L3VPN Configuration Guide for Cisco 8000 Series Routers, Cisco IOS XR Releases

Cisco 8000 Series Routers
Release: L3VPN | Updated June 17, 2026

Topics included

1 YANG Data Models for L3VPN Features

Topics:

Lists the supported YANG data models and their corresponding L3VPN features on this platform.

This chapter provides information about the YANG data models for L3VPN features.

Using YANG Data Models

Provides a summary of the locations, tools, and resources for finding and exploring Cisco IOS XR YANG data models.

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the Github repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPaths. To view a comprehensive list of the data models supported in a release, navigate to the *Available-Content.md* file in the repository.

You can also view the data model definitions using the YANG Data Models Navigator tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.

2 MPLS L3VPN Fundamentals and Core Deployment

Topics:

Introduces MPLS L3VPN fundamentals, core network deployment, and advanced VPN service options, outlining MPLS Layer 3 VPN concepts, topology operations, configuration procedures, verification steps, VRF-lite integration, and implementation using RSVP-TE and Segment Routing.

MPLS Layer 3 VPN services

Explains MPLS Layer 3 VPN core concepts, including topology operations, benefits, major components, VRF tables, VPN routing information distribution, BGP integration, forwarding processes, route distinguisher assignment, as well as prerequisites and restrictions for implementing MPLS L3VPN.

An MPLS Layer 3 VPN service is a private network service that

- interconnects customer sites through an MPLS provider core network
- uses PE routers to attach VPN and core labels to traffic, and
- uses the peer model to exchange Layer 3 routing information.

Feature history

The feature history table lists release support for this feature.

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
MPLS Layer 3 VPN	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: - 8011-32Y8L2H2FH - 8011-12G12X4Y-A/D
MPLS Layer 3 VPN	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on Cisco 8011-4G24Y4H-I routers.
MPLS Layer 3 VPN	Release 24.4.1	Introduced in this release on: Fixed Systems (8700) (select variants only*) *The MPLS Layer 3 VPN functionality is now extended to the Cisco 8712-MOD-M routers.
MPLS Layer 3 VPN	Release 24.3.1	Introduced in this release on: Fixed Systems (8200, 8700); Modular Systems (8800 [LC ASIC: P100]) (select variants only*) *The MPLS Layer 3 VPN functionality is now extended to: - 8212-48FH-M - 8711-32FH-M - 88-LC1-52Y8H-EM - 88-LC1-12TH24FH-E

Feature Name	Release Information	Feature Description
MPLS Layer 3 VPN	Release 24.2.11	Introduced in this release on: Modular Systems (8800 [LC ASIC: P100]) (select variants only*) MPLS VPNs offer a streamlined and scalable approach to creating private network services over public infrastructures by simplifying the management and expansion processes. Unlike conventional VPNs that require complex configurations of tunnels or PVCs for every site, MPLS VPNs utilize the peer model, allowing service providers to handle routing and data relay between customer sites. This means that adding a new site requires updates only to the service provider's edge router, greatly enhancing efficiency and reducing complexity. *This functionality is now extended to routers with the 88-LC1-36EH line cards.

Understanding MPLS VPNs

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:

Conventional VPNs require configuring a full mesh of tunnels or permanent virtual circuits (PVCs) for all sites in a VPN. This approach is difficult to maintain or expand, as adding a new site requires updating every edge device in the network.

MPLS-based VPNs:

MPLS-based VPNs operate at Layer 3 and are built on the peer model. This model allows the service provider and customer to exchange Layer 3 routing information, with the provider relaying data between customer sites without customer involvement. MPLS VPNs simplify management and expansion: when a new site is added, only the relevant service provider's edge router needs updating, making them easier to use and scale than traditional VPNs.

How MPLS Layer 3 VPNs work

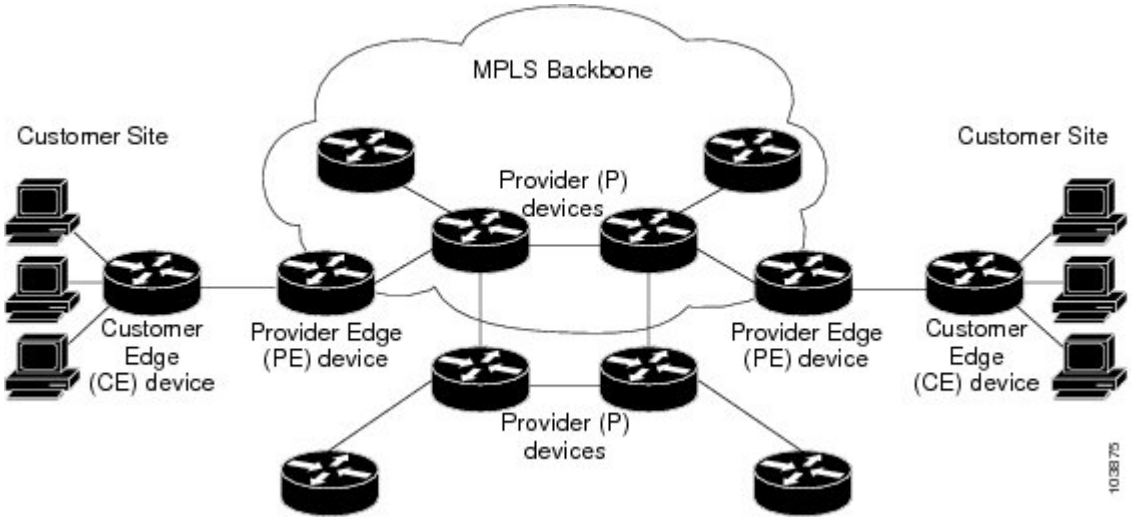
Describes the basic structure of MPLS Layer 3 VPN topologies, including the roles of provider, customer, provider edge (PE), provider (P), and customer edge (CE) routers, and how they connect customer networks through an MPLS provider core.

The following figure depicts a basic MPLS VPN topology.

The key components involved in the process are:

- Provider (P) router: Operates within the core of the provider network, switches MPLS labels, and does not attach VPN labels to packets.
- Provider edge (PE) router: Connects directly to the customer edge (CE) router, attaches VPN labels to incoming packets, and manages MPLS core labels for packets traveling within the provider network.
- Customer edge (CE) router: Sits at the edge of the customer's network, connects to the PE router, and interfaces with the provider’s network.
- Customer (C) router: Functions within the customer’s internal network, routing data to and from the CE router.

MPLS Layer 3 VPNs use specialized routers and label-based packet forwarding to securely connect customer sites over a provider's MPLS-enabled backbone. Each type of router has a specific role that ensures data is directed to the correct private network.



These stages describe how MPLS Layer 3 VPNs work.

1. The customer (C) router sends data toward an external network, forwarding packets to the customer edge (CE) router.
2. The CE router passes outgoing packets to the provider edge (PE) router.
3. The PE router attaches the appropriate VPN labels (based on the incoming interface or sub-interface) and MPLS core labels, and sends the packets into the provider (P) network.
4. Provider (P) routers within the core forward packets based on MPLS labels, without modifying VPN labels.
5. When the packets reach the PE router at the destination site, the router removes the MPLS core labels and forwards the packets to the appropriate CE router (and onward to the correct customer network), completing the VPN connection.

This topology securely connects multiple customer sites by routing traffic across the provider's MPLS backbone, using label forwarding to ensure each customer's data remains private and correctly delivered to each site.

Benefits of MPLS L3VPN

Lists the key benefits of MPLS L3VPN services, including scalable VPN deployment, centralized service delivery, security, QoS integration, and simplified migration.

MPLS L3VPN services offer several advantages for service providers and customers:

- Scalable VPN deployment: Service providers can build scalable VPNs using both connection-oriented and point-to-point overlays, delivering value-added services without prior host coordination.
- Centralized service delivery: Layer 3 VPNs enable targeted services for specific user groups, simplifying management through centralized control.
- Security: Security is enforced at the provider network edge and across the backbone, ensuring customer packets are placed on the correct VPN.
- Integrated Quality of Service (QoS): MPLS L3VPN supports multiple service levels, performance guarantees, and policy implementation directly within the VPN.
- Simplified migration: Providers can deploy VPN services using a straightforward migration path, and customers are not required to support MPLS on their edge routers or modify their existing intranets.

These benefits help service providers deliver flexible, secure, and efficient VPN solutions tailored to a wide range of customer requirements.

Major components of MPLS L3VPN

Explains the three major MPLS L3VPN components and how each contributes to VPN functionality and connectivity across the provider network.

An MPLS L3VPN component is a routing or forwarding element that

- defines VPN membership with route target communities
- propagates VRF reachability through MP-BGP peering, and
- transports VPN traffic across the provider network with MPLS forwarding.

Component details

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.

Virtual routing and forwarding tables

Explains how VRF instances define VPN membership, maintain separate routing and forwarding tables, and prevent traffic leakage between VPNs.

A virtual routing and forwarding table is a VPN-specific routing and forwarding context that

- defines VPN membership for a customer site attached to a PE router
- maintains separate routing and FIB tables for each VRF, and
- controls which interfaces and routing protocol parameters belong to the VRF.

Feature history

The feature history table lists release support for this feature.

Table 2: Feature History Table

Feature Name	Release	Description
Virtual Routing and Forwarding Tables	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D

Feature Name	Release	Description
Virtual Routing and Forwarding Tables	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Virtual Routing and Forwarding Tables	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The router now supports 2000 VRF instances which enhances network segmentation capabilities, allowing for more granular and efficient management of virtual routing and forwarding instances. This improvement supports larger and more complex network architectures, enabling service providers to offer more tailored services to their customers. The expanded VRF capacity ensures that businesses can grow their networks without compromising on performance or reliability. By accommodating up to 2000 VRFs, users benefit from greater flexibility and scalability, catering to diverse and demanding network environments. *Previously this feature was supported on Q200 and Q100. It is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

VRF details

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.

How VPN routing information works

Describes how BGP extended communities control export and import of VPN routing information between VRFs.

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:

The key components involved in the process are:

- VPN route target communities: Specify which VPN routes can be exported from and imported into a VRF.
- BGP extended communities: Carry the route target information in BGP updates, enabling precise route sharing.
- VRF: Maintains its own routing table, controlling route import/export through configured route targets.

VPN routing information is distributed by controlling export and import through VPN route target communities, which are implemented by BGP extended communities. This allows selective sharing of routes among different VRFs to maintain network segmentation.

These stages describe how VPN routing information works.

1. When a VPN route is learned from a CE router and injected into BGP, it is associated with a list of VPN route target extended community attributes. This list is typically set from the export list of route targets configured for the VRF from which the route was learned.
2. Each VRF has an import list of route target extended communities. For a VPN route to be imported into a VRF, it must carry at least one of the route target extended communities specified in that VRF's import list. For example, if the import list includes route target communities A, B, and C, any VPN route carrying A, B, or C will be imported into the VRF.

VPN routes are imported into a VRF only when their route target extended communities match the import list for that VRF, enabling controlled sharing of routing information between VPN segments.

How BGP distributes VPN routing information

Describes how PE routers learn IP prefixes, convert them to VPN-IPv4 prefixes, and distribute reachability information with BGP extensions.

A PE router learns IP prefixes from various sources, including static configuration on a CE router, eBGP sessions with a CE router, and interior gateway protocols like OSPF. The IP prefix is then converted into the VPN-IPv4 prefix using a 64-bit route distinguisher configured for the VRF. This process ensures a unique identifier for each customer site, even if private, unregistered IP addresses are used.

The key components involved in the process are:

- PE router: Learns customer IP prefixes, converts them to VPN-IPv4 prefixes, and manages VRFs.
- CE router: Supplies IP prefixes through static configuration or routing protocols.
- BGP protocol with extensions: Distributes VPN-IPv4 reachability information among PE routers, ensuring only VPN members receive pertinent routes.

BGP enables routing between customer sites in a VPN by using Multiprotocol BGP extensions to propagate VPN-IPv4 reachability among participating routers.

These stages how BGP distributes VPN routing information:

1. Prefix learning: The PE router learns IP prefixes from the CE router via static configuration, an eBGP session, or an IGP such as OSPF.
2. Prefix conversion: The PE router combines each IP prefix with a 64-bit route distinguisher to generate a VPN-IPv4 prefix, uniquely identifying the customer address.
3. Route distribution: BGP, using Multiprotocol Extensions (RFC 2283), distributes reachability information for VPN-IPv4 prefixes among PE routers. Routes for a given VPN are learned only by other members of the same VPN.
4. BGP communication: Reachability is propagated at two levels:
   - Internal BGP (iBGP): Distributed within the provider's autonomous system.
   - External BGP (eBGP): Shared between autonomous systems as needed.
5. VPN member communication: Only routers that are members of the relevant VPN learn these VPN-IPv4 routes, enabling inter-site connectivity within the VPN.

BGP distributes VPN-IPv4 reachability only to participating VPN members, enabling secure and scalable inter-site routing for customers while preventing route leaks and overlap between VPNs.

How MPLS forwarding works

Describes how PE routers bind labels to customer prefixes and forward VPN traffic using a top label and a VPN label across the provider backbone.

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

The key components involved in the process are:

- PE router: Connects to customer routers (CE routers), learns customer prefixes, assigns labels, and forwards traffic through the provider backbone.
- CE router: Connects to a PE router and exchanges routing information for customer sites.
- MPLS labels (top and VPN label): The top label determines the packet’s path across the provider network to the destination PE router; the VPN label identifies the final customer destination behind the PE router.

MPLS forwarding enables efficient transport of customer data across a service provider’s backbone by assigning and stacking labels on packets. This process directs traffic through the backbone and ensures delivery to the correct destination.

