308bec37a8 rev:2010 dpa-rev:4269922 cbf_enabled:0
pbts_enabled:0 surpf_enable:0 flag:0x0
            FEC key: 0x1440000f7d0 fec index: 0x2001ffd6(131030) num paths: 1 bkup
paths: 0
            p-rev:1973
            Path:0 fec index: 0x2001ffd6(131030) DSP:0x9f Dest fec index: 0x0(0)
            FEC:
            npu:0x0 fec:0x2001ffd6 port:0x8001811 encap:0x40010001
            npu:0x1 fec:0x2001ffd6 port:0x8001811 encap:0x40010001
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
            npu:0x2 fec:0x2001ffd6 port:0x8001811 encap:0x40010001
            npu:0x3 fec:0x2001ffd6 port:0x8001811 encap:0x40010001

            TX-NHINFO:
                PI: 0x308ea81340 PD: 0x308ea813d0 rev:1973 dpa-rev:4269323 Encap hdl:
0x308eb21298
                Encap id: 0x40010001 Remote: 0 L3 int: 17 flags: 0x3
transport_encap_id:0x0
                npu_mask: 0x2 DMAC: ba:d6:11:de:61:d8
                ENCAP:
             npu:0x1 mac:0xd861de11d6ba vlan:0x11  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

Verify the Overlay (L3VPN)

Imposition Path

- Verify if the BGP neighbor connection is established with the respective neighbor node:

```
Router-PE1#show bgp summary
BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 18003
BGP main routing table version 18003
BGP NSR Initial initsync version 3 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker           18003      18003      18003      18003       18003           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
21.21.21.1        0  2001   19173    7671    18003    0    0    1d07h       4000
192.13.2.149      0  7001    4615    7773    18003    0    0 09:26:21        125
```

- Verify if BGP routes are advertised and learnt:

```
Router-PE1#show bgp vpnv4 unicast
BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 305345
BGP NSR Initial initsync version 12201 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2001:1601 (default for vrf vrf1601)
*> 20.13.1.1/32       192.13.26.5                            0 7501 i
*> 20.13.1.2/32       192.13.26.5                            0 7501 i
*>i20.23.1.1/32       20.20.20.1                    100      0 6553700 11501 i
*>i20.23.1.2/32       20.20.20.1                    100      0 6553700 11501 i
```

- Verify BGP labels:

```
Router-PE1#show bgp label table
Label   Type              VRF/RD         Context
24041   IPv4 VRF Table   vrf1601         -
24042   IPv4 VRF Table   vrf1602         -
```

- Verify if the route is downloaded in the respective VRF:

```
Router-PE1#show cef vrf vrf1601 20.23.1.1
20.23.1.1/32, version 743, internal 0x5000001 0x0 (ptr 0x8f932174) [1], 0x0 (0x8fa99990),
 0xa08 (0x8f9fba58)
 Updated Apr 20 12:33:47.840
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 20.20.20.1/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x8c0e3148 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 20.20.20.1/32 via 24039/0/21
    next hop 191.23.1.2/32 Hu0/0/1/1    labels imposed {24059 24031}
```

Disposition Path

- Verify if the imposition and disposition labels are assigned and label bindings are exchanged for L3VPN prefixes:

```
Router-PE2#show mpls lsd forwarding
In_Label, (ID), Path_Info: <Type>
24030, (IPv4, 'default':4U, 13.13.13.1/32), 5 Paths
   1/1: IPv4, 'default':4U, Hu0/0/0/19.2, nh=191.31.1.93, lbl=24155,
            flags=0x0, ext_flags=0x0
24031, (VPN-VRF, 'vrf1601':4U), 1 Paths
   1/1: PopLkup-v4, 'vrf1601':4U, ipv4
24032, (VPN-VRF, 'vrf1602':4U), 1 Paths
   1/1: PopLkup-v4, 'vrf1602':4U, ipv4
```

- Verify if the label update is received by the FIB:

```
Router-PE2#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------

24019  Pop         18.18.18.3/32      Hu0/0/0/19   191.31.1.89     11151725032

24030  24155       13.13.13.1/32      Hu0/0/0/19   191.31.1.89     3639895

24031  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                             vrf1601                     32167647049
```

Providing VPN Connectivity Across Multiple Autonomous Systems with MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

Note This section is not applicable to Inter-AS over IP tunnels.

This section contains instructions for the following tasks:

Configuring ASBRs to Exchange IPv4 Routes and MPLS Labels

This example shows how to configure the autonomous system boundary routers (ASBRs) to exchange IPv4 routes and MPLS labels.

Configuration Example

```
Router# configure
Router(config)#router bgp 500
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#neighbor 16.1.1.1
Router(config-bgp-nbr)#remote-as 100
Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#commit
```

Running Configuration

```
router bgp 500
bgp router-id 60.200.11.1
address-family ipv4 unicast
  allocate-label all
!
neighbor 16.1.1.1
  remote-as 100
  address-family ipv4 labeled-unicast
   route-policy PASS-ALL in
   route-policy pass-all out
  !
!
```

Verification

```
Router#show bgp ipv4 labeled-unicast

BGP router identifier 60.200.11.1, local AS number 500
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 10
BGP main routing table version 10
BGP NSR Initial initsync version 6 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 10.200.1.1/32      16.1.1.1                 0             0 100 ?
*                     66.161.1.1               0             0 100 ?
*> 10.200.2.1/32      16.1.1.1                 5             0 100 ?
*                     66.161.1.1               5             0 100 ?
*> 10.200.5.1/32      16.1.1.1                11             0 100 ?
*                     66.161.1.1              11             0 100 ?
*> 10.200.6.1/32      16.1.1.1                 4             0 100 ?
*                     66.161.1.1               4             0 100 ?
*> 60.200.11.1/32     0.0.0.0                  0         32768 ?
*>i60.200.12.1/32     60.200.12.1              0    100      0 ?
*>i60.200.13.1/32     60.200.13.1              0    100      0 ?

Router#show bgp ipv4 labeled-unicast 10.200.1.1

BGP routing table entry for 10.200.1.1/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 31          31
    Local Label: 64006

Paths: (2 available, best #1)
  Advertised to peers (in unique update groups):
    60.200.12.1
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    60.200.12.1
  100
    16.1.1.1 from 16.1.1.1 (10.200.1.1)
      Received Label 3
      Origin incomplete, metric 0, localpref 100, valid, external, best, group-best,
multipath, labeled-unicast
      Received Path ID 0, Local Path ID 0, version 31
      Origin-AS validity: not-found

Router#show cef vrf default ipv4 10.200.1.1
10.200.1.1/32, version 161, internal 0x5000001 0x0 (ptr 0x8910c440) [1], 0x0 (0x87f73bc0),
 0xa00 (0x88f40118)
 Updated May  3 18:10:47.034
 Prefix Len 32, traffic index 0, precedence n/a, priority 4
 Extensions: context-label:64006
   via 16.1.1.1/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
    path-idx 0 NHID 0x0 [0x889e55a0 0x87b494b0]
    recursion-via-/32
    next hop 16.1.1.1/32 via 16.1.1.1/32
     local label 64006
     next hop 16.1.1.1/32 Te0/0/1/4/2  labels imposed {ImplNull ImplNull}
   via 66.161.1.1/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
    path-idx 1 NHID 0x0 [0x89113870 0x87b493e8]
    recursion-via-/32
    next hop 66.161.1.1/32 via 66.161.1.1/32
     local label 64006
     next hop 66.161.1.1/32 BE161        labels imposed {ImplNull ImplNull}
Router#
```

Associated Commands

- allocate-label all

- address-family ipv4 labeled-unicast

Configuring the Route Reflectors to Exchange VPN-IPv4 Routes

This example shows how to configure the route reflectors to exchange VPN-IPv4 routes by using multihop. This task specifies that the next-hop information and the VPN label are to be preserved across the autonomous system (AS).

Configuration Example

```
Router# configure
Router(config)# router bgp 500
Router(config-bgp)# neighbor 10.200.2.1
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# ebgp-multihop 255
Router(config-bgp-nbr)# update-source loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-unchanged
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-unchanged
```

Running Configuration

```
Router#show run router bgp 500
router bgp 500
bgp router-id 60.200.13.1
address-family ipv4 labeled-unicast
  allocate-label all
!
address-family vpnv4 unicast
!
address-family ipv6 unicast
!
address-family vpnv6 unicast
!
neighbor 10.200.2.1
  remote-as 100
  ebgp-multihop 255
  update-source Loopback0
  address-family vpnv4 unicast
   route-policy PASS-ALL in
   route-policy PASS-ALL out
   next-hop-unchanged
  !
  address-family vpnv6 unicast
   route-policy PASS-ALL in
   route-policy PASS-ALL out
   next-hop-unchanged
  !
```