These stages describe how MPLS forwarding works:

1. The PE router learns customer prefixes from the connected CE router and assigns (binds) an MPLS label to each prefix.
2. The PE router advertises the customer prefix along with its label to other PE routers in the network.
3. When forwarding a packet from a CE router to a remote site, the ingress PE router stacks two labels on the packet: the top label (for backbone transport) and the VPN label (for ultimate destination).
4. As the packet traverses the provider backbone, provider routers (P routers) forward the packet based on the top label.
5. The destination PE router receives the packet, removes the top label, and examines the VPN label to determine which CE router should receive the packet.

This process ensures that customer data is routed efficiently and securely across the service provider backbone to its intended destination using dynamic MPLS label switching.

How automatic route distinguisher assignment works

Describes how the rd auto command assigns persistent Type 1 route distinguishers to VRFs from the BGP router ID and an unused index.

To enable efficient iBGP load balancing, each network VRF must have a unique route distinguisher. This prevents conflicts between identical prefixes received from multiple VPNs.

The key components involved in the process are:

- VRF: Requires a unique route distinguisher to distinguish between identical prefixes from different VPNs.
- BGP router ID: Provides the IP address used to construct the route distinguisher.
- rd auto command: Assigns a persistent Type 1 route distinguisher to each VRF based on the router ID and an unused index.

Automatic route distinguisher assignment simplifies network configuration by ensuring that every network VRF receives a unique identifier for BGP load balancing.

These stages describe how automatic route distinguisher assignment works:

1. Configuration and management: In large-scale networks with thousands of routers and multiple VRFs, manual management of route distinguishers is complex. Cisco IOS XR simplifies this by using the **rd auto** command.
2. Assignment using rd auto: Each router must have a unique BGP router ID. The **rd auto** command assigns a Type 1 route distinguisher to each VRF in the format ip-address:number, where the IP address is the router ID and the number is an unused index in the range 0 to 65535.
3. Persistence and checkpointing: Assigned route distinguisher values are checkpointed so that they remain persistent across failover or process restart. If a route distinguisher is explicitly configured for a VRF, it is not overridden by the automatic assignment.

Automatic route distinguisher assignment remains persistent across failover or process restart unless a route distinguisher is explicitly configured for a VRF.

MPLS L3VPN prerequisites

Outlines the user-group and feature prerequisites required before configuring MPLS Layer 3 VPN.

To configure MPLS Layer 3 VPN, ensure the following requirements are met:

- You must belong to a user group associated with a task group that includes the necessary task IDs for these commands:
  - BGP
  - IGP
  - MPLS
  - MPLS Layer 3 VPN
- If user group assignment prevents you from using a command, contact your AAA administrator for assistance.
- The routers must support MPLS forwarding and Forwarding Information Base (FIB).

MPLS L3VPN restrictions

Outlines key requirements and restrictions for MPLS L3VPN deployments, including MTU behavior, Inter-AS LSPs, SR-TE, label assignment, per-VRF label mode, and L3 interface scaling.

When designing or configuring MPLS L3VPN deployments, adhere to the following requirements and restrictions:

- Do not configure MPLS packet fragmentation for packets that exceed the egress MTU. Fragmentation is unsupported for both IP-to-MPLS imposition and standard MPLS operations.

- Set the MTU to the maximum value (9216) on all interfaces within the MPLS core to avoid fragmentation issues.
- Accept that L3VPN prefix lookup yields a single path. When multiple paths are available at the IGP or BGP level, recognize that path selection occurs using a flow hash computed in the data plane.
- Do not rely on per VRF aggregate statistics, as they are not supported.
- For MPLS VPN Inter-AS deployments with ASBRs exchanging IPv4 routes and MPLS labels:
  - Configure a label switched path (LSP) between non-adjacent routers when using eBGP multihop.
  - Do not use Layer 3 VPN over SR-TE; this is not supported.
- For label assignments in MPLS VPNs
  - Allocate a local label for every VRF.
  - Assign one VPN label per VRF.
  - Use per VRF label mode across the entire VRF deployment.
- Do not exceed these maximum numbers of L3 interfaces per component when operating in P100 mode:

Component	Maximum number of L3 Interfaces Per Component
NPU (P100)	8000
Line card (88-LC1-36EH)	16000
Router	30000

Configure the core network

Outlines procedures for deploying the MPLS core network, describing operations of MPLS LDP core topology, OSPF-based core routing, and multiprotocol BGP PE router topology to ensure robust L3VPN connectivity.

Establish a robust MPLS L3VPN core network to enable secure and scalable VPN services.

In a network topology using MPLS L3VPN services over an MPLS LDP core, this task ensures the core is properly configured to handle routing and VPN requirements.

Configuring the core network involves these main tasks:

- Assess the Needs of MPLS VPN Customers
- Configure Routing Protocols in the Core
- Configure MPLS in the Core
- Determine if FIB is Enabled in the Core
- Configure Multiprotocol BGP on the PE Routers and Route Reflectors

- Identify the core topology and customer VPN needs.
- Select the routing protocol for the core (OSPF or IS-IS).
- Specify BGP load-sharing and redundant path requirements.

Follow these steps to configure the MPLS L3VPN core network:

1. Assess the needs of MPLS VPN customers.
   - Determine the number of customers, VPNs per customer, and VRF instances for each VPN.

     Confirm routing protocol requirements and whether BGP load sharing or redundant paths are needed.

2. Configure core routing protocols.

   Use OSPF or IS-IS for core routing.

```
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-vrf-ar-if)#commit
```

   use the **show ospf neighbor** command to verify neighbor status and ensure the state is 'FULL'.

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

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.0.2.11      1     FULL/DR  00:00:34    192.0.2.12      HundredGigE0/0/0/2

    Neighbor is up for 1d18h

Total neighbor count: 1
```

3. Configure MPLS in the core.

```
Router-PE1#configure
Router-PE1(config)#mpls ldp
Router-PE1(config-ldp)#router-id 192.0.2.10
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)#commit
```

   Repeat configuration as needed for additional core routers.

4. Use the **show mpls ldp neighbor** command to verify MPLS neighbors.

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 192.0.2.11:0
  TCP connection: 192.0.2.11:47619 - 192.0.2.10:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
  LDP Discovery Sources:
    IPv4: (1)
      HundredGigE0/0/0/2
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.64.98.32    192.0.2.13       192.0.2.14    192.0.2.15
      192.0.2.16      192.168.0.1
    IPv6: (0)
```

5. 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 8000 Series Routers.*

6. Configure multiprotocol BGP on the PE routers and route reflectors.

```
Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 10.0.0.1
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 172.16.0.1
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
```

7. Verify BGP neighbor and VPNv4 routes.

```
Router-PE1#show bgp neighbor

BGP neighbor is 172.16.0.1
 Remote AS 2001, local AS 2001, internal link
 Remote router ID 172.16.0.1
  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: 10.0.0.1, Local port: 35018, IF Handle: 0x00000000
  Foreign host: 172.16.0.1, Foreign port: 179
  Last reset 00:00:00
```

```
Router-PE1#show bgp vpnv4 unicast

BGP router identifier 10.0.0.1, 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)
*> 172.16.0.1/12       192.168.0.1                              0 7501 i
*> 172.16.0.1/12       192.168.0.2                              0 7501 i
*> 172.16.0.3/12       192.168.0.3                              0 7501 i
*> 172.16.0.4/12       192.168.0.4                              0 7501 i
*> 172.16.0.5/12       192.168.0.5                              0 7501 i
*>i172.16.0.1/1210.0.0.1                    100      0 8501 i
*>i172.16.0.2/1210.0.0.1                    100      0 8501 i
*>i172.16.0.3/1210.0.0.1                    100      0 8501 i
*>i172.16.0.4/1210.0.0.1                    100      0 8501 i
*>i172.16.0.5/1210.0.0.1                    100      0 8501 i
```

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.

You must configure the **label mode per-vrf** command to effectively manage labels in a VRF environment, optimizing label distribution and simplifying network operations.

Configuration Example

This example shows how to configure MP-BGP on PE1. The loopback address (192.0.2.17) of PE2 is specified as the neighbor of PE1. Similarly, you must perform this configuration on PE2 node as well, with the loopback address (192.0.2.10) of PE1 specified as the neighbor of PE2.

Running Configuration

Verification

- Verify if the BGP state is established, and if the Remote AS and local AS displays the same value (2001 in this example):
- Verify if all the IP addresses are learnt on PE1 from PE2:

The core network is configured for MPLS L3VPN, with verified OSPF, MPLS LDP, FIB, and MP-BGP neighbor and route states.

How MPLS LDP core works

Describes the mechanism by which the MPLS LDP core topology transports MPLS L3VPN services across the provider network, outlining the key components, their roles, and the stages involved.

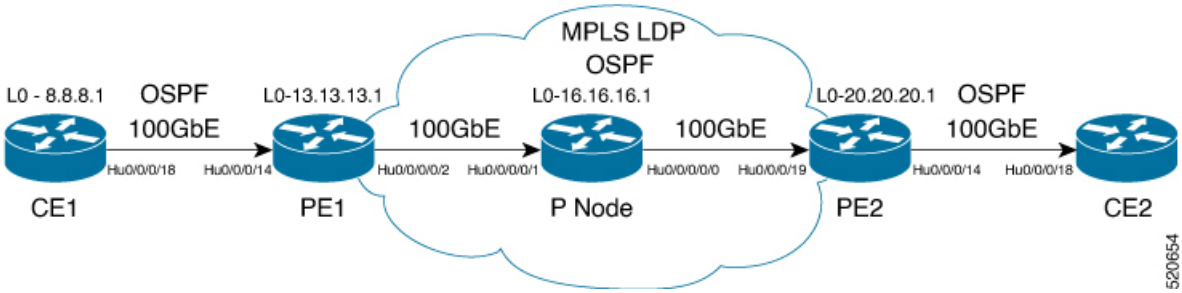
In a typical MPLS network topology, customer VPN traffic is transported using label-switched paths established by LDP across the provider's core infrastructure.

The key components involved in the process are:

- PE routers: Connect customer networks to the provider's MPLS backbone and initiate L3VPN services.
- Provider (P) routers: Form the MPLS backbone, responsible for forwarding labeled packets within the core without awareness of VPN-specific details.
- Label Distribution Protocol (LDP): Dynamically distributes labels and establishes label-switched paths throughout the provider network.

The MPLS LDP (Label Distribution Protocol) core topology enables the transport of MPLS L3VPN (Layer 3 Virtual Private Network) services through provider networks by dynamically assigning labels for packet forwarding.

Figure 1: L3VPN over MPLS LDP



These stages describe how MPLS LDP core works:

1. Establish LDP Sessions: PE and P routers exchange LDP messages to form LDP sessions and synchronize label information.
2. Distribute Routing Information: PE routers use BGP to share VPN routing information and identify endpoints for L3VPN services.
3. Assign and Map Labels: LDP distributes labels that map incoming packets to appropriate label-switched paths across the MPLS core.
4. Forward Labeled Packets: P routers forward packets based solely on labels, ensuring efficient and scalable transport within the provider network.
5. Deliver VPN Traffic: Packets arrive at the destination PE router, where labels are removed and traffic is delivered to the correct customer VPN endpoint.

The MPLS LDP core topology enables efficient and scalable transport of MPLS L3VPN services, allowing multiple customer VPNs to share the provider infrastructure without compromising traffic isolation or service quality.

How OSPF core routing works

Describes the key components and stages of OSPF core routing topology in a network supporting MPLS L3VPN transport.

In a typical core routing scenario, OSPF operates across the core routers to establish and maintain routes. The protocol ensures that the network topology is consistently synchronized, providing a foundation for advanced services like MPLS L3VPN.

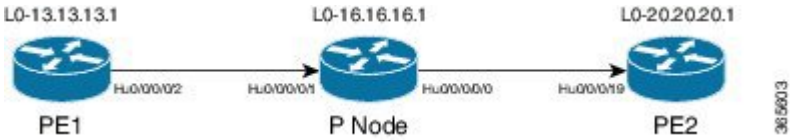
The key components involved in the process are:

- Core routers: Devices that participate in OSPF and maintain the link-state database for the network.

- OSPF process: The protocol logic responsible for exchanging routing information and calculating shortest paths.
- Link-state database: The repository on each router containing the topology information required for route calculation.

OSPF is a widely adopted protocol for core routing in enterprise and service provider networks. As a link-state protocol, OSPF enables efficient route calculation and rapid convergence, ensuring reliable packet delivery within the network core. OSPF was commonly used as the routing protocol before MPLS L3VPN configuration, but IS-IS is also a viable alternative.

Figure 2: OSPF as routing protocol in the core



These stages describe how OSPF core routing works:

1. Initialization: Core routers enable OSPF and establish OSPF process parameters.
2. Neighbor formation: Routers detect adjacent OSPF routers and create neighbor relationships.
3. Exchange of link-state advertisements (LSAs): Routers transmit LSAs to share network topology information.
4. Database synchronization: Each router populates its link-state database based on received LSAs.
5. Route calculation: Using the link-state database, routers compute the shortest path to each destination using Dijkstra's algorithm.
6. Route installation: Calculated routes are installed into the routers' forwarding tables.
7. Convergence and stability: OSPF continually monitors the network for changes and updates routes as needed.

OSPF (or IS-IS) provides a robust core routing protocol that supports reliable MPLS L3VPN transport by maintaining synchronized route information and ensuring seamless connectivity across the network core.

How multiprotocol BGP PE router works

Describes how multiprotocol BGP (MP-BGP) propagates VPN routing and forwarding (VRF) reachability information among provider edge (PE) routers. Outlines the key components, the peering relationships, and each stage involved in distributing VPN routes in a service provider MPLS VPN environment.

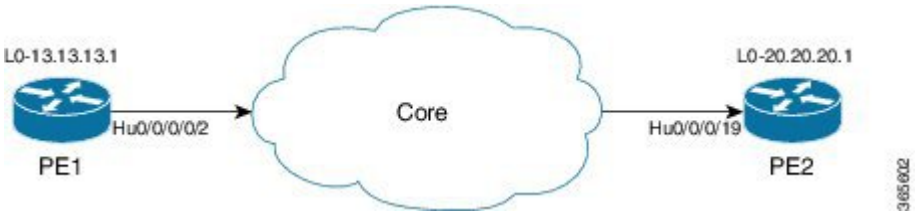
Multiprotocol BGP (MP-BGP) propagates VRF reachability information among PE routers in a VPN community through MP-BGP peering.

The key components involved in this process are:

- PE router: Maintains VRF tables and establishes MP-BGP sessions with other PE routers to exchange customer VPN routes.
- CE router: Connects the customer's network to the provider's network and advertises customer routes to its directly connected PE router.
- MP-BGP session (peering): A communication channel between PE routers that carries VPN route information encoded with customer-specific identifiers (e.g., Route Distinguisher and Route Target).

MP-BGP enables service providers to distribute customer VPN route information securely and efficiently between PE routers, so that customer networks can communicate across a shared provider backbone without leaking private routes outside their VPN communities.

Figure 3: Multiprotocol BGP on PE routers



These stages describe how multiprotocol BGP PE router works:

1. Route advertisement from CE to PE: Each CE router advertises its network prefixes to its directly connected PE router using static routes or protocols such as OSPF or EIGRP.
2. VRF import and route tagging: The PE router imports the customer routes into the correct VRF and tags them with identifiers such as Route Distinguishers and Route Targets.
3. MP-BGP peering across the provider core: The PE router establishes MP-BGP sessions with other PE routers, propagating the tagged customer routes across the provider network.
4. Route import on remote PE: Other PE routers receive the MP-BGP updates, import the routes into the appropriate VRF based on the Route Target, and make the routes available to their directly connected CEs.
5. End-to-end connectivity: The VRF routing tables on each PE router now contain all necessary customer routes; customer devices can communicate across the MPLS/VPN infrastructure as if directly connected.

MP-BGP ensures that customer VPN routes remain isolated and are reachable only within their assigned VPN communities, providing secure and scalable connectivity in a multi-customer service provider environment.

Verify MPLS L3VPN configuration

Details steps to validate MPLS L3VPN configuration, enabling users to confirm correct deployment and identify potential issues within the VPN service.

Ensure that MPLS L3VPN is fully operational by confirming proper traffic forwarding, transport setup, and route state following core configuration.

After configuring MPLS L3VPN core and PE devices, you must verify traffic flow, underlay transport, and overlay routes to confirm successful deployment.

Complete the MPLS L3VPN core and PE configuration.

1. Use the **show mpls forwarding** command on each router to check MPLS forwarding.

   Confirm that traffic is forwarded correctly, labels are assigned, and interfaces show expected switched bytes.

```
Router-P# show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24119  Pop        192.0.2.17/32     Hu0/0/0/0    192.0.2.18     2170204180148
```

```
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                      0
```

2. Verify underlay transport (LDP label distribution and neighbor status).

Confirm that LDP sessions are established with all peers, showing state "Oper".

```
Router-PE1# show mpls ldp neighbor
Peer LDP Identifier: 192.0.2.11:0
  TCP connection: 192.0.2.11:47619 - 192.0.2.10:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
  LDP Discovery Sources:
    IPv4: (1)
      TenGigE0/0/0/2
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.64.98.32    192.0.2.13       192.0.2.14    192.0.2.15
      192.0.2.16      192.0.2.19
    IPv6: (0)
```

```
Router-PE1# show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24036  Pop         192.0.2.11/32      Hu0/0/0/2    192.0.2.12      293294
24037  24165       192.0.2.20/32      Hu0/0/0/2    192.0.2.12      500
24039  24167       192.0.2.17/32      Hu0/0/0/2    192.0.2.12      17872433
       24167       192.0.2.17/32      Hu0/0/0/2.1  192.0.2.21      6345
24041  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                      vrf1601                      0
```

```
Router-PE1# show mpls forwarding labels 24001 hardware egress

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24039  24167       192.0.2.17/32      Hu0/0/0/2    192.0.2.12      N/A
    24167    192.0.2.17/32   Hu0/0/0/2.1  192.0.2.21    N/A

 Show-data Print at RPLC

 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0

Leaf H/W Result:

Leaf H/W Result on NP:0
Label           SwitchAction      EgressIf      Programmed
24039                      0    0x    200185    Programmed

nrLDI eng ctx:
     flags: 0x101, proto: 2, npaths: 0, nbuckets: 1
    ldi_tbl_idx: 0xc37e40, ecd_ref_cft: 0
    pbts_ldi_tbl_idx: 0x0, fastnrldi:0x0

NR-LDI H/W Result for path 0 [index: 0xc37e40 (BE),  common to all NPs]:

 ECMP Sw Idx: 12811840 HW Idx: 200185 Path Idx: 0

NR-LDI H/W Result for path 1 [index: 0xc37e41 (BE),  common to all NPs]:
```

```
 ECMP Sw Idx: 12811841 HW Idx: 200185 Path Idx: 1

SHLDI eng ctx:
    flags: 0x0, shldi_tbl_idx: 0, num_entries:0

SHLDI HW data for path 0 [index: 0 (BE)] (common to all NPs):
Unable to get HW NRLDI Element rc: 1165765120NRLDI Idx: 0
SHLDI HW data for path 1 [index: 0x1 (BE)] (common to all NPs):
Unable to get HW NRLDI Element rc: 1165765120NRLDI Idx: 1

TX H/W Result for NP:0 (index: 0x187a0 (BE)):

 Next Hop Data
 Next Hop Valid:         YES
 Next Hop Index:         100256
 Egress Next Hop IF:     100047
 Hw Next Hop Intf:       606
 HW Port:                0
 Next Hop Flags:         COMPLETE
 Next Hop MAC:           e4aa.5d9a.5f2e

NHINDEX H/W Result for NP:0 (index: 0 (BE)):
NhIndex is NOT required on this platform

NHINDEX STATS: pkts 0, bytes 0 (no stats)

RX H/W Result on NP:0 [Adj ptr:0x40 (BE)]:
Rx-Adj is NOT required on this platform

TX H/W Result for NP:0 (index: 0x189a8 (BE)):

 Next Hop Data
 Next Hop Valid:         YES
 Next Hop Index:         100776
 Egress Next Hop IF:     100208
 Hw Next Hop Intf:       607
 HW Port:                0
 Next Hop Flags:         COMPLETE
 Next Hop MAC:           e4aa.5d9a.5f2d

NHINDEX H/W Result for NP:0 (index: 0 (BE)):
NhIndex is NOT required on this platform

NHINDEX STATS: pkts 0, bytes 0 (no stats)

RX H/W Result on NP:0 [Adj ptr:0x40 (BE)]:
Rx-Adj is NOT required on this platform
```

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

Verify if the label update is received by the FIB:

Verify if label is updated in the hardware:

3. Verify overlay (L3VPN route and label state).

   Confirm that BGP neighbors are established, routes are advertised and learned, and VPN routes appear as expected.

```
Router-PE1# show bgp summary
BGP router identifier 192.0.2.10, 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
192.0.2.22        0  2001   19173    7671    18003    0    0    1d07h
4000
192.0.2.23      0  7001    4615    7773    18003    0    0 09:26:21
125
```

```
Router-PE1# show bgp vpnv4 unicast
BGP router identifier 192.0.2.10, 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)
*> 192.0.2.24/32       192.0.2.25                             0 7501 i
*> 192.0.2.26/32       192.0.2.25                             0 7501 i
*>i20.23.1.1/32       192.0.2.17                   100      0 6553700 11501
 i
*>i20.23.1.2/32       192.0.2.17                   100      0 6553700 11501
 i
```

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

```
Router-PE1# show cef vrf vrf1601 192.0.2.27
192.0.2.27/32, 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 192.0.2.17/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x8c0e3148 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
```

```
    next hop 192.0.2.17/32 via 24039/0/21
    next hop 192.0.2.28/32 Hu0/0/1/1    labels imposed {24059 24031}
```

```
Router-PE2# show mpls lsd forwarding
In_Label, (ID), Path_Info: <Type>
24030, (IPv4, 'default':4U, 192.0.2.10/32), 5 Paths
   1/1: IPv4, 'default':4U, Hu0/0/0/19.2, nh=192.0.2.29, 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
```

```
Router-PE2# show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24019  Pop         192.0.2.30/32      Hu0/0/0/19   192.0.2.31     11151725032
24030  24155       192.0.2.10/32      Hu0/0/0/19   192.0.2.31      3639895
24031  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                      vrf1601                      0
```

Imposition Path: Verify if the BGP neighbor connection is established with the respective neighbor node.

Verify if BGP routes are advertised and learnt.

Verify BGP labels.

Verify if the route is downloaded in the respective VRF.

Disposition Path

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

Verify if the label update is received by the FIB.

MPLS L3VPN configuration is verified as successful when traffic-flow, underlay, and overlay command outputs confirm proper forwarding, label, and BGP state.

L3VPN over RSVP-TE

Describes integrating L3VPN with RSVP-TE, including configuration procedures to establish advanced VPN services over traffic-engineered MPLS tunnels.

L3VPN over RSVP-TE is an MPLS L3VPN transport option that

- uses MPLS-TE to map traffic flows to paths based on network resources
- uses RSVP to signal MPLS-TE label switched paths (LSPs), and
- reserves resources for local and remote data flows.

Feature history

The feature history table lists release support for this feature.

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
L3VPN over RSVP-TE	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
L3VPN over RSVP-TE	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M
L3VPN over RSVP-TE	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
L3VPN over RSVP-TE	Release 7.3.2	Using labeled switch paths (LSPs), this feature enables resource reservations in each node across data paths on MPLS-configured Layer 3 VPNs. Such reservations allow service providers to offer high throughput to their subscribers with optimal network operations.

RSVP-TE details

MPLS Traffic Engineering (MPLS-TE) learns the topology and resources available in a network, then maps traffic flows to particular paths based on network resources. MPLS-TE builds unidirectional tunnels from source to destination in the form of label switched paths (LSPs), which are then used to forward traffic. MPLS-TE uses RSVP to signal LSPs.

RSVP processes protocol messages from other systems, handles resource requests from local clients, and generates protocol messages. As a result, resources are reserved for data flows on behalf of both local and remote clients. RSVP creates, maintains, and deletes these resource reservations. RSVP is automatically enabled on interfaces where MPLS-TE is configured.

For more information on RSVP-TE and MPLS-TE, see the *MPLS Configuration Guide for Cisco 8000 Series Routers*.

Configure L3VPN over RSVP-TE

Configure L3VPN over RSVP-TE by preparing core routing, deploying label distribution, enabling MP-BGP, supporting PE-CE protocols, and establishing RSVP-TE tunnels for end-to-end MPLS VPN connectivity.

Set up Layer 3 VPN (L3VPN) service over MPLS Traffic Engineering tunnels using RSVP-TE to optimize routing, resource utilization, and traffic management.

MPLS Traffic Engineering (MPLS-TE) builds label switched paths (LSPs) signaled with RSVP, allowing for efficient mapping of traffic flows. Each LSP acts as a unidirectional tunnel across the core. L3VPN, delivered over RSVP-TE, combines these tunnels with MP-BGP for multi-site connectivity.

For more information on RSVP-TE and MPLS-TE, see the *MPLS Configuration Guide for Cisco 8000 Series Routers*.

Plan the core routing protocol, MPLS label distribution method, MP-BGP peering, PE-CE protocol support, and tunnel destinations before you begin.

1. Configure core routing protocols.

   Ensure all core routers have consistent, reachability-focused routing configuration. Use standard IGP (OSPF or IS-IS). For specifics, see the *Routing Configuration Guide for Cisco 8000 Series Routers*.

2. Enable MPLS and configure label distribution.

   Enable MPLS on all core interfaces. Decide between:

   - MPLS LDP—See the *Implementing MPLS Label Distribution Protocol* chapter in the *MPLS Configuration Guide for Cisco 8000 Series Routers* for configuration information.
   - MPLS Traffic Engineering Resource Reservation Protocol (RSVP)—See the *Implementing RSVP for MPLS-TE* chapter in the *MPLS Configuration Guide for Cisco 8000 Series Routers* for configuration information.

3. Configure RSVP-TE tunnels in the core.

   - Signal label switched paths using RSVP-TE from each ingress PE.
   - RSVP is automatically enabled on interfaces with MPLS-TE configuration.

4. Set up MP-BGP for VPNv4 routes.

   Configure MP-BGP peering between all PEs to carry VPN routes across the MPLS core.

5. Configure PE-CE protocol support.

   Set up appropriate routing protocols or static routes for PE-CE connections according to your solution design.

6. Verify and test end-to-end L3VPN connectivity.

   Ensure services are reachable and that traffic engineering objectives are met.

After all steps are complete, L3VPN transport runs over RSVP-TE tunnels across your MPLS core, allowing for optimal routing, resiliency, and cross-domain VPN connectivity.

VRF-lite

Explains VRF-lite fundamentals, covering interface requirements, BGP label allocation principles, and configuration tasks for deploying lightweight VPN segmentation without a full MPLS core.

VRF-lite is a VRF deployment model that

- uses VRFs without MPLS
- supports two or more VPNs with overlapping IP addresses, and
- places Layer 3 interfaces into one VRF at a time.

Additional reference information

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 interface and BGP label allocation

Outlines the requirements for assigning interfaces and BGP label allocation modes in a VRF-lite environment.

To ensure proper VRF-lite operation when using BGP:

- Use only Layer 3 interfaces for VRF-lite configurations.
- Never assign an interface to more than one VRF at any time.
- You may configure multiple interfaces to be part of the same VRF, but all interfaces must participate in the same VPN.
- If you use the BGP protocol with VRF-lite, change the BGP label allocation mode to per-VRF.