Verification

```
Router#show cef vrf vrf2001 ipv4 111.1.1.2/32 hardware egress location0/0/CPU00/RP0/CPU0
111.1.1.2/32, version 39765, internal 0x5000001 0x0 (ptr 0x9f4d326c) [1], 0x0 (0xa0263058),
 0x808 (0x899285b8)
 Updated Oct 27 10:58:39.350
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
```

```
   via 10.200.1.1/32, 307 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0x89a59100 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 10.200.1.1/32 via 69263/0/21
     next hop 63.13.1.1/32 Te0/3/0/17/0 labels imposed {24007 64007 64023}


 LEAF - HAL pd context :
 sub-type : IPV4, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0
HW Walk:
LEAF:
    PI:0x9f4d326c PD:0x9f4d3304 Rev:3865741 type: 0
    FEC handle: 0x890c0198

    LWLDI:
        PI:0xa0263058 PD:0xa0263098 rev:3865740 p-rev: ldi type:0
        FEC hdl: 0x890c0198 fec index: 0x0(0) num paths:1, bkup: 0

 REC-SHLDI HAL PD context :
ecd_marked:0, collapse_bwalk_required:0, load_shared_lb:0

    RSHLDI:
        PI:0x9f17bfd8 PD:0x9f17c054 rev:0 p-rev:0 flag:0x1
        FEC hdl: 0x890c0198 fec index: 0x20004fa6(20390) num paths: 1
        Path:0 fec index: 0x20004fa6(20390) DSP fec index: 0x2000120e(4622)
               MPLS Encap Id: 0x4001381e

 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0
HW Walk:
LEAF:
    PI:0x89a59100 PD:0x89a59198 Rev:3864195 type: 2
    FEC handle: (nil)

    LWLDI:
        EOS0/1 LDI:
        PI:0xb9a51838 PD:0xb9a51878 rev:3864192 p-rev: ldi type:0
        FEC hdl: 0x890c0818 fec index: 0x20004fa2(20386) num paths:1, bkup: 0
        DSP fec index:0x2000120e(4622)
        Path:0  fec index: 0x20004fa2(20386) DSP fec index:0x2000120e(4622)
                MPLS encap hdl: 0x400145ed MPLS encap id: 0x400145ed Remote: 0
        IMP LDI:
        PI:0xb9a51838 PD:0xb9a51878 rev:3864192 p-rev:
        FEC hdl: 0x890c0b58 fec index: 0x20004fa0(20384) num paths:1
        Path:0  fec index: 0x20004fa0(20384) DSP fec index: 0x2000120e(4622)
                MPLS encap hdl: 0x400145ec MPLS encap id: 0x400145ec Remote: 0

 REC-SHLDI HAL PD context :
ecd_marked:0, collapse_bwalk_required:0, load_shared_lb:0

    RSHLDI:
        PI:0xb7e387f8 PD:0xb7e38874 rev:0 p-rev:0 flag:0x1
        FEC hdl: 0x890c0e98 fec index: 0x20004f9e(20382) num paths: 1
        Path:0 fec index: 0x20004f9e(20382) DSP fec index: 0x2000120e(4622)

 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0
HW Walk:
LEAF:
    PI:0x89a59028 PD:0x89a590c0 Rev:31654 type: 2
```

```
FEC handle: (nil)

LWLDI:
    PI:0x8c69c1c8 PD:0x8c69c208 rev:31653 p-rev:31652  ldi type:5
    FEC hdl: 0x8903a718 fec index: 0x0(0) num paths:1, bkup: 0
    Path:0  fec index: 0x0(0) DSP:0x0
    IMP LDI:
    PI:0x8c69c1c8 PD:0x8c69c208 rev:31653 p-rev:31652
    FEC hdl: 0x8903aa58 fec index: 0x2000120e(4622) num paths:1
    Path:0  fec index: 0x2000120e(4622) DSP:0x518
            MPLS encap hdl: 0x40013808 MPLS encap id: 0x40013808 Remote: 0

    SHLDI:
        PI:0x8af02580 PD:0x8af02600 rev:31652 dpa-rev:66291 flag:0x0
        FEC hdl: 0x8903a718 fec index: 0x2000120d(4621) num paths: 1 bkup paths: 0
        p-rev:2373
        Path:0 fec index: 0x2000120d(4621) DSP:0x518 Dest fec index: 0x0(0)

    TX-NHINFO:
        PD: 0x89bf94f0 rev: 2373 dpa-rev: 9794 Encap hdl: 0x8a897628
        Encap id: 0x40010002 Remote: 0 L3 int: 1043 npu_mask: 4
```

Associated Commands

- address-family vpnv4 unicast
- allocate-label all
- ebgp-multihop
- next-hop-unchanged

Configure the Route Reflectors to Reflect Remote Routes in its AS

This example shows how to enable the route reflector (RR) to reflect the IPv4 routes and labels learned by the autonomous system boundary router (ASBR) to the provider edge (PE) routers in the autonomous system. This task is accomplished by making the ASBR and PE as the route reflector clients of the RR.

Configuration Example

```
Router#configure
Router(config)#router bgp 500
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#neighbor 60.200.11.1
Router(config-bgp-nbr)#remote-as 500
Router(config-bgp-nbr)#update-source loopback0
Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#route-reflector-client
Router(config-bgp-nbr-af)#neighbor 60.200.12.1
Router(config-bgp-nbr)#remote-as 500
Router(config-bgp-nbr)#update-source loopback0
Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#route-reflector-client
Router(config-bgp-nbr)#address-family vpnv4 unicast
Router(config-bgp-nbr-af)#route-reflector-client
```

Running Configuration

```
Router#show run router bgp 500
router bgp 500
 bgp router-id 60.200.13.1
 address-family ipv4 unicast
  allocate-label all
 !
 address-family vpnv4 unicast
 !
 neighbor 60.200.11.1
  remote-as 500
  update-source Loopback0
  !
  address-family ipv4 labeled-unicast
   route-reflector-client
  !
  address-family vpnv4 unicast
  !
 !
 neighbor 60.200.12.1
  remote-as 500
  update-source Loopback0
  address-family ipv4 labeled-unicast
   route-reflector-client
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
```

Providing VPN Connectivity Across Multiple Autonomous Systems with MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

This section contains instructions for the following tasks:

Configuring the ASBRs to Exchange VPN-IPv4 Addresses for IP Tunnels

Perform this task to configure an external Border Gateway Protocol (eBGP) autonomous system boundary router (ASBR) to exchange VPN-IPv4 routes with another autonomous system.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters the XR Config mode.

Step 2 **router bgp** *autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config)# router bgp 120
RP/0/RP0/CPU0:router(config-bgp)#
```

Enters Border Gateway Protocol (BGP) configuration mode allowing you to configure the BGP routing process.

Step 3 **address-family** { **ipv4 tunnel** }

Example:

```
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 tunnel
RP/0/RP0/CPU0:router(config-bgp-af)#
```

Configures IPv4 tunnel address family.

Step 4 **address-family** { **vpnv4 unicast** }

Example:

```
RP/0/RP0/CPU0:router(config-bgp-af)# address-family vpnv4 unicast
```

Configures VPNv4 address family.

Step 5 **neighbor** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-af)# neighbor 172.168.40.24
RP/0/RP0/CPU0:router(config-bgp-nbr)#
```

Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address 172.168.40.24 as an ASBR eBGP peer.

Step 6 **remote-as** *autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 2002
```

Creates a neighbor and assigns it a remote autonomous system number.

Step 7 **address-family** { **vpnv4 unicast** }

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)#
```

Configures VPNv4 address family.

Step 8 **route-policy** *route-policy-name* { **in** }

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
```

Applies a routing policy to updates that are received from a BGP neighbor.

- Use the *route-policy-name* argument to define the name of the of route policy. The example shows that the route policy name is defined as pass-all.
- Use the **in** keyword to define the policy for inbound routes.

Step 9 **route-policy** *route-policy-name* { **out** }

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
```

Applies a routing policy to updates that are sent from a BGP neighbor.

- Use the *route-policy-name* argument to define the name of the route policy. The example shows that the route policy name is defined as pass-all.
- Use the **out** keyword to define the policy for outbound routes.

Step 10 **neighbor** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# neighbor 175.40.25.2
RP/0/RP0/CPU0:router(config-bgp-nbr)#
```

Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address 175.40.25.2 as an VPNv4 iBGP peer.

Step 11 **remote-as** *autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 2002
```

Creates a neighbor and assigns it a remote autonomous system number.

Step 12 **update-source** *type interface-path-id*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# update-source loopback0
```

Allows BGP sessions to use the primary IP address from a particular interface as the local address.

Step 13 **address-family** { **ipv4 tunnel** }

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 tunnel
RP/0/RP0/CPU0:router(config-bgp-nbr-af)#
```

Configures IPv4 tunnel address family.

Step 14 **address-family** { **vpnv4 unicast** }

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# address-family vpnv4 unicast
```

Configures VPNv4 address family.

Step 15 Use the **commit** or **end** command.

commit - Saves the configuration changes and remains within the configuration session.

end - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

Configuring a Static Route to an ASBR Peer

Perform this task to configure a static route to an ASBR peer.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters the XR Config mode.

Step 2 **router static**

Example:

```
RP/0/RP0/CPU0:router(config)# router static
RP/0/RP0/CPU0:router(config-static)#
```

Enters router static configuration mode.

Step 3 **address-family ipv4 unicast**

Example:

```
RP/0/RP0/CPU0:router(config-static)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-static-afi)#
```

Enables an IPv4 address family.

Step 4 **A.B.C.D/length** *next-hop*

Example:

```
RP/0/RP0/CPU0:router(config-static-afi)# 10.10.10.10/32 10.9.9.9
```

Enters the address of the destination router (including IPv4 subnet mask).

Step 5 Use the **commit** or **end** command.

commit - Saves the configuration changes and remains within the configuration session.

end - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.

- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

Configuring EBGP Routing to Exchange VPN Routes Between Subautonomous Systems in a Confederation

Perform this task to configure external Border Gateway Protocol (eBGP) routing to exchange VPN routes between subautonomous systems in a confederation.

Note To ensure that host routes for VPN-IPv4 eBGP neighbors are propagated (by means of the Interior Gateway Protocol [IGP]) to other routers and PE routers, specify the **redistribute connected** command in the IGP configuration portion of the confederation eBGP (CEBGP) router. If you are using Open Shortest Path First (OSPF), make sure that the OSPF process is not enabled on the CEBGP interface in which the “redistribute connected” subnet exists.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters XR Config mode.

Step 2 **router bgp** *autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config)# router bgp 120
RP/0/RP0/CPU0:router(config-bgp)#
```

Enters BGP configuration mode allowing you to configure the BGP routing process.

Step 3 **bgp confederation peers** *peer autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config-bgp)# bgp confederation peers 8
```

Configures the peer autonomous system number that belongs to the confederation.

Step 4 **bgp confederation identifier** *autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config-bgp)# bgp confederation identifier 5
```

Specifies the autonomous system number for the confederation ID.