Configure VRF-lite

Configure VRF-lite by creating VRFs, assigning interfaces, configuring routing protocols, and verifying route tables.

Set up VRF-lite on a router so each customer's traffic and routing are kept separate using virtual routing tables.

Customers often use VRF-lite to isolate traffic and routing information between different customers or departments when connecting multiple VPN sites to the same PE router. Each VRF represents a separate customer or use case, ensuring their routing and interfaces remain independent.

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

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

- Plan customer VRFs, route targets, interfaces, subinterfaces, and routing protocols.
- Ensure required route-policy (such as `pass-all`) exists.
- Have interface and IP details ready for each VRF.

1. Create each customer 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
```

   Repeat for VRF 'vrf2' or any additional VRFs.

2. Assign interfaces to VRFs

```
Router#configure
Router(config-subif)#interface TenGigE0/0/0/0.2001
```

```
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
```

Repeat for other interfaces and VRFs as needed.

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

3. Configure routing protocols under VRF instances.

```
Router#configure
Router(config)#router ospf 100 area 0
Router(config-ospf-ar)#interface loopback 0
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface TenGigE0/0/0/1
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface TenGigE0/0/0/1.2001
Router(config-ospf-ar-if)#vrf vrf1
Router(config-ospf-vrf)#default-information originate
Router(config-ospf-vrf)#exit
Router(config-ospf)#exit
Router(config)#router ospf 100 area 0
Router(config-ospf-ar)#interface TenGigE0/0/0/1.2000
Router(config-ospf-ar-if)#vrf vrf2
Router(config-ospf-vrf)#default-information originate
Router(config-ospf-vrf)#commit
```

Repeat for each VRF and associated interface.

4. Review the 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 ospf 100 area 0
interface Loopback0
!

interface TenGigE0/0/0/1
!
interface TenGigE0/0/0/1.20001
 vrf vrf1
 default-information originate
  !

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

5. Verify VRF route tables.

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

Codes: C - connected, S - static, 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, 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
```

VRF-lite is successfully configured when each VRF contains the expected interfaces and routes in its own routing table, and customer traffic remains isolated.

MPLS L3VPN services using Segment Routing

Introduces MPLS L3VPN with Segment Routing, detailing topology operations, guided MPLS core configuration using Segment Routing, and verification procedures to ensure correct L3VPN deployment over Segment Routing.

An MPLS L3VPN service with Segment Routing is a transport approach that

- uses Segment Routing instead of MPLS LDP for MPLS L3VPN transport
- applies Segment Routing directly to the MPLS architecture without changing the forwarding plane, and
- uses IGP or BGP for label distribution.

Additional reference information

Currently, MPLS Label Distribution Protocol (LDP) is the widely used transport for MPLS L3VPN services. You can achieve better resilience and convergence for 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 protocols are not required; instead, label distribution is performed by IGP (IS-IS or OSPF) or BGP. Removing protocols from the network simplifies its operation and makes it more robust and stable by eliminating the need for protocol interaction. Segment Routing also utilizes network bandwidth more effectively than traditional MPLS networks and offers lower latency.

How MPLS L3VPN over Segment Routing works

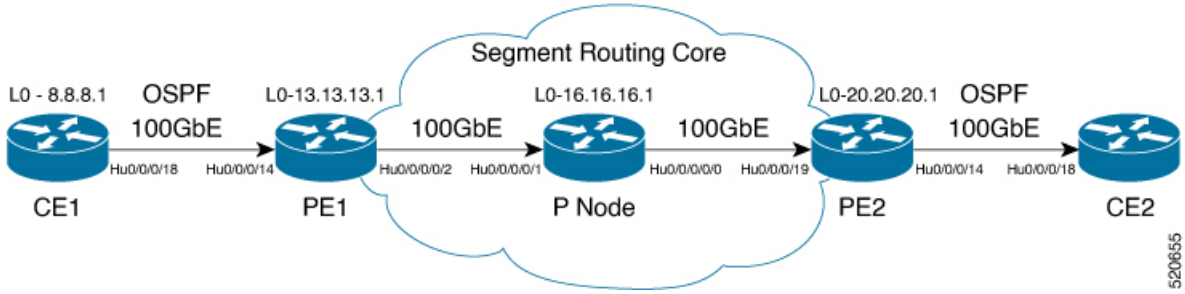
Describes the CE, PE, P, OSPF, autonomous system, VRF, route-target, and loopback roles in the Segment Routing topology.

Here is a network scenario in which an MPLS L3VPN service is transported using Segment Routing.

The key components involved in the process are:

- CE routers: Devices (e.g., CE1 and CE2) at the customer premises connecting to the provider's network.
- PE routers: ISP routers (example, PE1 and PE2) interfacing between customer and core networks, hosting customer VPNs.
- P routers: ISP core routers forwarding labeled packets between PE routers.
- OSPF protocol: Used as an Interior Gateway Protocol (IGP) to facilitate routing between CE and PE routers.
- Autonomous Systems (AS): Distinct routing domains for customer (AS 65534) and ISP (AS 65000).
- Virtual Routing and Forwarding (VRF): Logical instances (example, vrf1601) on PE routers to isolate customer traffic.
- Route-targets: BGP attributes controlling VPN route import/export between VRFs.
- Loopback interfaces: Simulated network addresses used for testing and routing stability.

Figure 4: MPLS L3VPN over Segment Routing



These stages describe how MPLS L3VPN over Segment Routing works:

1. Topology setup: CE1 and CE2 are deployed as customer routers connecting to the ISP network that contains two PE routers (PE1, PE2) and a P router.
2. IGP configuration: OSPF is configured as the IGP between CE and PE routers, supporting label distribution through IS-IS, OSPF, or BGP (OSPF is used here).
3. VPN separation: The customer's autonomous system (65534) peers with the ISP's autonomous system (65000) using VRF peering, preventing route advertisement into the global IPv4 table. PE routers host customer VRF instances and import/export route targets as needed.
4. Loopback implementation: Loopback interfaces are designated to simulate network endpoints and ensure reliable routing.

The topology is ready for configuration when the CE, PE, P, OSPF, autonomous system, VRF, route-target, and loopback roles are planned, enabling secure, isolated transport of customer VPN services using Segment Routing.

Configure Segment Routing on PE1, P, and PE2 routers

Enable Segment Routing in the MPLS core on PE1, P, and PE2 routers and review the running configuration.

Configure Segment Routing on core routers (PE1, P, and PE2) in the MPLS network to enable traffic engineering and segment routing.

Perform these steps on each designated core router (PE1, P, and PE2) to activate Segment Routing using OSPF or IS-IS routing protocols. Ensure you use the correct values (router ID, prefix SID index, global block, and interface configurations) for each device.

Plan the OSPF or IS-IS core, Segment Routing global block, prefix SID indexes, and core interfaces for PE1, P, and PE2.

Follow these steps to enable Segment Routing in the MPLS core on PE1, P, and PE2 routers:

1. Configure OSPF for Segment Routing on each router.

```
Router-PE1# configure
Router-PE1(config)# router ospf dc-sr
Router-PE1(config-ospf)# router-id 192.0.2.10
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 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
```

2. Configure the Segment Routing global block on each router.

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

3. Configure IS-IS for Segment Routing on each router (if using IS-IS).

```
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 sr-prefer
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
```

4. Review the running configuration.
   PE1

```
router ospf dc-sr
 router-id 192.0.2.10
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
```

```
 segment-routing 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 sr-prefer
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30101
  !
 !
```

P node

```
router ospf dc-sr
 router-id 192.0.2.11
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing 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 sr-prefer
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30102
  !
 !
```

PE2

```
router ospf dc-sr
 router-id 192.0.2.17
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing 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 sr-prefer
 !
 interface Loopback0
```

```
 address-family ipv4 unicast
  prefix-sid index 30103
 !
```

Segment Routing is configured and active in the MPLS core when the running configuration shows Segment Routing enabled and appropriate values on PE1, P, and PE2 routers.

Verify MPLS L3VPN configuration over Segment Routing

Verify Segment Routing transport counters and VPN label counters for MPLS L3VPN over Segment Routing.

Ensure MPLS L3VPN over Segment Routing is functioning by monitoring label counter increments on core and edge routers.

This verification task confirms that transport and VPN label counters are increasing, which indicates proper operation of MPLS L3VPN over Segment Routing. You must complete the initial configuration before proceeding.

Complete MPLS L3VPN over Segment Routing configuration before you verify label counters.

Follow these steps to verify MPLS L3VPN label counters over Segment Routing:

1. On the core router (P node), verify the statistics for the IGP transport label, and confirm that label 64003 (in this example) is increasing.

```
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   192.0.2.32      572842
```

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

   P node:

2. On PE1, verify the statistics for MPLS labels. Confirm that label counters such as 64001 and 60003 are increasing.

```
Router-PE1# 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    192.0.2.12      532978
```

3. On PE2, verify the statistics for the VPN label, and confirm that label 24031 (in this example) is increasing.

```
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                      0
```

If the IGP transport label and VPN label counters increase as expected on core and edge routers, MPLS L3VPN over Segment Routing is verified and operating correctly.

3 Inter-AS L3VPN Fundamentals and Route Reflector Exchange

Topics:

Introduces Inter-AS L3VPN architecture, covering VPN connectivity across autonomous systems, ASBR and route reflector roles, BGP route and label exchange, and configuration procedures for scalable MPLS VPN services in multi-provider environments.

Inter-AS L3VPN

Explains Inter-AS L3VPN capabilities, including how VPN connectivity is extended across multiple autonomous systems and service provider backbones, the foundational concepts, supported options, and use cases for seamless multi-provider L3VPN deployments.

Inter-AS L3VPN is a cross-provider VPN capability that

- extends VPN connectivity across multiple autonomous systems
- supports VPNs in different geographic areas or service provider networks, and
- keeps VPN connectivity independent of the complexity and location of the sites.

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.

Feature history

The feature history table lists release support for this feature.

Table 4: Feature History Table

Feature Name	Release	Description
Inter-AS Support for L3VPN	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on: - 8711-48Z-M - 8011-32Y8L2H2FH - 8011-12G12X4Y-A/D
Inter-AS Support for L3VPN	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release	Description
Inter-AS Support for L3VPN	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) You can now enhance cross-provider VPN connectivity with Inter-AS Option - A and C, enabling seamless Layer 3 VPN communication between different autonomous systems. This feature facilitates the exchange of VPN routing information and forwarding data across provider boundaries, enhancing flexibility and scalability in network designs. Inter-AS Option - A utilizes static routing for simple setups, while Option - C provides comprehensive MPLS label-swapping capabilities for more complex configurations. These options support effective management of multi-provider network. *Previously this feature was supported on Q200 and Q100. It is now extended: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

Additional reference information

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, Inter-AS L3VPN provides a seamless connection between autonomous systems.

Benefits of Inter-AS L3VPN

Lists the features and advantages of using Inter-AS L3VPN for VPN connectivity across multiple service provider backbones and autonomous systems.

An MPLS VPN Inter-AS provides these benefits:

- Enables a VPN to cross more than one service provider backbone, allowing service providers operating separate autonomous systems to jointly offer MPLS VPN services to the same customer. A VPN can begin at one customer site and traverse different service provider backbones before reaching another site for the same customer. Previously,

MPLS VPN could traverse only a single BGP autonomous system backbone. Inter-AS L3VPN enables multiple autonomous systems to form a continuous, seamless network between customer sites.

- Allows a VPN to exist in different geographic areas. By directing all VPN traffic through a single point between areas, the service provider can better control network traffic rates between locations.

Autonomous system boundary routers

Explains how autonomous system boundary routers exchange VPN reachability between separate autonomous systems.

An autonomous system boundary router is an inter-AS border router that

- uses eBGP to exchange VPN reachability with ASBR peers
- connects autonomous systems that use IGP internally, and
- supports route exchange across service provider boundaries.

Additional reference information

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 eBGP. eBGP allows service providers to set up an interdomain routing system that guarantees loop-free exchange of routing information between separate autonomous systems. The primary function of 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.

How MPLS VPN Inter-AS BGP label distribution works

Describes the process by which ASBRs and route reflectors exchange IPv4 routes, MPLS labels, and VPN-IPv4 routes for scalable and efficient MPLS VPN operation in Inter-AS systems.

In large-scale service provider networks, connecting multiple autonomous systems (ASes) while supporting VPN services presents operational challenges. Traditional methods may require ASBRs to store and forward all VPN-IPv4 routes, leading to scalability and configuration complexity. MPLS VPN Inter-AS BGP label distribution addresses these challenges by allowing route reflectors to manage and disseminate VPN-IPv4 routes, while ASBRs focus solely on exchanging labeled IPv4 routes. This approach improves scalability, simplifies configuration at network boundaries, and supports the seamless transport of VPN traffic—even across non-MPLS core networks or through third-party providers—without introducing additional label distribution protocols.

The key components involved in the process are:

- Autonomous System Boundary Routers (ASBRs): Exchange IPv4 routes with MPLS labels between different provider networks.
- Route reflectors (RRs): Store and forward VPN-IPv4 routes to provider edge routers using multihop, multiprotocol eBGP.

- Provider edge (PE) routers: Connect customer sites to the provider network and receive labeled routes for forwarding VPN traffic.

MPLS VPN Inter-AS BGP label distribution enables efficient exchange of IPv4 and VPN-IPv4 routing information between different autonomous systems, supporting network scalability and simplified configuration.

These stages describe how MPLS VPN Inter-AS BGP label distribution works:

1. Setup: ASBRs are configured to exchange IPv4 prefixes and their associated MPLS labels between autonomous systems.
2. Route reflection: Route reflectors use multihop, multiprotocol external BGP to exchange VPN-IPv4 routes. RRs hold VPN-IPv4 routes and share them with the relevant PE routers.
3. Label distribution: ASBRs distribute MPLS labels for IPv4 routes, eliminating the need for additional label distribution protocols between adjacent label switch routers (LSRs) that are also BGP peers.
4. Scalability and simplification: Having RRs store the VPN-IPv4 routes improves scalability and simplifies configuration at the network border. A non-MPLS VPN transit network can also carry labeled VPN traffic.
5. Inter-AS traffic handling: The setup enables IPv4 routes with MPLS labels to traverse non-MPLS VPN core networks or service providers and supports VPN traffic forwarding without requiring additional protocols.

ASBRs exchange IPv4 routes with MPLS labels while route reflectors efficiently manage and distribute VPN-IPv4 routes, supporting scalable MPLS VPN services across multiple autonomous systems.

Exchanging IPv4 routes with MPLS labels

Describes how route reflectors, ASBRs, and PE routers exchange VPN-IPv4 routes, IPv4 routes, and MPLS labels across autonomous systems in an MPLS VPN environment.

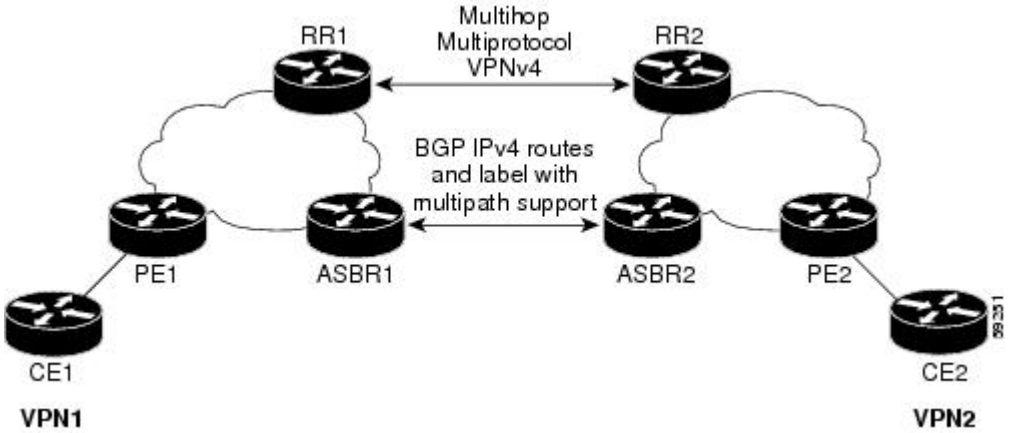
You can set up a VPN service provider network to exchange IPv4 routes with MPLS labels. You can configure the VPN service provider network as follows:

The key components involved in the process are:

- Route reflector (RR): Facilitates distribution of VPN-IPv4 routes, IPv4 routes, and MPLS labels among PE routers, especially across different autonomous systems.
- Autonomous System Boundary Router (ASBR): Exchanges routing and MPLS label information between autonomous systems, using either eBGP, IGP with LDP, or iBGP label distribution.

  Provider edge (PE) router: Maintains local and remote VPN-IPv4 routes and MPLS labels to support correct traffic forwarding within the VPN.

Figure 5: VPNs Using eBGP and iBGP to Distribute Routes and MPLS Labels



These stages describe exchanging IPv4 routes with MPLS labels:

1. Route distribution through eBGP: Route reflectors exchange VPN-IPv4 routes and MPLS labels between autonomous systems using multihop, multiprotocol eBGP sessions. This preserves next-hop information and VPN labels as they traverse different ASes.
2. Synchronization of routing and label information: Each local PE router must learn the relevant route and label information for remote PE routers in the VPN.
3. Route and label exchange between ASBRs and PEs: There are two main approaches:
   - IGP and LDP redistribution: The ASBR redistributes IPv4 routes and corresponding MPLS labels learned from eBGP into the local IGP and LDP, and vice versa.
   - iBGP IPv4 label distribution: The ASBR and PE establish direct iBGP sessions to exchange VPN-IPv4 and IPv4 routes alongside their MPLS labels.
4. Reflection of learned routes: Route reflectors can reflect IPv4 routes and MPLS labels learned from the ASBRs to other PE routers, enabling scalable and efficient route propagation throughout the VPN.

VPN-IPv4 routes and MPLS labels are efficiently exchanged across autonomous systems, while next-hop information is preserved, enabling end-to-end MPLS VPN connectivity between sites in different ASes.

BGP routing information

Lists the BGP route attributes included in routing information.

BGP routing information includes the following items:

- Network number (prefix): The IP address of the destination.
- Autonomous system (AS) path: 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 is farthest from the local router and usually the AS where the route began.
- Path attributes: Additional information about the AS path, such as the next hop.

BGP messages and MPLS labels

Lists the types of BGP messages, explains how MPLS labels are included in update messages, and details the key functions of each BGP message type.

MPLS labels are included in the update messages that a router sends to its neighbors as part of BGP operations. Routers exchange the following types of BGP messages:

- Open messages–After a router establishes a TCP connection with a neighboring router, both 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 learns about a new, changed, or broken route, it sends an update message to its neighbor. This message contains the Network Layer Reachability Information (NLRI), which lists usable IP routes, includes routes that are no longer usable, and contains path attributes and the lengths of usable and unusable paths. MPLS labels for VPN-IPv4 routes are encoded in the update message, as described in RFC 2858, and for IPv4 routes as described in RFC 3107.
- Keepalive messages–Routers exchange keepalive messages at regular intervals to determine if a neighboring router is still available to exchange routing information. The keepalive message contains only a message header and no routing data. For Cisco routers, the default interval is 60 seconds.
- Notification messages– If a router detects an error, it sends a notification message to its neighbor to inform them of the problem.

How BGP sends MPLS labels with routes

Describes how BGP carries MPLS label mapping information in update messages.

In MPLS-enabled networks, BGP distributes both route information and the associated MPLS labels so routers can properly forward packets using label-switching. This process ensures efficient setup and operation of label-switched paths.

The key components involved in the process are:

- BGP routers (eBGP and iBGP peers): Exchange routing and MPLS label mapping information.
- MPLS labels: Identify the path for packet forwarding in the MPLS network.
- BGP update messages: Carry routing and label mapping information between peers.

BGP distributes MPLS label mapping information along with routing updates, allowing for efficient label-switched path setup in MPLS-enabled networks.

These stages describe wow BGP sends MPLS labels with routes:

1. Both BGP routers negotiate their capability to send and receive MPLS label information by advertising label capability to each other.
2. Once label capability is negotiated, each router includes MPLS label mapping information in its BGP update messages that advertise routes.
3. The receiving router processes the BGP update, associates the MPLS label with the advertised route, and uses this information for MPLS forwarding.
4. If the next-hop is unchanged in the routing update, the associated MPLS label remains unchanged.
5. Throughout the session, both routers ensure that all outgoing route updates include appropriate MPLS label mappings as long as label capability is established.

When both BGP peers advertise and recognize label capability, BGP includes MPLS label mapping information in routing updates, enabling MPLS forwarding along advertised paths.

Configure VPN connectivity with ASBRs exchanging IPv4 routes and MPLS labels

Details procedures for configuring BGP route reflectors to exchange and reflect VPN-IPv4 routes across autonomous systems, preserving next-hop and label information to ensure seamless MPLS VPN operation.

Enable VPN connectivity across Autonomous Systems by configuring route reflectors to facilitate the exchange and reflection of VPN-IPv4 routes.

This procedure does not apply to Inter-AS VPN connectivity over IP tunnels. It is intended for scenarios where ASBRs exchange IPv4 routes and MPLS labels using route reflectors.

Plan the settings for route reflectors, ASBRs, PEs, VPN-IPv4 or VPNv6, eBGP multihop, and route policies.

1. Configure route reflectors to exchange VPN-IPv4 routes.

   For detailed instructions, see Configure route reflectors for VPN-IPv4 route exchange on page 48.

2. Configure the route reflectors to reflect remote routes within an AS.

   For detailed instructions, see Configure route reflectors to reflect remote routes within an AS on page 50.

VPN connectivity is provided when route reflectors exchange VPN-IPv4 routes and reflect remote routes within the AS.

Configure route reflectors for VPN-IPv4 route exchange

Configure route reflectors to exchange VPN-IPv4 routes by using multihop, while preserving next-hop information and VPN labels across autonomous systems.

Establish BGP route reflectors that exchange VPN-IPv4 routes between autonomous systems with next-hop and VPN label preservation.

Route reflectors enable the exchange of VPN-IPv4 and VPN-IPv6 routes between BGP peers across different autonomous systems. This setup is essential for multi-AS MPLS VPN deployments, ensuring that routes are properly reflected and next-hop and label information remains intact for seamless forwarding.

Plan the route reflector peer, remote AS, loopback update source, VPNv4 and VPNv6 address families, route policies, and next-hop behavior before you begin.

1. Configure route reflectors to exchange VPN-IPv4 routes.

```
Router# configure
Router(config)# router bgp 500
Router(config-bgp)#
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# ebgp-multihop
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
```

2. Review the running configuration.

```
Router#show run router bgp 500
router bgp 500
bgp router-id 192.0.2.10
address-family ipv4 labeled-unicast
  allocate-label all
!
address-family vpnv4 unicast
!
address-family ipv6 unicast
!
address-family vpnv6 unicast
!
neighbor 10.200.1.1
  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
  !
```

3. Verify the configuration and the state of routes and labels.

```
Router#show cef vrf vrf2001 ipv4 172.16.0.1/32 hardware egress location0/0/CPU0
192.0.2.11/32, 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 192.0.2.12/32 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
```

Route reflectors exchange VPN-IPv4 routes across autonomous systems, maintaining next-hop and VPN label information. Verification commands confirm the expected state of route and label propagation.

Configure route reflectors to reflect remote routes within an AS

onfigure route reflectors to reflect IPv4 routes and labels learned from an ASBR to PE routers within the autonomous system.

Configure a route reflector so it can reflect IPv4 routes and labels learned from an ASBR to PE routers in the same autonomous system.

Use this task to enable a route reflector (RR) to pass along IPv4 routes and labels, learned by the autonomous system boundary router (ASBR), to provider edge (PE) routers in the autonomous system. This is accomplished by configuring the ASBR and PE routers as route reflector clients of the RR.

Plan the ASBR and PE route reflector clients, labeled-unicast address families, VPNv4 address family, and BGP router ID before starting this task.

1. Configure route reflectors to reflect remote routes within an AS.

```
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 192.0.2.13
```

```
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 192.0.2.14
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
```

2. Review the running configuration.

```
Router#show run router bgp 500
router bgp 500
 bgp router-id 192.0.2.10
 address-family ipv4 unicast
  allocate-label all
 !
 address-family vpnv4 unicast
 !
 neighbor 192.0.2.13
  remote-as 500
  update-source Loopback0
  !
  address-family ipv6 labeled-unicast
   route-reflector-client
  !
  address-family vpnv6 unicast
  !
 !
 neighbor 192.0.2.14
  remote-as 500
  update-source Loopback0
  address-family ipv4 labeled-unicast
   route-reflector-client
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
```

Remote routes are reflected within the autonomous system when the route reflector client configuration appears in the running configuration.

Configure Inter-AS VPN connectivity by defining a static route to an ASBR peer

Outlines steps for establishing Inter-AS VPN connectivity by configuring a static route to an ASBR peer, including CLI procedures for directing traffic toward remote autonomous systems.

Define a static route that directs traffic intended for the ASBR peer, facilitating Inter-AS VPN connectivity within your network.

Use this task when you need to manually specify a static route to an ASBR peer, typically as part of configuring or troubleshooting Inter-AS VPN connections on a router.

- Identify the ASBR peer prefix (destination network) and the next-hop IP address for the static route.
- Ensure you have appropriate privileges to access and configure router settings through the CLI.

Configure a static route to an ASBR peer.

```
Router# configure
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 10.10.10.10/32 10.9.9.9
Router(config-static-afi)# commit
```

The static route to the ASBR peer is configured once the route is committed. Traffic destined for the ASBR peer will now be routed according to your specification.

4 Inter-AS Option B for L3VPN

Topics:

Explains Inter-AS Option B for L3VPN, outlining core functions, topology mechanisms, and configuration procedures to enable seamless Layer 3 VPN integration across autonomous systems.

Inter-AS Option B for L3VPN

Introduces Inter-AS Option B for L3VPN, describing its primary functions and operational topology, including how data flows between autonomous systems and key mechanisms for connectivity.

An Inter-AS Option B method is an Inter-AS VPN approach that

- uses ASBR ports and router subinterfaces to receive MPLS traffic
- employs MP-BGP sessions to distribute labeled VPN prefixes between ASBRs, and
- assigns a VPN label each time the BGP next hop changes.

Feature history

The feature history table lists release support for this feature.

Table 5: Feature History Table

Feature Name	Release	Description
Inter-AS Support for L3VPN	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on: - 8711-48Z-M - 8011-32Y8L2H2FH - 8011-12G12X4Y-A/D
Inter-AS Support for L3VPN	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release	Description
Inter-AS Support for L3VPN	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) You can now enhance cross-provider VPN connectivity with Inter-AS Option - A and C, enabling seamless Layer 3 VPN communication between different autonomous systems. This feature facilitates the exchange of VPN routing information and forwarding data across provider boundaries, enhancing flexibility and scalability in network designs. Inter-AS Option - A utilizes static routing for simple setups, while Option - C provides comprehensive MPLS label-swapping capabilities for more complex configurations. These options support effective management of multi-provider network. *Previously this feature was supported on Q200 and Q100. It is now extended: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

Additional reference information

A Multiprotocol Label Switching (MPLS) Layer 3 VPN consists of multiple interconnected sites using an MPLS provider core network. Within each customer site, one or more customer edge (CE) routers attach to provider edge (PE) routers. Inter-AS Option B is one of the methods for sharing VPN routes among sites.