Step 5 **address-family vpnv4 unicast**

Example:

```
RP/0/RP0/CPU0:router(config-bgp)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)#
```

Configures VPNv4 address family.

Step 6 **neighbor** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-af)# neighbor 10.168.40.24
RP/0/RP0/CPU0:router(config-bgp-nbr)#
```

Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address 10.168.40.24 as a BGP peer.

Step 7 **remote-as** *autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 2002
```

Creates a neighbor and assigns it a remote autonomous system number.

Step 8 **address-family vpnv4 unicast**

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)#
```

Configures VPNv4 address family.

Step 9 **route-policy** *route-policy-name* **in**

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy In-Ipv4 in
```

Applies a routing policy to updates received from a BGP neighbor.

Step 10 **route-policy** *route-policy-name* **out**

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy Out-Ipv4 out
```

Applies a routing policy to updates advertised to a BGP neighbor.

Step 11 **next-hop-self**

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# next-hop-self
```

Disables next-hop calculation and let you insert your own address in the next-hop field of BGP updates.

Step 12 Use the **commit** or **end** command.

commit - Saves the configuration changes and remains within the configuration session.

end - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

Configuring MPLS Forwarding for ASBR Confederations

Perform this task to configure MPLS forwarding for autonomous system boundary router (ASBR) confederations (in BGP) on a specified interface.

Note This configuration adds the implicit NULL rewrite corresponding to the peer associated with the interface, which is required to prevent BGP from automatically installing rewrites by LDP (in multihop instances).

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters XR Config mode.

Step 2 **router bgp** *as-number*

Example:

```
RP/0/RP0/CPU0:router(config)# router bgp 120
RP/0/RP0/CPU0:router(config-bgp)
```

Enters BGP configuration mode allowing you to configure the BGP routing process.

Step 3 **mpls activate**

Example:

```
RP/0/RP0/CPU0:router(config-bgp)# mpls activate
RP/0/RP0/CPU0:router(config-bgp-mpls)#
```

Enters BGP MPLS activate configuration mode.

Step 4 **interface** *type interface-path-id*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-mpls)# interface GigabitEthernet 0/3/0/0hundredGigE 0/9/0/0
```

Enables MPLS on the interface.

Step 5 Use the **commit** or **end** command.

commit - Saves the configuration changes and remains within the configuration session.

end - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

Configuring a Static Route to an ASBR Confederation Peer

Perform this task to configure a static route to an Inter-AS confederation peer.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters XR Config mode.

Step 2 **router static**

Example:

```
RP/0/RP0/CPU0:router(config)# router static
RP/0/RP0/CPU0:router(config-static)#
```

Enters router static configuration mode.

Step 3 **address-family ipv4 unicast**

Example:

```
RP/0/RP0/CPU0:router(config-static)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-static-afi)#
```

Enables an IPv4 address family.

Step 4 **A.B.C.D/length** *next-hop*

Example:

```
RP/0/RP0/CPU0:router(config-static-afi)# 10.10.10.10/32 10.9.9.9
```

Enters the address of the destination router (including IPv4 subnet mask).

Step 5 Use the **commit** or **end** command.

commit - Saves the configuration changes and remains within the configuration session.

end - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

VRF-lite

VRF-lite is the deployment of VRFs without MPLS. VRF-lite allows a service provider to support two or more VPNs with overlapping IP addresses. With this feature, multiple VRF instances can be supported in customer edge devices.

VRF-lite interfaces must be Layer 3 interface and this interface cannot belong to more than one VRF at any time. Multiple interfaces can be part of the same VRF, provided all of them participate in the same VPN.

Configure VRF-lite

Consider two customers having two VPN sites each, that are connected to the same PE router. VRFs are used to create a separate routing table for each customer. We create one VRF for each customer (say, vrf1 and vrf2) and then add the corresponding interfaces of the router to the respective VRFs. Each VRF has its own routing table with the interfaces configured under it. The global routing table of the router does not show these interfaces, whereas the VRF routing table shows the interfaces that were added to the VRF. PE routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP or RIP.

To summarize, VRF-lite configuration involves these main tasks:

- Create VRF
- Configure VRF under the interface
- Configure VRF under routing protocol

Configuration Example

- **Create VRF:**

```
Router#configure
Router(config)#vrf vrf1
Router(config-vrf)#address-family ipv4 unicast

/* You must create route-policy pass-all before this configuration */
Router(config-vrf-af)#import from default-vrf route-policy pass-all
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#100:100
Router(config-vrf-import-rt)#exit
Router(config-vrf-af)#export route-target
Router(config-vrf-import-rt)#100:100
Router(config-vrf-import-rt)#exit
Router(config-vrf-import-rt)#commit
```

  Similarly create vrf2, with route-target as 100:100.

- **Configure VRF under the interface:**

```
Router#configure
Router(config)#interface TenGigE0/0/0/0.2001
Router(config-subif)#vrf vrf1
Router(config-subif)#ipv4 address 192.0.2.2 255.255.255.252
Router(config-subif)#encapsulation dot1q 2001
Router(config-subif)#exit

Router(config)#interface TenGigE0/0/0/0.2000
Router(config-subif)#vrf vrf2
Router(config-subif)#ipv4 address 192.0.2.5/30 255.255.255.252
Router(config-subif)#encapsulation dot1q 2000
Router(config-vrf-import-rt)#commit
```

  Similarly configure vrf1 under interface TenGigE0/0/0/1.2001 and vrf2 under interface TenGigE0/0/0/1.2000 TenGigE0/0/0/0.2001 and vrf2 under interface TenGigE0/0/0/0.2000

- **Configure VRF under routing protocol**:

```
Router#configure
Router(config)#router rip
Router(config-rip)#vrf vrf1
Router(config-rip-vrf)#interface TenGigE0/0/0/0.2001
Router(config-rip-vrf-if)#exit
Router(config-rip-vrf)#interface TenGigE0/0/0/1.2001
Router(config-rip-vrf-if)#exit
Router(config-rip-vrf)#default-information originate
Router(config-vrf-import-rt)#commit
```

  Similarly configure vrf2 under rip, with interface TenGigE0/0/0/0.2000 and interface TenGigE0/0/0/1.2000

Running Configuration

```
/* VRF Configuration */
```

```
vrf vrf1
address-family ipv4 unicast
  import route-target
   100:100
  !
  export route-target
   100:100
  !
!
!
vrf vrf2
address-family ipv4 unicast
  import route-target
   100:100
  !
  export route-target
   100:100
  !
!
!


/* Interface Configuration */

interface TenGigE0/0/0/0.2001
vrf vrf1
ipv4 address 192.0.2.2 255.255.255.252
encapsulation dot1q 2001
!

interface TenGigE0/0/0/0.2000
vrf vrf2
ipv4 address 192.0.2.5/30 255.255.255.252
encapsulation dot1q 2000
!

interface TenGigE0/0/0/1.2001
vrf vrf1
ipv4 address 203.0.113.2 255.255.255.252
encapsulation dot1q 2001
!

interface TenGigE0/0/0/1.2000
vrf vrf2
ipv4 address 203.0.113.5 255.255.255.252
encapsulation dot1q 2000
!

/* Routing Protocol Configuration */
router rip
interface Loopback0
!
interface TenGigE0/0/0/0
!
interface TenGigE0/0/0/0.2000
!
interface TenGigE0/0/0/0.2001
!
interface TenGigE0/0/0/1
!
interface TenGigE0/0/0/1.2000
!
interface TenGigE0/0/0/1.2001
!
```

```
vrf vrf1
  interface TenGigE0/0/0/0.2001
  !
  interface TenGigE0/0/0/1.2001
  !
  default-information originate
 !
vrf vrf2
  interface TenGigE0/0/0/1.2000
  !
  interface TenGigE0/0/0/1.2000
  !
  default-information originate
 !
```

Verification

```
Router#show route vrf vrf1
Mon Jul  4 19:12:54.739 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       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, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

C    203.0.113.0/24 is directly connected, 00:07:01, TenGigE0/0/0/1.2001
L    203.0.113.2/30 is directly connected, 00:07:01, TenGigE0/0/0/1.2001
C    192.0.2.0/24 is directly connected, 00:05:51, TenGigE0/0/0/1.2001
L    192.0.2.2/30 is directly connected, 00:05:51, TenGigE0/0/0/1.2001


Router#show route vrf vrf2
Mon Jul  4 19:12:59.121 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       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, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

R    198.51.100.53/30 [120/1] via 192.0.2.1, 00:01:42, TenGigE0/0/0/0.2000
C    203.0.113.0/24 is directly connected, 00:08:43, TenGigE0/0/0/1.2000
L    203.0.113.5/30 is directly connected, 00:08:43, TenGigE0/0/0/1.2000
C    192.0.2.0/24 is directly connected, 00:06:17, TenGigE0/0/0/0.2000
L    192.0.2.5/30 is directly connected, 00:06:17, TenGigE0/0/0/0.2000
```

Related Topics

- VRF-lite, on page 70

Associated Commands

- import route-target
- export route-target
- vrf

MPLS L3VPN Services using Segment Routing

Currently, MPLS Label Distribution Protocol (LDP) is the widely used transport for MPLS L3VPN services. The user can achieve better resilience and convergence for the network traffic, by transporting MPLS L3VPN services using Segment Routing (SR), instead of MPLS LDP. Segment routing can be directly applied to the MPLS architecture without changing the forwarding plane. In a segment-routing network using the MPLS data plane, LDP or other signaling protocol is not required; instead label distribution is performed by IGP (IS-IS or OSPF) or BGP protocol. Removing protocols from the network simplifies its operation and makes it more robust and stable by eliminating the need for protocol interaction. Segment routing utilizes the network bandwidth more effectively than traditional MPLS networks and offers lower latency.

Configure MPLS L3VPN over Segment Routing

Topology

Given below is a network scenario, where MPLS L3VPN service is transported using Segment Routing.

In this topology, CE1 and CE2 are the two customer routers. ISP has two PE routers, PE1 and PE2 and a P router. RIP is used for the edge protocol support between the CE and PE routers. Label distribution can be performed by IGP (IS-IS or OSPF) or BGP. OSPF is used in this scenario.

Customer's autonomous system is 65534, which peers with ISP's autonomous system 65000. This must be a vrf peering to prevent route advertisement into the global IPv4 table. The ISP routers PE1 and PE2 contain the VRF (for example, vrf1601) for the customer. PE1 and PE2 export and import the same route targets, although this is not necessary.

Loopback interfaces are used in this topology to simulate the attached networks.

Configuration

You must complete these tasks to ensure the successful configuration of MPLS L3VPN over segment routing:

- Configure protocol support on PE-CE (refer, Connect MPLS VPN Customers, on page 40)
- Configure protocol support on PE-PE (refer, Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 37)

Configure Segment Routing in MPLS Core

This section takes you through the configuration procedure to enable segment routing in MPLS core. You must perform this configuration in PE1, P and PE2 routers in the topology, using the corresponding values.

Configuration Example