When configuring Inter-AS Option B, router subinterfaces enable ASBR ports to receive MPLS traffic. ASBRs use MP-BGP sessions to distribute labeled VPN prefixes with each other. A VPN label is assigned whenever the BGP next hop is changed.

Functions of Inter-AS Option B

Lists the functions and label-distribution behaviors of Inter-AS Option B.

- This feature allows an iBGP VPNv4 session between routers within an autonomous system (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 included in the same BGP update message that distributes the route itself.

- When BGP distributes a particular route, it also distributes an MPLS label mapped to that route. Many ISPs prefer this method because it ensures complete IGP isolation between different sites.

How Inter-AS Option B works

Describes how Inter-AS Option B topologies enable MPLS VPN services across two ISPs, detailing the distinct roles of CE, PE, P, and ASBR routers in the exchange of routes and MPLS labels.

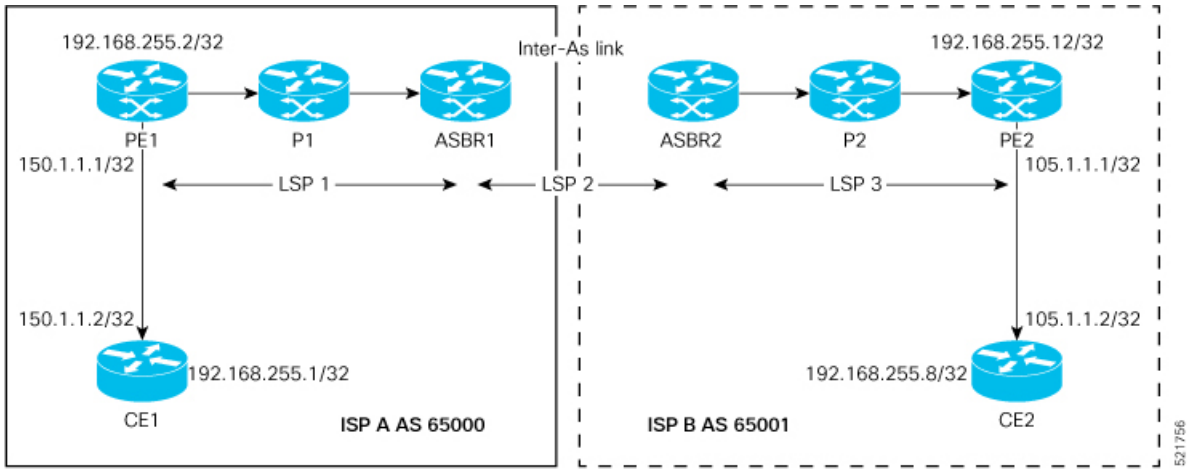
In Inter-AS Option B deployments, Layer 3 VPN (L3VPN) services are extended across multiple autonomous systems by connecting provider edge (PE), provider (P), and autonomous system boundary routers (ASBRs) in each domain. This configuration requires careful alignment of routing protocols (such as IS-IS and BGP), MPLS label distribution, and VRF assignments to ensure seamless VPN connectivity. Properly coordinating these settings across all relevant routers is essential for successful inter-domain transport and isolation of VPN traffic, while also addressing resource limitations related to label handling.

Inter-AS Option B enables L3VPN services by connecting two different autonomous systems (AS) and exchanging both routing and MPLS label information across their boundaries. The architecture uses a combination of protocols, role-specific routers, and selective label distribution mechanisms to ensure traffic can traverse ISP networks seamlessly.

The key components involved in the process are:

- Customer edge routers: Connect customer sites to the provider network using eBGP as the routing protocol.
- Provider edge (PE) routers: Exchange VPN routes with customer sites and the ASBRs via MP-iBGP and eBGP VPNv4 sessions.
- Provider (P) routers: Function within the ISP core, running IGP (such as IS-IS) and LDP to maintain label-switched paths.
- Autonomous System Border Routers (ASBRs): Interconnect the two ISPs' MPLS networks, exchange VPN information via eBGP VPNv4 peering, and handle label exchange over the inter-AS link.
- IGP and LDP: Provide label distribution and routing within each ISP's domain.
- MP-BGP: Carries VPNv4 routes and labels between PEs and ASBRs within a customer network.
- 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.

Figure 6: L3VPN Inter-AS Option B



These stages describe how Inter-AS Option B works:

1. CE to PE Routing: eBGP is configured between CE and PE routers, enabling route exchange at the customer-provider boundary.

2. Core IGP/LDP Setup: All core ISP links run IS-IS (IGP) with LDP to build MPLS label-switched paths except on the inter-ASBR link.
3. ASBR interconnection: Between ASBR1 and ASBR2, LDP and IGP are not configured. Instead, the ASBRs form eBGP VPNv4 peering and exchange labels using BGP; MPLS is not enabled with LDP on this inter-AS link.
4. Route exchange and label binding:
   - MP-iBGP distributes routes between PE and ASBR within each ASN.
   - On ASBRs, eBGP VPNv4 sessions allow VPN label exchange between ISPs.
   - When eBGP VPNv4 peering is established, the `mpls bgp forwarding` feature automatically configures the inter-AS link for BGP label exchange.
5. Route Policy Requirements: On platforms such as Cisco IOS XR, a route policy (for example, **pass-all**) must be configured for eBGP peers to send and receive routing updates.
6. Route handling considerations:
   - No VRF is required on ASBRs.
   - The **retain route-target all** command on the ASBR prevents the automatic dropping of updates from VRFs without a locally configured route target (RT).
   - A static /32 route to the remote ASBR's next-hop address is needed to correctly bind MPLS labels; without this, the control plane may function, but traffic forwarding fails.
7. Platform limitations: Note that BGP-LU is not supported as an underlay for Inter-AS Option B topologies.

The Inter-AS Option B topology enables seamless end-to-end MPLS VPN services across multiple ISPs by efficiently utilizing specialized routing and label distribution mechanisms. Proper setup of the routing and label exchange ensures full interoperability and secure VPN connectivity between networks.

Configure Inter-AS Option B for L3VPN

Provides step-by-step instructions to configure Inter-AS Option B for L3VPN, detailing necessary tasks to establish Layer 3 VPN connectivity between autonomous systems.

Establish L3VPN connectivity between two autonomous systems using Inter-AS Option B, ensuring correct routing, label distribution, and forwarding.

In an Inter-AS Option B deployment, multiple autonomous systems (AS) are interconnected to extend Layer 3 VPN services. The procedure requires configuring routing protocols, MPLS label distribution, BGP VPNv4/VPNv6 address families, VRFs, and validation via control-plane and data-plane verification.

> **Note**
>
> Although L3VPN Inter-AS Option B supports per-prefix mode, use per-next-hop-label mode because of resource limitations.

- Plan the topology, including all PE, P, and ASBR devices and their interconnections.
- Determine interface addressing, IS-IS implementation, LDP sessions, MP-BGP configuration, VRF details, route-targets, and appropriate label modes.

- Ensure resource limitations are addressed by using per-next-hop-label mode instead of per-prefix mode when supported (due to scaling restrictions).

Follow these steps to configure and verify Inter-AS Option B for L3VPN:

1. Configure PE1 for Inter-AS Option B.

```
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 192.0.2.12
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
```

2. Configure P1 for Inter-AS Option B.

```
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 192.0.2.13 255.240.0.0
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
```

3. Configure ASBR1 for Inter-AS Option B.

```
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 10.1.0.1 255.255.255.0
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)# label mode per-nexthop-received-label
Router(config-bgp-af)# exit
Router(config-bgp)# address-family vpnv6 unicast
Router(config-bgp-af)# label mode per-nexthop-received-label
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)# 10.0.0.1/32 Bundle-Ether11 10.1.0.2
Router(config-static-afi)# commit
```

4. Configure ASBR2 for Inter-AS Option B.

```
Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 10.0.0.1 255.255.255.255
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 192.0.2.14 255.240.0.0
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 10.1.0.2 255.255.255.0
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)# label mode per-nexthop-received-label
Router(config-bgp-af)# retain route-target all
Router(config-bgp-af)# exit
Router(config-bgp)# address-family vpnv6 unicast
Router(config-bgp-af)# label mode per-nexthop-received-label
Router(config-bgp-af)# retain route-target all
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 192.0.2.15
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 Bundle-Ether11 10.1.0.1
Router(config-static-afi)# commit
```

5. Configure P2 for Inter-AS Option B.

```
Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 192.0.2.2 255.255.255.255
Router(config-if)# exit
Router(config)# interface Bundle-Ether12
Router(config-if)# description interface towards ASBR2
Router(config-if)# ipv4 address 192.0.2.16 255.240.0.0
Router(config-if)# exit
Router(config)# interface Bundle-Ether25
Router(config-if)# description interface towards PE2
Router(config-if)# ipv4 address ipv4 address 192.0.2.17 255.240.0.0
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
```

6. Configure PE2 for Inter-AS Option B.

```
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 192.0.2.12
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
```

7. Review the running configuration on all devices.

   PE1 running configuration

```
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 192.0.2.12
   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
   !
  !
 !
```

P1 running configuration

```
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 192.0.2.13 255.240.0.0
!
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
 !
!
```

ASBR1 running configuration

```
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 10.1.0.1 255.255.255.0
!
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
  label mode per-nexthop-received-label
  retain route-target all
 !
  address-family vpnv6 unicast
   label mode per-nexthop-received-label
   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
  10.0.0.1/32 Bundle-Ether11 10.1.0.2
 !
!
```

ASBR2 running configuration

```
interface Loopback0
 ipv4 address 10.0.0.1 255.255.255.255
 ipv6 address 1:1:1::1/128
!
interface Bundle-Ether12
 description interface to P2
 ipv4 address 192.0.2.14 255.240.0.0
!
interface Bundle-Ether11
 description interface to ASBR1
 ipv4 address 10.1.0.2 255.255.255.0
!
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
  label mode per-nexthop-received-label
  retain route-target all
 !
 address-family vpnv6 unicast
  label mode per-nexthop-received-label
  retain route-target all
 !
 neighbor 192.0.2.15
  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 Bundle-Ether11 10.1.0.1
 !
```

P2 running configuration

```
interface Loopback0
 ipv4 address 192.0.2.2 255.255.255.255
!
interface Bundle-Ether12
 description interface towards ASBR2
 ipv4 address 192.0.2.16 255.240.0.0
!
interface Bundle-Ether25
 description interface towards PE2
 ipv4 address 192.0.2.17 255.240.0.0
!

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
 !
!
```

PE2 running configuration

```
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 192.0.2.11
   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 192.0.2.18 255.240.0.0
 ipv6 address 105:1:1::1/96
 encapsulation dot1q 1
!

interface Loopback0
 ipv4 address 192.0.2.15 255.255.255.255
 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
```

8. Verify route, label, and forwarding state across the topology.

   Verification on PE1.

   L3VPN route 192.0.2.10/24 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 - IS-IS, 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    192.0.2.19/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:04:43
C     192.0.2.20/24 is directly connected, 01:14:27, TenGigE0/0/0/22/0.1
L     192.0.2.21/32 is directly connected, 01:14:27, TenGigE0/0/0/22/0.1
B    192.0.2.10/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:00:08
B    192.0.2.22/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:00:08
```

   The following output shows that you can reach 192.0.2.10/24 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 100:1 192.0.2.10/24
Sun Jun  6 23:12:12.140 UTC
BGP routing table entry for 192.0.2.10/24, 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 192.0.2.10
Mon Jun  7 02:07:39.841 UTC
192.0.2.10/24, 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 192.0.2.13/32 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 192.0.2.10/24.

```
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         192.0.2.13       141107
--------------------------------------------------------------------------------

Router:PE1# show cef vrf vrf1 192.0.2.10
Mon Jun  7 02:07:39.841 UTC
192.0.2.10/24, 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 192.0.2.13/32 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# 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 192.0.2.10/24 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 100:1 192.0.2.10
Sun Jun  6 19:28:09.018 EDT
BGP routing table entry for 192.0.2.10/24, 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
192.0.2.19/24       10.10.10.10     10.0.0.1         200?
192.0.2.10/24       10.10.10.10     10.0.0.1         200 501i
192.0.2.22/24       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  100:1:192.0.2.10/24              10.1.0.1         0
--------------------------------------------------------------------------------
```

Verification on ASBR2.

The prefix 192.0.2.10/24 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 100:1 192.0.2.10
Sun Jun  6 23:06:32.812 EDT
BGP routing table entry for 192.0.2.10/24, 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
    192.0.2.15 (metric 30) from 192.0.2.15 (192.0.2.15)
      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
192.0.2.19/24       10.0.0.1         192.0.2.15         200?
192.0.2.20/24       10.0.0.1         10.10.10.10     200 100?
192.0.2.10/24       10.0.0.1         192.0.2.15         200 501i
192.0.2.22/24       10.0.0.1         192.0.2.15         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                           192.0.2.15         654
--------------------------------------------------------------------------------
```

Verification on P2.

P2 is along the transit path of the traffic. It label switches or pops 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 192.0.2.15/32
Mon Jun  7 03:09:11.532 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24005  Pop         192.0.2.15/32         BE25         192.0.2.23
11921958
--------------------------------------------------------------------------------
```

Verification on PE2.

L3VPN route 192.0.2.10/24 is learned from eBGP neighbor 192.0.2.11 (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 - IS-IS, 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    192.0.2.19/24 is directly connected, 6w4d, TenGigE0/0/0/30.1
L    192.0.2.18/32 is directly connected, 6w4d, TenGigE0/0/0/30.1
B    192.0.2.20/24 [200/0] via 10.0.0.1 (nexthop in vrf default), 03:58:16
B    192.0.2.24/24 [200/0] via 10.0.0.1 (nexthop in vrf default), 04:30:07
B    192.0.2.10/24 [20/0] via 192.0.2.11, 01:30:05
B    192.0.2.22/24 [20/0] via 192.0.2.11, 01:30:05
```

The route 192.0.2.10/24 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 100:1 192.0.2.10

BGP routing table entry for 192.0.2.10/24, 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
    192.0.2.11 from 192.0.2.11 (192.0.2.25)
      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 192.0.2.12.

```
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 192.0.2.10
Mon Jun  7 03:04:08.268 UTC
192.0.2.10/24, 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 192.0.2.11/32, recursive

   via 192.0.2.11/32, 3 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0x8d575b80 0x0]
    next hop 192.0.2.11/32 via 192.0.2.11/32

    Load distribution: 0 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   TenGigE0/0/0/30.1         192.0.2.11
```

Inter-AS Option B for L3VPN is configured and verified when routing, label, and forwarding tables display correct states and expected label mappings across all PE, P, and ASBR devices.

5 Carrier Supporting Carrier for L3VPN

Topics:

Outlines Carrier Supporting Carrier architecture for L3VPN, detailing core concepts, packet flow processes, key benefits, and complete configuration procedures for implementing carrier transport and connectivity in multi-provider MPLS environments.

Carrier Supporting Carrier for L3VPN

Introduces the Carrier Supporting Carrier concept for L3VPN, explaining foundational principles and illustrating customer-carrier and MPLS service-provider packet flow through end-to-end process descriptions.

A carrier supporting carrier (CSC) model is a service provider architecture that:

- lets a backbone carrier provide a segment of its backbone network to another provider
- uses CSC-CE and CSC-PE routers at the carrier edge, and
- supports both Internet service provider (ISP) and BGP/MPLS VPN service-provider scenarios.

Feature history

The feature history table lists release support for this feature.

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Carrier Supporting Carrier for L3VPN	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: - 8711-48Z-M - 8011-32Y8L2H2FH - 8011-12G12X4Y-A/D
Carrier Supporting Carrier for L3VPN	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Carrier Supporting Carrier for L3VPN	Release 24.4.1	Introduced in this release on: Fixed Systems (8700) (select variants only*) *The Carrier Supporting Carrier for L3VPN functionality is now extended to the Cisco 8712-MOD-M routers.

Feature Name	Release Information	Feature Description
Carrier Supporting Carrier for L3VPN	Release 24.3.1	Introduced in this release on: Fixed Systems (8200, 8700); Modular Systems (8800 [LC ASIC: P100]) (select variants only*) *The Carrier Supporting Carrier for L3VPN functionality is now extended: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Carrier Supporting Carrier for L3VPN	Release 24.2.11	Introduced in this release on: Modular Systems (8800 [LC ASIC: P100]) (select variants only*) *The Carrier Supporting Carrier for L3VPN functionality is now extended to routers with the 88-LC1-36EH line cards.
Carrier Supporting Carrier for L3VPN	Release 7.3.15	This feature enables MPLS VPN-based backbone carriers to allow customer carriers to use a segment of the backbone network. The backbone carrier can accommodate many customer carriers and provide access to the backbone. Customer carriers no longer have to bear the burden of configuring, operating, and maintaining their own backbone.

CSC terminology

- Backbone carrier: Service provider that provides a segment of the backbone network to another provider, offering BGP and MPLS VPN services.
- Customer carrier: Service provider that uses the provided backbone segment. This may be an Internet service provider (ISP) or a BGP/MPLS VPN service provider.
- CSC-CE router: A customer edge router that is part of a customer network and connects to a provider edge (CSC-PE) router. The CSC-CE sits on the edge of the customer carrier network.
- CSC-PE router: A provider edge router that is part of the service provider's network connected to a CSC-CE router. The CSC-PE sits on the edge of the backbone carrier network.
- Types of customer carriers:
  - Internet service provider (ISP)
  - BGP/MPLS VPN service provider

How packet flow works when the customer carrier is an ISP

Describes the ISP customer-carrier topology, eBGP and iBGP route distribution, and single-label packet flow into CSC-CE.

The following topology shows a network configuration where the customer carrier is an ISP. The customer carrier has two sites, each of which is a point of presence (POP). The customer carrier connects these sites using a VPN service provided by the backbone carrier. The backbone carrier uses MPLS to provide VPN services. The ISP sites use MPLS.

The key components involved in the process are:

- Customer Edge (CE) routers: Connect the ISP's local sites (POPs) to the backbone carrier and exchange routes via eBGP.
- Provider Edge (PE) routers: Interface between customer sites and backbone, distributing VPN routes and labels using multiprotocol iBGP.
- Backbone carrier network: Uses MPLS to provide VPN services connecting ISP POPs.
- MPLS protocol: Enables label-based forwarding across backbone and ISP sites.
- eBGP and iBGP protocols: Distribute IPv4 routes, MPLS labels, and VPNv4 routes between CE and PE routers.

A customer carrier ISP topology involves two sites (each a POP), connected via a VPN service provided by a backbone carrier using MPLS. The ISP sites also use MPLS. Packet flow and route distribution leverage eBGP and iBGP protocols.

Figure 7: Customer Carrier Is an ISP

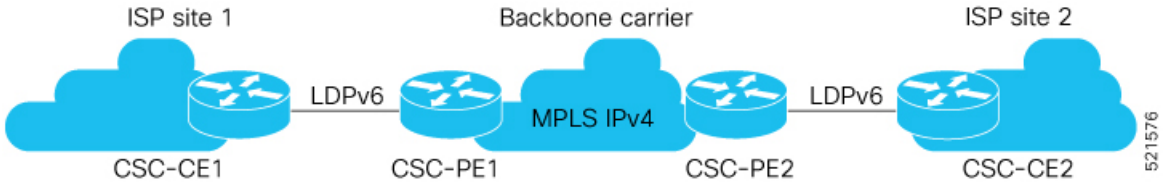
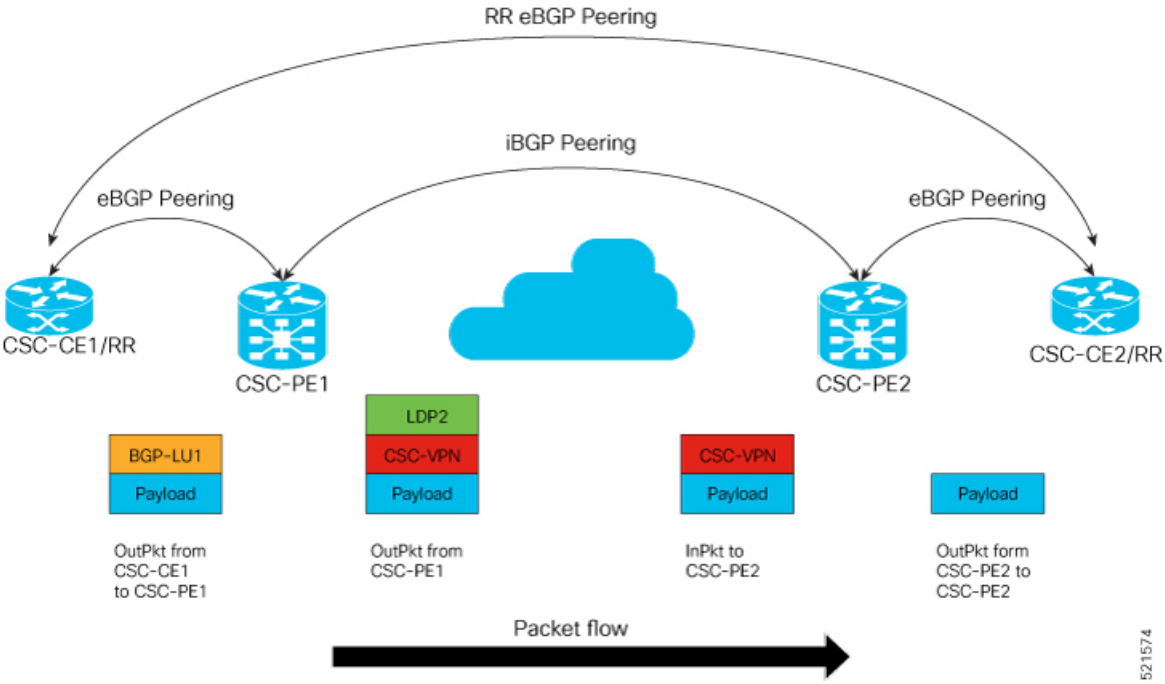


Figure 8: MPLS Single-Label Packet Flow into CSC-CE (CSC-CE/RR eBGP Peering)



These stages describe how packet flow works when the customer carrier is an ISP.

1. Topology establishment: The ISP customer carrier sets up two points of presence (POPs), connecting these via a VPN service from the backbone carrier.

2. MPLS VPN configuration: Both ISP sites and the backbone carrier enable MPLS for efficient VPN service delivery.
3. Route and label distribution: CE and PE routers use eBGP to exchange IPv4 routes and MPLS labels. PE routers internally use multiprotocol iBGP to distribute VPNv4 routes across the backbone.
4. Packet forwarding into CSC-CE: When traffic flows into the customer carrier's CE device (CSC-CE), it follows a single-label MPLS path based on the route information distributed by eBGP and iBGP.
5. Traffic confirmation: The process concludes as the topology delivers traffic into CSC-CE, using the distributed labels and routes to achieve efficient packet transfer.

The ISP customer-carrier scenario uses the described MPLS topology and routing protocols to forward packets efficiently into CSC-CE, ensuring reliable connectivity between ISP POPs via the backbone VPN.

How packet flow works when the customer carrier is an MPLS service provider

Describes how customer and backbone carriers use MPLS VPN topology, BGP-LU, LDP, eBGP route redistribution, and two-label packet flow into the CSC-CE device.

In an MPLS VPN environment, a customer carrier may have sites in multiple locations and uses MPLS to carry VPN traffic among those sites. The backbone service provider connects these sites and may use either MPLS or IP tunnels within its own core. Packet flow between the two carriers requires both routing coordination via eBGP and label distribution protocols like BGP-LU and LDP.

The key components involved in the process are:

- Backbone carrier: Provides core network infrastructure for transporting MPLS or IP-tunneled traffic and interconnects with customer carriers at the service-provider edge.
- Customer carrier: Operates its own MPLS network, connects multiple customer sites, and peers with the backbone carrier via BGP/MPLS VPN.
- CSC-CE router: The customer carrier’s edge router that exchanges routing information with the backbone carrier and handles redistributed eBGP routes using IGP.
- CSC-PE router: The backbone carrier’s edge router that peers with the customer carrier’s CSC-CE via eBGP.

MPLS service-provider packet flows enable two independent carriers (customer and backbone) to interconnect via BGP/MPLS VPN topology, efficiently routing packets between customer sites across different service-provider domains.

Figure 9: Customer Carrier is an MPLS VPN Service Provider

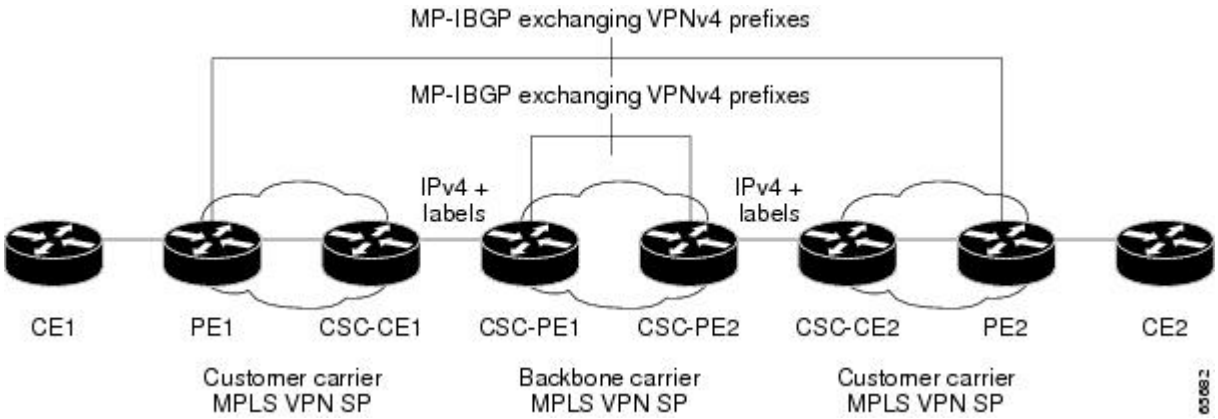
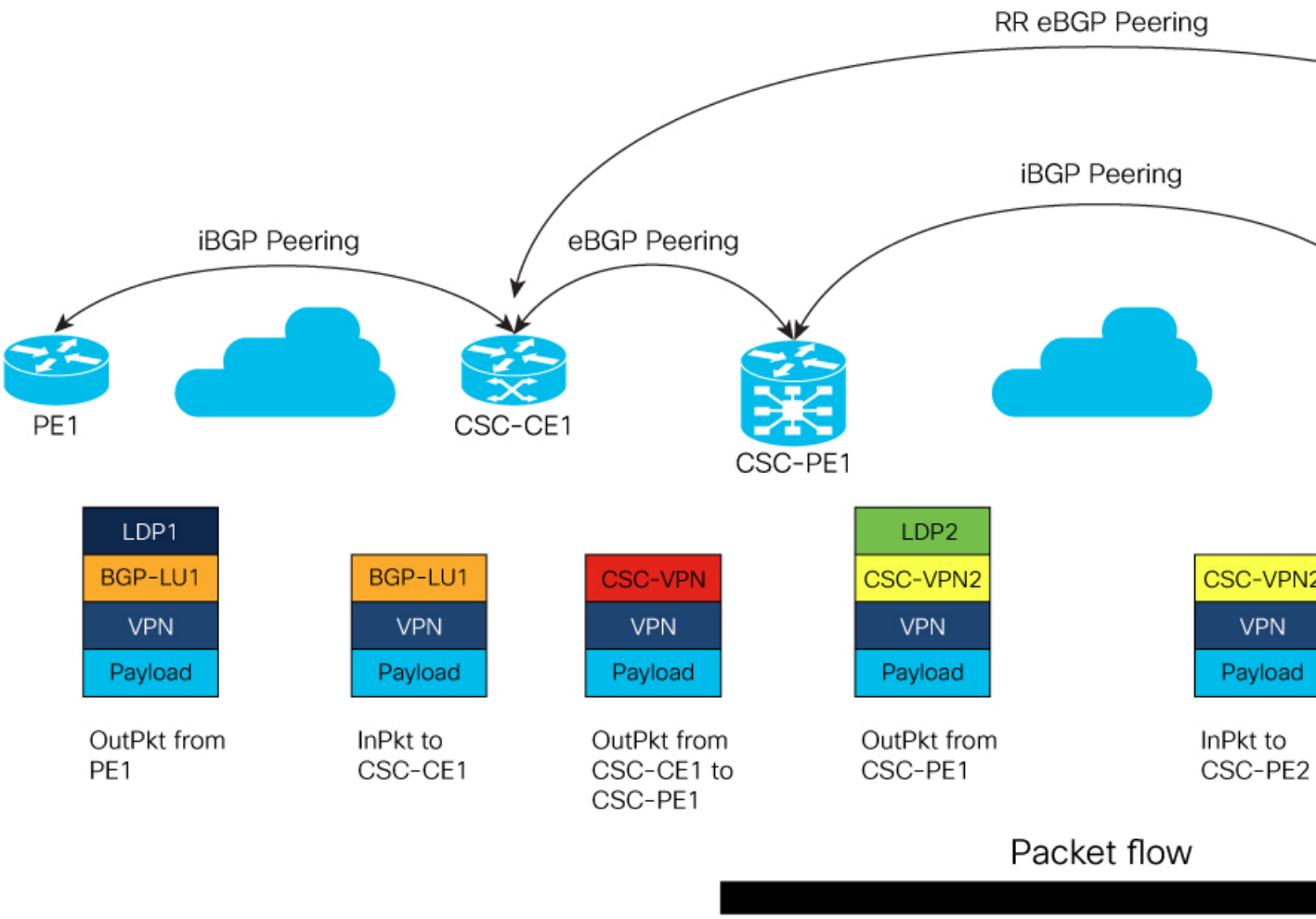