```
/* Configure Segment Routing using OSFP */

Router-PE1#configure
Router-PE1(config)# router ospf dc-sr
Router-PE1(config-ospf)#router-id 13.13.13.1
Router-PE1(config-ospf)#segment routing mpls
Router-PE1(config-ospf)#segment routing forwarding mpls
Router-PE1(config-ospf)#mpls ldp sync
Router-PE1(config-ospf)#mpls ldp auto-config
Router-PE1(config-ospf)#segment-routing mpls
Router-PE1(config-ospf)#segment-routing mpls sr-prefer
Router-PE1(config-ospf)#segment-routing prefix-sid-map advertise-local
Router-PE1(config-ospf)#exit
Router-PE1(config-ospf)#area 1
Router-PE1(config-ospf-ar)#interface HundredGigE0/0/0/2
Router-PE1(config-ospf-ar-if)#exit
Router-PE1(config-ospf-ar)#interface Loopback0
Router-PE1(config-ospf-ar-if)#prefix-sid index 1
Router-PE1(config-ospf-ar-if)#commit

/ * Configure segment routing global block */

Router# configure
Router(config)# segment-routing
Router(config-sr)# global-block 180000 200000
Router(config-sr)# commit
Router(config-sr)# exit


/* Configure Segment Routing using ISIS */

Router# configure
Router(config)# router isis ring
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.0001.1921.6800.1001.00
Router(config-isis)# nsr
Router(config-isis)# distribute link-state
Router(config-isis)# nsf cisco
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# mpls traffic-eng level-1
Router(config-isis-af)# mpls traffic-eng router-id loopback0
Router(config-isis-af)# segment-routing mpls
Router(config-isis-af)# exit
!
Router(config-isis)# interface loopback0
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-af)# prefix-sid index 30101
Router(config-isis-af)# exit
```

Running Configuration

PE1:

```
router ospf dc-sr
 router-id 13.13.13.1
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing mpls
 segment-routing mpls sr-prefer
 segment-routing prefix-sid-map receive
 segment-routing prefix-sid-map advertise-local
 !
 area 1
  interface HundredGigE0/0/0/2
  !
  interface Loopback0
   prefix-sid index 1
  !
 !
!

configure
 segment-routing
  global-block 180000 200000
 !
!

configure
 router isis ring
  net 49.0001.1921.6800.1001.00
  nsr
  distribute link-state
  nsf cisco
  address-family ipv4 unicast
   metric-style wide
   mpls traffic-eng level-1
   mpls traffic-eng router-id Loopback0
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30101
  !
 !
```

P node:

```
router ospf dc-sr
 router-id 16.16.16.1
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing mpls
 segment-routing mpls sr-prefer
 segment-routing prefix-sid-map receive
 segment-routing prefix-sid-map advertise-local
 !
 area 1
```

```
  interface HundredGigE0/0/1/0
  !
  interface HundredGigE0/0/1/1
  !
  interface Loopback0
   prefix-sid index 1
  !
 !
!

configure
 segment-routing
  global-block 180000 200000
 !
!

configure
 router isis ring
  net 49.0001.1921.6800.1002.00
  nsr
  distribute link-state
  nsf cisco
  address-family ipv4 unicast
   metric-style wide
   mpls traffic-eng level-1
   mpls traffic-eng router-id Loopback0
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30102
  !
 !
```

PE2:

```
router ospf dc-sr
 router-id 20.20.20.1
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing mpls
 segment-routing mpls sr-prefer
 segment-routing prefix-sid-map receive
 segment-routing prefix-sid-map advertise-local
 !
 area 0
  interface HundredGigE0/0/0/19
  !
  interface Loopback0
   prefix-sid index 1
  !
 !
!

configure
 segment-routing
  global-block 180000 200000
 !
!

configure
 router isis ring
```

```
  net 49.0001.1921.6800.1003.00
  nsr
  distribute link-state
  nsf cisco
  address-family ipv4 unicast
   metric-style wide
   mpls traffic-eng level-1
   mpls traffic-eng router-id Loopback0
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30103
  !
```

Related Topics

You must perform these tasks as well to complete the MPLS L3VPN configuration over segment routing:

- Connect MPLS VPN Customers, on page 40
- Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 37

Associated Commands

- index
- prefix-sid
- router isis
- router ospf
- segment-routing

The applicable segment routing commands are described in the *Segment Routing Command Reference for Cisco NCS 5500 Series Routers*

Verify MPLS L3VPN Configuration over Segment Routing

- Verify the statistics in core router and ensure that the counter for IGP transport label (64003 in this example) is increasing:

  P node:

```
Router-P#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ---------
64003  Pop         SR Pfx (idx 0)     Hu0/0/0/0   193.16.1.2      572842
```

- Verify the statistics in PE1 router:

  PE1:

```
Router-P#show mpls forwarding
```

```
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ ------------    --------
64001  60003       SR Pfx (idx 0)     Hu0/0/0/2    191.22.1.2      532978
```

- Verify the statistics in PE2 router and ensure that the counter for the VPN label (24031 in this example) is increasing:

  PE2:

```
Router-PE2#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ---------
24031  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                      vrf1601                      501241
```

Also, refer Verify MPLS L3VPN Configuration, on page 51 for a detailed list of commands and sample outputs.

Implementing MPLS L3VPNs - References

MPLS L3VPN Benefits

MPLS L3VPN provides the following benefits:

- Service providers can deploy scalable VPNs and deliver value-added services.
- Connectionless service guarantees that no prior action is necessary to establish communication between hosts.
- Centralized Service: Building VPNs in Layer 3 permits delivery of targeted services to a group of users represented by a VPN.
- Scalability: Create scalable VPNs using connection-oriented and point-to-point overlays.
- Security: Security is provided at the edge of a provider network (ensuring that packets received from a customer are placed on the correct VPN) and in the backbone.
- Integrated Quality of Service (QoS) support: QoS provides the ability to address predictable performance and policy implementation and support for multiple levels of service in an MPLS VPN.
- Straightforward Migration: Service providers can deploy VPN services using a straightforward migration path.
- Migration for the end customer is simplified. There is no requirement to support MPLS on the CE router and no modifications are required for a customer intranet.

Major Components of MPLS L3VPN—Details

Virtual Routing and Forwarding Tables

Each VPN is associated with one or more VPN routing and forwarding (VRF) instances. A VRF defines the VPN membership of a customer site attached to a PE router. A VRF consists of the following components:

- An IP version 4 (IPv4) unicast routing table
- A derived FIB table
- A set of interfaces that use the forwarding table
- A set of rules and routing protocol parameters that control the information that is included in the routing table

These components are collectively called a VRF instance.

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A VRF contains all the routes available to the site from the VPNs of which it is a member.

Packet forwarding information is stored in the IP routing table and the FIB table for each VRF. A separate set of routing and FIB tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN and also prevent packets that are outside a VPN from being forwarded to a router within the VPN.

VPN Routing Information: Distribution

The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by BGP extended communities. VPN routing information is distributed as follows:

- When a VPN route that is learned from a CE router is injected into a BGP, a list of VPN route target extended community attributes is associated with it. Typically, the list of route target community extended values is set from an export list of route targets associated with the VRF from which the route was learned.
- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes that a route must have for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target extended communities A, B, and C, then any VPN route that carries any of those route target extended communities—A, B, or C—is imported into the VRF.

BGP Distribution of VPN Routing Information

A PE router can learn an IP prefix from the following sources:

- A CE router by static configuration
- An eBGP session with the CE router
- Open Shortest Path First (OSPF) and RIP as Interior Gateway Protocols (IGPs)

The IP prefix is a member of the IPv4 address family. After the PE router learns the IP prefix, the PE converts it into the VPN-IPv4 prefix by combining it with a 64-bit route distinguisher. The generated prefix is a member of the VPN-IPv4 address family. It uniquely identifies the customer address, even if the customer site is using globally nonunique (unregistered private) IP addresses. The route distinguisher used to generate the VPN-IPv4 prefix is specified by the **rd** command associated with the VRF on the PE router.

BGP distributes reachability information for VPN-IPv4 prefixes for each VPN. BGP communication takes place at two levels:

- Internal BGP (iBGP)—within the IP domain, known as an autonomous system.
- External BGP (eBGP)—between autonomous systems.

BGP propagates reachability information for VPN-IPv4 prefixes among PE routers by the BGP protocol extensions (see RFC 2283, Multiprotocol Extensions for BGP-4), which define support for address families other than IPv4. Using the extensions ensures that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

MPLS Forwarding

Based on routing information stored in the VRF IP routing table and the VRF FIB table, packets are forwarded to their destination using MPLS.

A PE router binds a label to each customer prefix learned from a CE router and includes the label in the network reachability information for the prefix that it advertises to other PE routers. When a PE router forwards a packet received from a CE router across the provider network, it labels the packet with the label learned from the destination PE router. When the destination PE router receives the labeled packet, it pops the label and uses it to direct the packet to the correct CE router. Label forwarding across the provider backbone is based on dynamic label switching. A customer data packet carries two levels of labels when traversing the backbone:

- The top label directs the packet to the correct PE router.
- The second label indicates how that PE router should forward the packet to the CE router.

Automatic Route Distinguisher Assignment

To take advantage of iBGP load balancing, every network VRF must be assigned a unique route distinguisher. VRF is require a route distinguisher for BGP to distinguish between potentially identical prefixes received from different VPNs.

With thousands of routers in a network each supporting multiple VRFs, configuration and management of route distinguishers across the network can present a problem. Cisco IOS XR software simplifies this process by assigning unique route distinguisher to VRFs using the **rd auto** command.

To assign a unique route distinguisher for each router, you must ensure that each router has a unique BGP router-id. If so, the **rd auto** command assigns a Type 1 route distinguisher to the VRF using the following format: *ip-address:number*. The IP address is specified by the BGP router-id statement and the number (which is derived as an unused index in the 0 to 65535 range) is unique across theVRFs.

Finally, route distinguisher values are checkpointed so that route distinguisher assignment to VRF is persistent across failover or process restart. If an route distinguisher is explicitely configured for a VRF, this value is not overridden by the autoroute distinguisher.

Layer 3 QinQ

The Layer 3 QinQ feature enables you to increase the number of VLAN tags in an interface and increment the number of subinterfaces up to 4094. Hence, with the dual tag, the number of VLANs can reach up to 4094*4094. You can enable this feature either on a physical interface or a bundle interface. When you cofigure this feature with the dual tag, interfaces check for IP addresses along with MAC addresses. Layer 3 QinQ is an extension of IEEE 802.1 QinQ VLAN tag stacking.

A dot1q VLAN subinterface is a virtual interface that is associated with a VLAN ID on a routed physical interface or a bundle interface. Subinterfaces divide the parent interface into two or more virtual interfaces on which you can assign unique Layer 3 parameters, such as IP addresses and dynamic routing protocols. The IP address for each subinterface must be in a different subnet from any other subinterface on the parent interface.

This feature supports:

- 802.1Q standards like 0x8100, 0x9100, 0x9200 (used as outer tag ether-type) and 0x8100 (used as inner tag ether-type).
- L3 802.1ad VLAN subinterfaces with 0x88a8 as the outer S-tag ether-type.
- Co-existence of Layer 2 and Layer 3 single tagged and double tagged VLANs.
- QinQ and dot1ad over ethernet bundle subinterfaces.

The Layer 3 QinQ feature allows you to provision quality of service (QoS), access lists (ACLs), bidirectional forwarding detection (BFD), NetFlow, routing protocols, IPv4 unicast and multicast, and IPv6 unicast and multicast.

Types of Subinterfaces

Interface type	Outer tag	Inner tag
Dot1q subinterface	0x8100	None
QinQ subinterface	0x8100	0x8100
QinQ subinterface	0x88a8	0x8100
QinQ subinterface	0x9100	0x8100
QinQ subinterface	0x9200	0x8100

Restrictions

- Only default VRF is supported.
- MPLS is not supported.

Configure Layer 3 QinQ

Configuration Example

Perform this task to configure the Layer 3 QinQ feature.

```
Router# configure
Router(config)# interface Bundle-Ether1000.3
Router(config-subif)# ipv4 address 192.0.2.1/24
Router(config-subif)# ipv6 address 2001:DB8::1/32
Router(config-subif)# ipv6 address 2001:DB8::2/32
Router(config-subif)# encapsulation dot1q 3 second-dot1q 4000
Router(config-subif)# commit
```

Running Configuration

This section shows the running configuration of Layer 3 QinQ.

```
configure
 interface Bundle-Ether1000.3
  ipv4 address 192.0.2.1/24
  ipv6 address 2001:DB8::1/32
  ipv6 address 2001:DB8::2/32
  encapsulation dot1q 3 second-dot1q 4000
   !
  !
```

Verification

Verify Layer 3 QinQ configuration.

```
Router# show interfaces Bundle-Ether1000.3
Bundle-Ether1000.3 is up, line protocol is up
  Interface state transitions: 1
  Hardware is VLAN sub-interface(s), address is 0c75.bd30.1c88
  Internet address is 192.0.2.1/24
  MTU 1522 bytes, BW 30000000 Kbit (Max: 30000000 Kbit)
     reliability 255/255, txload 0/255, rxload 6/255
  Encapsulation 802.1Q Virtual LAN, VLAN Id 3, 2nd VLAN Id 4000,
  loopback not set,
  Last link flapped 19:30:41
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:01:59
  Last clearing of "show interface" counters never
  5 minute input rate 797298000 bits/sec, 844605 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     59288018302 packets input, 6995904900380 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 2 broadcast packets, 516 multicast packets
     419 packets output, 54968 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
```

Related Topics

- Layer 3 QinQ , on page 82

Associated Commands

- show interfaces

L3VPN over GRE Tunnels

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
L3VPN over GRE Tunnels on NCS 5700 fixed port routers	Release 24.2.11	Introduced in this release on: NCS 5700 fixed port routers This feature support is now extended to NCS 5700 fixed port routers.
L3VPN over GRE Tunnels	Release 24.2.1	Introduced in this release on: NCS 5500 modular routers (NCS 5700 line cards [Mode: Native]) Generic Routing Encapsulation (GRE) allows you to configure point-to-point and multiple traffic types connections to send the various types of network traffic. GRE supports L3VPN by encapsulating various network layer protocols, allowing IPv4 and IPv6 protocols to transport within the same GRE tunnel.

Generic Routing Encapsulation (GRE) is a tunneling protocol developed by Cisco Systems. The GRE encapsulates various network layer protocols inside virtual point-to-point links over an Internet Protocol internetwork. For more information, refer to the *Configuring GRE Tunnels* chapter in the *Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers*.

MPLS L3VPN over GRE

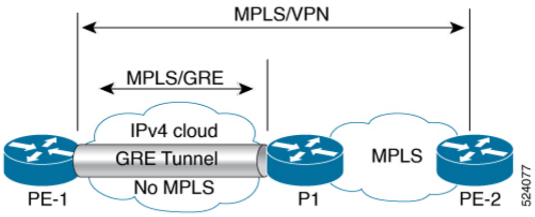
The MPLS VPN over GRE feature provides a mechanism for tunneling Multiprotocol Label Switching (MPLS) packets over a non-MPLS network. This feature uses MPLS over generic routing encapsulation (MPLSoGRE) to encapsulate packets with L3VPN label inside IP tunnels. The encapsulation of MPLS packets inside IP tunnels creates a virtual point-to-point link across non-MPLS networks.

L3VPN over GRE basically means encapsulating L3VPN traffic in the GRE header and its outer IPv4 header with tunnel destination and source IP addresses after imposing zero or more MPLS labels, and transporting it across the tunnel over to the remote tunnel end point. The incoming packet can be a pure IPv4 packet or an MPLS packet. If the incoming packet is IPv4, the packet enters the tunnel through a VRF interface, and if the incoming packet is MPLS, then the packet enters through an MPLS interface. In the IPv4 case, before encapsulating in the outer IPv4 and GRE headers, a VPN label corresponding to the VRF prefix and any IGP label corresponding to the IGP prefix of the GRE tunnel destination is imposed on the packet. In the case of MPLS, the top IGP label is swapped with any label corresponding to the GRE tunnel destination address.

PE-to-P Tunneling

As shown in the following figure, the provider-edge-to-provider (PE-to-P) tunneling configuration connects a provider edge device (PE-1) to a provider device (P1) within an MPLS segment over a non-MPLS network. In this configuration, PE-1 encapsulates the originating non-MPLS traffic within a GRE tunnel and sends it across the non-MPLS network. Upon reaching P1, the traffic exits the GRE tunnel, receives MPLS labels for forwarding, and is then sent through the MPLS network to PE-2.

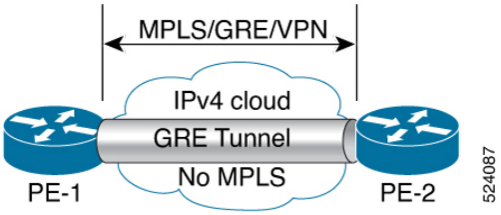
Figure 10: PE-to-P Tunneling



PE-to-PE Tunneling

As shown in the following figure, the provider-edge-to-provider-edge (PE-to-PE) tunneling configuration connects a provider edge device (PE-1) to another provider edge device (PE2) across a non-MPLS network. In this configuration, PE-1 encapsulates the originating non-MPLS traffic within a GRE tunnel and sends it across the non-MPLS network to PE-2.

Figure 11: PE-to-PE Tunneling



The GRE tunneling protocol enables:

- Service providers (that don’t run MPLS in their core network) to provide VPN services along with the security services.

GRE is used with IP to create a virtual point-to-point link to routers at remote points in a network.

L3VPN over GRE is supported only on 2-pass GRE tunnels. For more information about the Single-pass and 2-pass tunnels, see the *Configuring GRE Tunnels* chapter in the *Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers*.

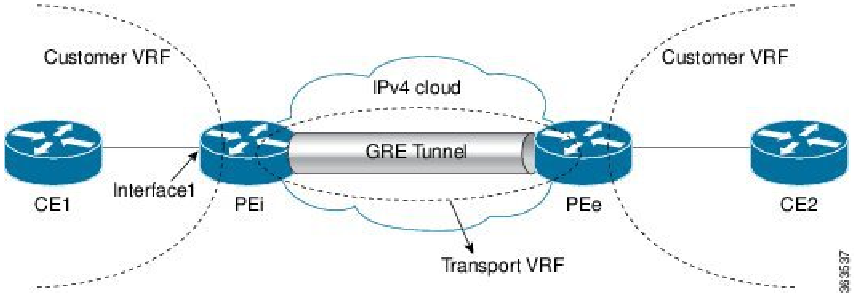
Note For a PE to PE (core) link, enable LDP (with implicit null) on the GRE interfaces for L3VPN.

Note GRE uses IP to create a virtual point-to-point link to routers at remote points in a network.

GRE Tunnel in VRF Domains

You can configure an IPv4/IPv6 GRE tunnel between two interfaces that belong to a Virtual Forwarding and Routing (VRF) instance. This contains or limits the tunnel path within this specific VRF instance. For example, packets can be sent internally within a default or non-default VRF instance separated through an intermediate VRF that contains the GRE tunnel.

Figure 12: GRE Tunnel in a VRF Instance



In the above topology, a GRE tunnel is configured in the core network, which is an IPv4 cloud. For packets entering through Interface1, the provider edge (PE) devices PEi and PEe are the tunnel head and tunnel exit respectively.

The VRF configured on Interface1 is the customer VRF. Packets entering this interface are routed using this customer VRF to the tunnel. The routing by the customer VRF is called inner IP packet routing. You can configure the tunnel to be visible to the customer VRF instance using the **vrf** *vrf-name* command. This enables only the configured VRF instance to use the tunnel, that is, forward traffic from PEi into this tunnel and also receive all incoming PEi tunnel packets.

The VRF configured on the tunnel using the **tunnel vrf** command is the transport VRF. The packet entering the tunnel is encapsulated with the tunnel source and destination addresses. The transport VRF routes this encapsulated payload between the tunnel endpoints. The routing by the transport VRF is the outer IP packet routing. If no transport VRF is configured for the tunnel, the PEi device looks up the tunnel endpoint addresses in the default VRF instance, that is, the global routing table.

GRE IPv4/IPv6 Transport Over MPLS

The Generic Routing Encapsulation (GRE) IPv4/IPv6 transport over Multiprotocol Label Switching (MPLS) feature provides a mechanism to configure GRE tunnels, where the tunnel destination IPv4/IPv6 address is reachable through an MPLS label switched path (LSP). With this feature, IPv4, IPv6, routing protocols - OSPF, ISIS, and L3VPN packets are accepted as payload packets for GRE encapsulation. IPv4/IPv6 is used as the GRE delivery protocol.

This feature overcomes the restriction of not being able to configure the tunnel destination endpoint through an MPLS LSP during tunnel configuration.

The GRE IPv4/IPv6 transport over MPLS feature facilitates creating a GRE tunnel over LSPs, through L3VPN inter-AS (autonomous system) options:

- External Border Gateway Protocol (EBGP) redistribution of labeled VPN IPv4/IPv6 routes from an AS to a neighboring AS
- Multihop EBGP redistribution of labeled VPN IPv4/IPv6 routes between source and destination ASs, with EBGP redistribution of labeled IPv4/IPv6 routes from an AS to a neighboring AS.

Supports the multipoint GRE IPv4/IPv6 transport over MPLS.

All NCS5700 linecards Supports the GRE IPv4/IPv6 transport over MPLS feature.

VPNv4 Forwarding Using GRE Tunnels

This section describes the working of VPNv4 forwarding over GRE tunnels. The following description assumes that GRE is used only as a core link between the encapsulation and decapsulation provider edge (PE) routers that are connected to one or more customer edge (CE) routers.

Ingress of Encapsulation Router

On receiving prefixes from the CE routers, Border Gateway Protocol (BGP) assigns the VPN label to the prefixes that need to be exported. These VPN prefixes are then forwarded to the Forwarding Information Base (FIB) using the Route Information Base (RIB) or the label switched database (LSD). The FIB then populates the prefix in the appropriate VRF table. The FIB also populates the label in the global label table. Using BGP, the prefixes are then relayed to the remote PE router (decapsulation router).

Egress of Encapsulation Router

The forwarding behavior on egress of the encapsulation PE router is similar to the MPLS VPN label imposition. Regardless of whether the VPN label imposition is performed on the ingress or egress side, the GRE tunnel forwards a packet that has an associated label. This labeled packet is then encapsulated with a GRE header and forwarded based on the IP header.

Ingress of Decapsulation Router

A P router or a PE router can be an ingress GRE decapsulation router that performs the following VPN traffic forwarding process:

- The P router removes the GRE encapsulation and forwards the packet based on the inner MPLS label.
- The PE router decapsulates the GRE and forwards the packet based on the VPN label, or in the case of L3VPN, the forwarding decision is made based on the label mode of the inner packet.

BGP (Border Gateway Protocol) actively distributes VPN prefix information and label mappings between PE routers. It sets the GRE tunnel interface address that connects the PE routers as the next-hop for these VPN prefixes. BGP then sends this information to the RIB (Routing Information Base), which passes the routes to the FIB (Forwarding Information Base). The FIB installs the routes into the router's hardware, ensuring packets are forwarded correctly based on the VPN routes learned.

Egress of Decapsulation Router

The egress forwarding behavior on the decapsulation PE router is similar to VPN disposition and forwarding, based on the protocol type of the inner payload.

Limitations and Restrictions for L3VPN over GRE Tunnels

The following restrictions are applicable for a GRE tunnel:

- GRE over BVI is not supported.
- MPLS packets cannot be transported within an IPv6 GRE tunnel. Therefore, the following features are not supported on an IPv6 GRE tunnel:
  - MPLS/L3VPN over GRE
  - 6PE/6VPE
  - 6PE/6VPE over GRE
- Multicast packets cannot be transported within an IPv6 GRE tunnel.
- Multicast packets cannot be transported within an IPv4 GRE tunnel that is configured in a transport VRF.
- Keep-Alive packets are not supported on GRE tunnels.
- The IPv4 addresses are mandatory for configuring GRE tunnels under the VRF, as this would ensure the traffic flows through the tunnel in an expected manner. Use either an IP unnumbered interface or a loopback interface belonging to that VRF for establishing the GRE tunnels under a VRF. Though the tunnel may come up without the aforementioned configuration, the traffic may not pass over the GRE tunnel, since the IP information on the tunnel interface is not available for forwarding the traffic correctly. Also, for the VRF information to be written in hardware database the IP information is required. Therefore, the IP unnumbered GRE tunnels may not work as expected as they may not forward traffic on the device.
- L3VPN over GRE is supported only on 2-pass GRE tunnels.

The following restrictions are applicable to L3VPN forwarding over GRE:

- GRE tunnel is supported only as a core link (PE-PE, PE-P, P-P, P-PE). A PE-CE (edge) link is not supported.
- In the L3VPN per VRF scenario, when both the L3VPN label and GRE are terminated on the same node, the labeled packet is not accounted under the tunnel interfaces and hence not displayed under the **show interfaces <tunnel interface> accounting** command.

The following features are not supported:

- GRE IPv4/IPv6 transport over MPLS-TE tunnels is not supported.
- GREoMPLS with IP Fast Reroute (IPFRR).

The following restrictions are applicable to MTU and TOS:

- Configurable MTU is not supported on Single-pass GRE interface, but supported on 2-pass GRE interface. For more information on GRE Tunnels with Supported MTU and TOS Hardware, see the *Guidelines* *and Restrictions for Configuring GRE Tunnels* section in the *Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers*.

Configuring L3VPN over GRE

You can configure the L3VPN over GRE in the following ways:

- Configure L3VPN with IGP (OSPF), MPLS LDP, and GRE
- Configure L3VPN with BGP LU, IGP (OSPF), MPLS LDP, and GRE

Configure L3VPN with IGP (OSPF), MPLS LDP, and GRE

Perform the following tasks to configure L3VPN with IGP (OSPF), MPLS LDP, and GRE:

- Configure GRE tunnel between PE routers, refer to Configuring a GRE Tunnel between Provider Edge Routers, on page 89.
- Configure IGP (OSPF) between PE routers, refer to Configuring IGP between Provider Edge Routers, on page 90.
- Configure MPLS LDP on the PE routers, refer to Configuring LDP on the Provider Edge Routers, on page 91.
- Configure L3VPN, refer to Configuring L3VPN, on page 94.

Verify the L3VPN with IGP (OSPF), MPLS LDP, and GRE configuration, refer to Verify L3VPN with IGP (OSPF), MPLS LDP, and GRE, on page 97.

Configure L3VPN with BGP LU, IGP (OSPF), MPLS LDP, and GRE

Perform the following tasks to configure L3VPN with BGP LU, IGP (OSPF), MPLS LDP, and GRE:

- Configure GRE tunnel between PE routers, refer to Configuring a GRE Tunnel between Provider Edge Routers, on page 89.
- Configure IGP (OSPF) between PE routers, refer to Configuring IGP between Provider Edge Routers, on page 90.
- Configure MPLS LDP on the PE routers, refer to Configuring LDP on the Provider Edge Routers, on page 91.
- Configure BGP LU between the PE routers, refer to Configuring BGP LU between the Provider Edge Routers, on page 93.
- Configure L3VPN, refer to Configuring L3VPN, on page 94.

Verify the L3VPN with BGP LU, IGP (OSPF), MPLS LDP, and GRE, refer to Verify L3VPN with BGP LU, IGP (OSPF), MPLS LDP, and GRE, on page 99.

Configuring a GRE Tunnel between Provider Edge Routers

Perform this task to configure a GRE tunnel between provider edge routers.

Procedure

Step 1 Configure an interface tunnel.

Example:

```
Router# configure
Router(config)# interface tunnel-ip 4000
```

Step 2 Configure IPv4 and IPv6 addresses for the tunnel interface.

Example:

```
Router(config-if)# ipv4 address 10.0.0.1 255.0.0.0
Router(config-if)# ipv6 address 100:1:1:1::1/64
```

Step 3 Configure the GRE tunnel interface with a source and destination.

Example:

```
Router(config-if)# tunnel mode gre ipv4
Router(config-if)# tunnel source TenGigE0/2/0/1
Router(config-if)# tunnel destination 145.12.5.2
Router(config-if)# commit
```

Step 4 View the running configuration to verify the configuration that you have configured.

Example:

```
interface tunnel-ip 4000
 ipv4 address 10.0.0.1 255.0.0.0
 ipv6 address 100:1:1:1::1/64
 tunnel mode gre ipv4
 tunnel source TenGigE0/2/0/1
 tunnel destination 145.12.5.2
!
```

Configuring IGP between Provider Edge Routers

Perform this task to configure IGP between provider edge routers.

Procedure

Step 1 Configure the OSPF (Open Shortest Path First).

Example:

```
Router# configure
Router(config)# router ospf 1
```

Step 2 Enable the BGP Non-Stop Routing within the OSPF

Example:

```
Router(config-ospf)# nsr
```

Step 3 Configures a router ID for the OSPF process.

Example:

```
Router(config-ospf)# router-id 10.0.0.1
```

Step 4 Enables MPLS LDP synchronization.

Example:

```
Router(config-ospf)# mpls ldp sync
```

Step 5 Sets the time to wait for a hello packet from a neighbor before declaring the neighbor down.

Example:

```
Router(config-ospf)# dead-interval 60
```

Step 6 Specifies the interval between hello packets that OSPF sends on the interface.

Example:

```
Router(config-ospf)# hello-interval 15
```

Step 7 Enters area configuration mode and configures an area for the OSPF process.

Example:

```
Router(config-ospf)# area 0
```

Step 8 Configure an interface tunnel.

Example:

```
Router(config-ospf)# interface tunnel-ip 4000
```

Step 9 View the running configuration to verify the configuration that you have configured.

Example:

```
Router# show run router ospf 1
router ospf 1
 nsr
 router-id 10.0.0.1 <=== Loopback0
 mpls ldp sync
 mtu-ignore enable
 dead-interval 60
 hello-interval 15
 area 0
  interface TenGigE0/2/0/1
 !
```

Configuring LDP on the Provider Edge Routers

Perform this task to configure LDP on the provider edge routers.

Procedure

Step 1 Enable MPLS LDP configuration mode.

Example:

```
Router# configure
Router(config)# mpls ldp
```

Step 2 Configure a router ID for the OSPF process.

Example:

```
Router(config-ldp)# router-id 10.0.0.1
```

Note
We recommend using a stable IP address as the router ID.

Step 3 Define the period of time a discovered LDP neighbor is remembered without receipt of an LDP Hello message from the neighbor.

Example:

```
Router(config-ldp)# discovery hello holdtime 40
```

Step 4 Define the period of time between the sending of consecutive Hello messages.

Example:

```
Router(config-ldp)# discovery hello holdtime 20
```

Step 5 Enable the BGP Non-Stop Routing for for MPLS LDP.

Example:

```
Router(config-ospf)# nsr
```

Step 6 Enables graceful restart on the router.

Example:

```
Router(config-ldp)# graceful-restart
```

Step 7 Define the time for which the neighbor should wait for a reconnection if the LDP session is lost.

Example:

```
Router(config-ldp)# graceful-restart recoonect-timeout 180
```

Step 8 Defines the time that the neighbor should retain the MPLS forwarding state during a recovery.

Example:

```
Router(config-ldp)# graceful-restart forwarding-state-holdtime 300
```

Step 9 Configures the hold time for an interface.

Example:

```
Router(config-ldp)# holdtime 90
```

Step 10 Define a neighboring router and configure tunnel interface.

Example:

```
Router(config-ldp)# neighbor 10.1.1.0
Router(config-ldp)# interface tunnel-ip 4000
```

Step 11 View the running configuration to verify the configuration that you have configured.

Example:

```
mpls ldp
 router-id 10.0.0.1
 discovery hello holdtime 40
 discovery hello holdtime 20
 nsr
```

```
  graceful-restart
  graceful-restart reconnect-timeout 180
  graceful-restart forwarding-state-holdtime 300
  holdtime 90
  neighbor 10.1.1.0
  interface tunnel-ip 4000
 !
```

Configuring BGP LU between the Provider Edge Routers

Perform this task to configure BGP LU between the provider edge routers.

Procedure

Step 1 Configure the BGP routing and assign a router ID.

Example:

```
Router# configure
Router(config)# router bgp 1
Router(config-bgp)# bgp router-id 10.0.0.1
```

Step 2 Assign an address family to the BGP routing and enable the redistribution of connected routes into BGP.

Example:

```
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# redistribute connected
```

Step 3 Configure to exchange labels for each IPv4 or IPv6 Prefix.

Example:

```
Router(config-bgp-af)# allocate-label all unlabeled-path
```

Step 4 Configure a BGP neighbor, assign a remote autonomous system number to it, and configure the BGP source.

Example:

```
Router(config-bgp)# neighbor 192.168.0.1
Router(config-bgp)#remote-as 1
Router(config-bgp)# update-source 10.0.0.1
```

Step 5 Enable the IPv4 or IPv6 labeled unicast for the BGP session and apply the BGP policy to pass all the incoming and outgoing IPv4 or IPv6 labeled-unicast address family routes from this neighbor.

Example:

```
Router(config-bgp)# address-family ipv6 labeled-unicast
Router(config-bgp-af)# route-policy pass-all in
Router(config-bgp-af)# route-policy pass-all out
```

Step 6 View the running configuration to verify the configuration that you have configured.

Example:

```
router bgp 1
 bgp router-id 10.0.0.1
 address-family ipv4 unicast
  redistribute connected
  allocate-label all unlabeled-path
  !
neighbor 192.168.0.1
remote-as 1
update-source Loopback 0
address-family ipv6 labeled-unicast
 route-policy pass-all in
 route-policy pass-all out
 !
!
```

Configuring L3VPN

Perform this task to configure L3VPN.

Procedure

Step 1 Configure a VRF instance.

Example:

```
Router# configure
Router(config)# vrf vpn1
```

Step 2 Configure the IPv4 or IPv6 address family and set import and export routes.

Example:

```
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target 2:1
Router(config-vrf-af)# export route-target 1:1
```

For import routes, only prefixes that are associated with the specified import route target extended communities are imported into the VRF.

Export route target communities are associated with prefixes when they are advertised to remote PEs. The remote PEs import them into VRFs which have import RTs that match these exported route target communities.

Step 3 Configure an interface and assign a VRF instance and an IPv4 or IPv6 address to that interface.

Example:

```
Router(config)# interface TenGigE0/2/0/0.1
Router(config)# interface TenGigE0/2/0/0.1
Router(config-if)# vrf vpn1
Router(config-if)# ipv4 address 172.16.0.1 255.240.0.0
```

Step 4 Assign the native VLAN ID of a physical interface trunking 802.1Q VLAN traffic.

Example:

```
Router(config-if)# dot1q native vlan 1
```

Step 5 Configure the BGP routing.

Example:

```
Router# configure
Router(config)# router bgp 1
```

Step 6 Enable the BGP Non-Stop Routing.

Example:

```
Router(config-ospf)# nsr
```

Step 7 Configure a router ID.

Example:

```
Router(config-bgp)# bgp router-id 10.0.0.1
```

Step 8 Configure VPNv4 address family.

Example:

```
Router(config-bgp)# address-family vpnv4 unicast
```

Step 9 Configure the neighbor IP address as a BGP peer, assign a remote autonomous system number to it, and configure te BGP souce.

Example:

```
Router(config-bgp)# neighbor 192.168.0.1
Router(config-bgp-nbr)#remote-as 1
Router(config-bgp-nbr)#update-source Loopback0
```

Step 10 Enable the VPNV4 or VPNV6 labeled unicast for the BGP session and apply the BGP policy to pass all the incoming and outgoing VPNV4 or VPNV6 labeled-unicast address family routes from this neighbor.

Example:

```
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
```

Step 11 Configure a VRF instance.

Example:

```
Router(config)# vrf vpn1
```

Step 12 Configure the route distinguisher

Example:

```
Router(config-vrf)#rd 1:1
```

Step 13 Configure IPv4 or IPv6 address family to the VRF instance.

Example:

```
Router(config-vrf)# address-family ipv4 unicast
```

Step 14 Configure the VRF to redistribute routes for directly connected networks into the VRF's routing table.

Example:

```
Router(config-vrf-af)# redistribute connected
```

Step 15 Configure the VRF to redistribute static routes into the VRF's routing table.

Example:

```
Router(config-vrf-af)#  redistribute static
```

Step 16 Configure the neighbor IP address as a BGP peer and assign a remote autonomous system number to it.

Example:

```
Router(config-bgp)# neighbor 172.16.0.2
Router(config-bgp-nbr)#remote-as 7501
```

Step 17 Configure the CE neighbor to accept and attempt BGP connections to external peers residing on networks that are not directly connected.

Example:

```
Router(config-bgp-nbr)# ebgp-multihop 10
```

Step 18 Enable the IPv4 or IPv6 labeled unicast for the BGP session and apply the BGP policy to pass all the incoming and outgoing IPv4 or IPv6 labeled-unicast address family routes from this neighbor.

Example:

```
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy BGP_pass_all in
Router(config-bgp-nbr-af)#route-policy BGP_pass_all out
```

Step 19 View the running configuration to verify the configuration that you have configured.

Example:

```
vrf vpn1
 address-family { ipv4 | ipv6 } unicast
 import route-target 2:1
 export route-target 1:1
 !
interface TenGigE0/2/0/0.1
 vrf vpn1
 ipv4 address 172.16.0.1 255.240.0.0
 dot1q native vlan 1
 !
router bgp 1
 nsr
 bgp router-id 10.0.0.1
 address-family vpnv4 unicast
 neighbor 192.168.0.1
 remote-as 1
 update-source Loopback0
 address-family vpnv4 unicast
  route-policy pass-all in
  route-policy pass-all out
 !
!
vrf vpn1
 rd 1:1
 address-family ipv4 unicast
  redistribute connected
  redistribute static
 !
neighbor 172.16.0.2
 remote-as 7501
 ebgp-multihop 10
 address-family ipv4 unicast
  route-policy BGP_pass_all in
  route-policy BGP_pass_all out
  !
 !
!
```

Verify L3VPN with IGP (OSPF), MPLS LDP, and GRE

The following example shows how to verify L3VPN with IGP (OSPF), MPLS LDP, and GRE:

Verify the GRE Tunnel between PE1 and PE2:

The following example is to verify the GRE tunnel between PE1 and PE2:

```
PE1# sh run int tunnel-ip 1
interface tunnel-ip1
 ipv4 address 172.16.0.1 255.240.0.0
 ipv6 address 100:1:1:1::1/64
 tunnel mode gre ipv4
 tunnel source TenGigE0/2/0/1
 tunnel destination 145.12.5.2
!
PE2# sh run int tunnel-ip 1
interface tunnel-ip1
 ipv4 address 172.16.0.2 255.240.0.0
 ipv6 address 100:1:1:1::2/64
 tunnel mode gre ipv4
 tunnel source TenGigE0/1/0/2
 tunnel destination 192.168.0.1
```

Verify the IGP between PE1 and PE2:

The following example is to verify the IGP for PE1. Follow the same steps to verify the IGP for PE2.

```
PE1# sh run router ospf 1
router ospf 1
 nsr
 router-id 10.0.0.1 <=== Loopback0
 mpls ldp sync
 mtu-ignore enable
 dead-interval 60
 hello-interval 15
 area 0
  interface TenGigE0/2/0/1
  !
PE1# sh run router ospf 0
router ospf 0
 nsr
 router-id 10.0.0.1
 mpls ldp sync
 dead-interval 60
 hello-interval 15
 area 0
  interface Loopback0
  !
  interface tunnel-ip1
  !

* Check for OSPF neighbors

PE1#sh ospf neighbor

Neighbors for OSPF 0

Neighbor ID     Pri   State           Dead Time   Address         Interface
4.4.4.4         1     FULL/  -        00:00:47    172.16.0.2       tunnel-ip1      <==
Neighbor PE2
```

```
    Neighbor is up for 00:13:40
Neighbors for OSPF 1

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2         1     FULL/DR         00:00:50    192.168.0.1      TenGigE0/2/0/1  <==
Neighbor P1
    Neighbor is up for 00:13:43
```

Verify the LDP on PE1 and PE2:

The following example is to verify the LDP on PE1. Follow the same steps to verify the LDP on PE2.

```
PE1# sh run mpls ldp
mpls ldp
 router-id 10.0.0.1 <=== Loopback0
 discovery hello holdtime 45
 discovery hello interval 15
 nsr
 graceful-restart
 graceful-restart reconnect-timeout 180
 graceful-restart forwarding-state-holdtime 300
 holdtime 90
 log
  neighbor
 !
 interface tunnel-ip1
 !

*Check for mpls forwarding

PE1#sh mpls forwarding prefix 10.0.0.2/8
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
---- ------- -------------- -------- ----------- ----------
16003  Pop         10.0.0.2/8         ti1          172.16.0.2       0
```

Verify the L3VPN on PE1 and PE2

The following example is to verify the L3VPN on PE1. Follow the same steps to verify the L3VPN on PE2.

```
PE1# sh run vrf vpn1
vrf vpn1
 address-family ipv4 unicast
  import route-target
   2:1
  !
  export route-target
   1:1
  !
PE1# sh run int tenGigE 0/2/0/0.1
interface TenGigE0/2/0/0.1
 vrf vpn1
 ipv4 address 172.16.0.1 255.255.255.0
 encapsulation dot1q 1
!

PE1# sh run router bgp
router bgp 1
 nsr
 bgp router-id 10.0.0.1 <===Loopback0
 address-family vpnv4 unicast
 !
 neighbor 192.168.0.1      <===iBGP session with PE2
  remote-as 1
  update-source Loopback0
  address-family vpnv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 vrf vpn1
  rd 1:1
  address-family ipv4 unicast
   redistribute connected
   redistribute static
  !
  neighbor 172.16.0.2   <=== VRF neighbor
   remote-as 7501
   ebgp-multihop 10
   address-family ipv4 unicast
    route-policy BGP_pass_all in
    route-policy BGP_pass_all out
   !

* Check vrf ping to the 172.16.0.2

PE1# ping vrf vpn1 172.16.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms

* Send traffic to vrf routes adverstised and verify that mpls counters increase in tunnel
interface accounting

PE1# sh int tunnel-ip1 accounting
tunnel-ip1
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  IPV4_MULTICAST              3              276            3              276
   MPLS                   697747         48842290            0                0
```

Verify L3VPN with BGP LU, IGP (OSPF), MPLS LDP, and GRE

The following example shows how to verify the L3VPN with BGP LU, IGP (OSPF), MPLS LDP, and GRE:

Verify the GRE Tunnel between PE1 and PE2:

The following example is to verify the GRE tunnel between PE1 and PE2:

```
PE1# sh run int tunnel-ip 1
interface tunnel-ip1
 ipv4 address 172.16.0.1 255.240.0.0
 ipv6 address 100:1:1:1::1/64
 tunnel mode gre ipv4
 tunnel source TenGigE0/2/0/1
 tunnel destination 145.12.5.2
!
PE2# sh run int tunnel-ip 1
interface tunnel-ip1
 ipv4 address 172.16.0.2 255.240.0.0
 ipv6 address 100:1:1:1::2/64
 tunnel mode gre ipv4
 tunnel source TenGigE0/1/0/2
 tunnel destination 192.168.0.1
```

Verify the IGP between PE1 and PE2:

The following example is to verify the IGP for PE1. Follow the same steps to verify the IGP for PE2.

```
PE1# sh run router ospf 1
router ospf 1
 nsr
 router-id 10.0.0.1 <=== Loopback0
 mpls ldp sync
 mtu-ignore enable
 dead-interval 60
 hello-interval 15
 area 0
  interface TenGigE0/2/0/1
  !
PE1# sh run router ospf 0
router ospf 0
 nsr
 router-id 10.0.0.1
 mpls ldp sync
 dead-interval 60
 hello-interval 15
 area 0
  interface Loopback0
  !
  interface tunnel-ip1
  !

* Check for OSPF neighbors

PE1#sh ospf neighbor

Neighbors for OSPF 0

Neighbor ID     Pri   State           Dead Time   Address         Interface
4.4.4.4         1     FULL/  -        00:00:47    172.16.0.2       tunnel-ip1      <==
Neighbor PE2
    Neighbor is up for 00:13:40
Neighbors for OSPF 1

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2         1     FULL/DR         00:00:50    192.168.0.1      TenGigE0/2/0/1  <==
Neighbor P1
    Neighbor is up for 00:13:43
```

Verify the LDP on PE1 and PE2:

The following example is to verify the LDP on PE1. Follow the same steps to verify the LDP on PE2.

```
PE1# sh run mpls ldp
mpls ldp
 router-id 10.0.0.1 <=== Loopback0
 discovery hello holdtime 45
 discovery hello interval 15
 nsr
 graceful-restart
 graceful-restart reconnect-timeout 180
 graceful-restart forwarding-state-holdtime 300
 holdtime 90
 log
  neighbor
 !
 interface tunnel-ip1
 !

*Check for mpls forwarding

PE1#sh mpls forwarding prefix 10.0.0.2/8
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
---- ------- -------------- -------- ----------- ----------
16003  Pop         10.0.0.2/8         ti1          172.16.0.2       0
```

Configure BGP LU between PE1 and PE2:

The following example is to verify the BGP LU for PE1. Follow the same steps to verify the BGP LU for PE2.

```
router bgp 1
 bgp router-id 2001:DB8::1
 address-family ipv6 unicast
  redistribute connected
  allocate-label all unlabeled-path
!
 neighbor 2001:DB8::2
 remote-as 1
 update-source Loopback0
!
 address-family ipv6 labeled-unicast
 route-policy pass-all in
 route-policy pass-all out
!
```

Verify the L3VPN on PE1 and PE2

The following example is to verify the L3VPN on PE1. Follow the same steps to verify the L3VPN on PE2.

```
PE1# sh run vrf vpn1
vrf vpn1
 address-family ipv4 unicast
  import route-target
   2:1
  !
  export route-target
   1:1
  !
PE1# sh run int tenGigE 0/2/0/0.1
interface TenGigE0/2/0/0.1
 vrf vpn1
 ipv4 address 172.16.0.1 255.255.255.0
 encapsulation dot1q 1
!

PE1# sh run router bgp
router bgp 1
 nsr
 bgp router-id 10.0.0.1 <===Loopback0
 address-family vpnv4 unicast
 !
 neighbor 192.168.0.1      <===iBGP session with PE2
```

```
  remote-as 1
  update-source Loopback0
  address-family vpnv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 vrf vpn1
  rd 1:1
  address-family ipv4 unicast
   redistribute connected
   redistribute static
  !
  neighbor 172.16.0.2   <=== VRF neighbor
   remote-as 7501
   ebgp-multihop 10
   address-family ipv4 unicast
    route-policy BGP_pass_all in
    route-policy BGP_pass_all out
   !

* Check vrf ping to the 172.16.0.2

PE1# ping vrf vpn1 172.16.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms

* Send traffic to vrf routes adverstised and verify that mpls counters increase in tunnel
interface accounting

PE1# sh int tunnel-ip1 accounting
tunnel-ip1
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  IPV4_MULTICAST              3              276            3              276
   MPLS                   697747         48842290            0                0
```