Figure 10: MPLS Two-Labels Packet Flow into CSC-CE (CSC-CE/RR BGP Peering)



These stages describe how packet flow works when the customer carrier is an MPLS service provider.

1. Service-provider topology establishment: The backbone and customer carriers establish a BGP/MPLS VPN interconnection, with CSC-CE and CSC-PE routers at their respective network edges.
2. Label distribution protocol setup: The customer carrier configures BGP-LU (label unicast) and LDP (Label Distribution Protocol) in its core to support MPLS label switching for VPN traffic.
3. Route exchange and redistribution: The CSC-CE router receives eBGP-learned routes from the backbone carrier's CSC-PE router and redistributes them into the customer carrier's IGP to ensure consistent routing within its domain.
4. Packet forwarding with two-label stack: When packets reach the CSC-CE, they carry a two-label MPLS stack: the top label represents the backbone (transport) path, and the second label identifies the end VPN destination in the customer carrier network.
5. Final delivery to customer site: The CSC-CE removes the appropriate labels and forwards the packet to the target customer site over the MPLS-enabled customer carrier network.

The MPLS service-provider customer-carrier solution allows secure, scalable, and efficient end-to-end traffic forwarding between customer sites across multiple provider domains using label-based routing and VPN technologies.

Benefits of Carrier Supporting Carrier for backbone and customer carriers

Describes key benefits of Carrier Supporting Carrier, highlighting advantages in scalability, operational efficiency, and seamless integration for multi-provider L3VPN deployments.

Carrier Supporting Carrier (CSC) offers several advantages for different stakeholders in MPLS VPN environments:

- Benefits to the backbone carrier:
  - The MPLS VPN CSC feature is scalable, supporting growth in network size and complexity.
  - It is a flexible solution, accommodating various network topologies and requirements.
- Benefits to the customer carriers:
  - Customer carriers receive the same level of security as legacy Frame Relay or ATM-based VPNs.
  - Any link layer technology may be used to connect CE routers to PE routers.
  - Customer carriers have full flexibility in addressing schemes and are still supported by the backbone carrier.
- Benefits of implementing MPLS VPN CSC using BGP:
  - BGP replaces the need for an IGP and LDP in the VPN forwarding and routing instance (VRF) table.
  - BGP is the preferred routing protocol for connecting two Internet service providers (ISPs), improving interoperability and scalability.

By leveraging CSC, service providers and their customers gain enhanced scalability, flexibility, and secure, efficient route and label distribution across interconnected networks.

Configure Carrier Supporting Carrier for L3VPN on CSC-PE

Provides configuration procedures for enabling Carrier Supporting Carrier in L3VPN environments, offering step-by-step guidance to set up, validate, and optimize carrier-based MPLS service connectivity.

Enable Carrier Supporting Carrier for L3VPN on the CSC Provider Edge (CSC-PE) router.

Perform this task on the CSC-PE to configure advanced L3VPN label allocation and BGP settings required for Carrier Supporting Carrier.

- Plan route policy names and definitions for label allocation.
- Identify required BGP neighbor-groups, VRF configuration, route distinguishers, label allocation strategies, and peer addresses.

Follow these steps to configure and verify Carrier Supporting Carrier for L3VPN on CSC-PE:

1. Configure Carrier Supporting Carrier for L3VPN on CSC-PE.

```
Router# configure
Router(config)# route-policy LABEL_ALLOC
Router(config-rpl)# if destination in CSC-Prefix then
Router(config-rpl-if)# set label-mode per-prefix
Router(config-rpl-if)# else
Router(config-rpl-else)# set label-mode per-vrf
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
Route(config)# router bgp 100
Router(config-bgp)# neighbor-group ebgplu
```

```
Router(config-bgp-nbrgrp)# remote-as 200
Router(config-bgp-nbrgrp)# address-family ipv4 labeled-unicast
Router(config-bgp-nbrgrp-af)# multipath
Router(config-bgp-nbrgrp-af)# route-policy pass in
Router(config-bgp-nbrgrp-af)# route-policy deny-l3vpn out
Router(config-bgp-nbrgrp-af)# as-override
Router(config-bgp-nbrgrp-af)# next-hop-self
Router(config-bgp-nbrgrp-af)# site-of-origin 100:1
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#exit
Router(config-bgp)#exit
Router(config)# router bgp 100
Router(config-bgp)# vrf vpn1
Router(config-bgp-vrf)# rd 1:1
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# label mode route-policy LABEL_ALLOC
Router(config-bgp-vrf-af)# maximum-paths ebgp 16
Router(config-bgp-vrf-af)# maximum-paths ibgp 16 unequal-cost
Router(config-bgp-vrf-af)# allocate-label route-policy deny-l3vpn
Router(config-bgp-vrf-af)#!
Router(config-bgp-vrf-af)#172.16.0.1
Router(config-bgp-vrf-nbr)# use neighbor-group ebgplu
Router(config-bgp-vrf-nbr)# commit
```

2. Review the Carrier Supporting Carrier running configuration.

```
route-policy LABEL_ALLOC
  if destination in CSC-Prefix then
    set label-mode per-prefix
  else
    set label-mode per-vrf
  endif
end-policy
!
router bgp 100
neighbor-group ebgplu
  remote-as 200
address-family ipv4 labeled-unicast
   multipath
   route-policy pass in
   route-policy deny-l3vpn out
   as-override
   next-hop-self
   site-of-origin 100:1
!
router bgp 100
vrf vpn1
  rd 1:1
  address-family ipv4 unicast
   label mode route-policy LABEL_ALLOC
   maximum-paths ebgp 16
   maximum-paths ibgp 16 unequal-cost
   allocate-label route-policy deny-l3vpn
!
neighbor 172.16.0.1
   use neighbor-group ebgplu
```

This section provides the Carrier Supporting Carrier running configuration.

3. Verify the Carrier Supporting Carrier configuration.

```
Router:CSC-PE1# show cef vrf vpn1 10.0.0.1 detail
10.0.0.1, version 24, internal 0x1000001 0x30 (ptr 0xaf408058) [1], 0x0 (0x0),
 0x208 (0xaebf14e8)
 Updated Nov  6 14:56:14.554
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
  gateway array (0xae8934e0) reference count 3, flags 0x2078, source rib (7),
 0 backups
                [1 type 5 flags 0x48441 (0xd2a2b1b8) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Nov  6 14:56:14.554
 LDI Update time Nov  6 14:56:14.554
   via 192.0.2.10/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags
 0x60a0]
    path-idx 0 NHID 0x0 [0xa09e9fa8 0x0]
    recursion-via-/32
    next hop 192.0.2.10/32 via 100516/0/21
     local label 100927
     next hop 192.0.2.10/32 Hu0/0/0/5.1  labels imposed {ImplNull 200013}
   via 192.0.2.11/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags
 0x60a0]
    path-idx 1 NHID 0x0 [0xa09ea1b8 0x0]
    recursion-via-/32
    next hop 192.0.2.11/32 via 100520/0/21
     local label 100927
     next hop 192.0.2.11/32 BE12.1       labels imposed {ImplNull 200013}
    Load distribution: 0 1 (refcount 1)
    Hash  OK  Interface                 Address
    0     Y   recursive                 100516/0
    1     Y   recursive                 100520/0
-------------------------------------------------------------------------------
Router:CSC-CE2# show cef 10.0.0.1 detail
10.0.0.1 , version 47069, internal 0x1000001 0x30 (ptr 0x89638750) [1], 0x0
(0x8a815198), 0xa28 (0xa9434690)
 Updated Nov  6 10:35:47.662
 local adjacency to HundredGigE0/0/0/1.1

 Prefix Len 32, traffic index 0, precedence n/a, priority 3
  gateway array (0x8a643108) reference count 3, flags 0x68, source lsd (5),
1 backups
                [2 type 5 flags 0x8401 (0xa9479a48) ext 0x0 (0x0)]
  LW-LDI[type=5, refc=3, ptr=0x8a815198, sh-ldi=0xa9479a48]
  gateway array update type-time 1 Nov  6 10:35:47.662
 LDI Update time Nov  6 10:35:47.662
 LW-LDI-TS Nov  6 10:35:47.662
   via 192.0.2.12/32, HundredGigE0/0/0/1.1, 3 dependencies, weight 0, class
0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8ac12a48 0x0]
    next hop 192.0.2.12/32
    local adjacency
     local label 200013      labels imposed {ExpNullv4}

    Load distribution: 0 (refcount 2)

    Hash  OK  Interface                 Address
    0     Y   HundredGigE0/0/0/1.1      192.0.2.12
```

Carrier Supporting Carrier for L3VPN is configured and verified when the CEF output shows the expected recursive paths, local labels, and imposed labels.

6 Layer-3 Route Synchronization for EVPN Multihoming

Topics:

Outlines Layer-3 route synchronization concepts, operational requirements, benefits, and configuration procedures for EVPN multihoming, enabling resilient and efficient routing in multi-home network environments.

Layer 3 route synchronization for EVPN multihoming

Introduces Layer 3 route synchronization for EVPN multihoming, highlighting its benefits, detailing VRF configuration requirements on subinterfaces, and explaining operational workflows for effective route synchronization.

Layer 3 route synchronization for EVPN multihoming is a Layer 3 service function that

- enhances resilience and load balancing for Layer 3 services in multihoming environments
- keeps routing information consistent across redundant PE routers
- synchronizes ARP, ND, and multicast route information using BGP-EVPN to enable seamless traffic forwarding and consistent multicast behavior.

Feature history

The feature history table lists release support for this feature.

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Layer-3 route synchronization for EVPN multi-homing	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Layer-3 route synchronization for EVPN multi-homing	Release 25.3.1	Introduced in this release on: Fixed Systems(8200, 8700, 8011)(select variants only*); Modular Systems (8800 [LC ASIC: P100]) With Layer 3 route synchronization, you can ensure seamless failover and optimal traffic distribution in multihoming environments by synchronizing critical Layer 3 routing information, such as ARP/ND entries and multicast routes, across redundant Provider Edge (PE) routers. This feature leverages the BGP-EVPN route synchronization mechanism to maintain consistent routing states and accelerate convergence across your network. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 8011-4G24Y4H-I

Route synchronization behavior

Layer 3 route synchronization is a functionality that

- enhances the resilience and load-balancing capabilities of Layer 3 services in multihoming environments,
- addresses challenges of maintaining consistent routing information across redundant PE routers, and
- ensures synchronized Layer 3 routing information (ARP/ND, multicast routes) on backup or standby PEs.

In multihoming setups, traffic can arrive at any redundant PE. However, Layer 3 routing information, such as ARP/ND entries for hosts and multicast routes for PIM Designated Routers (DRs), might only be learned by the primary or active PE. This can lead to traffic drops on backup or standby PEs until these entries are resolved. Additionally, PIM DR election might not always align with the currently active service PE, causing inconsistencies in multicast routing.

Layer 3 route synchronization eliminates these issues by proactively synchronizing ARP/ND entries and multicast routes across all redundant PEs using BGP-EVPN. This capability ensures that all PEs possess the necessary routing information, enabling seamless traffic forwarding and consistent multicast behavior, even on backup or standby devices.

Layer 3 route synchronization benefits

Lists the resilience, load-balancing, routing consistency, and convergence benefits of Layer 3 route synchronization for EVPN multihoming deployments.

Layer 3 route synchronization offers several benefits for redundant PE designs that use EVPN multihoming for Layer 3 services:

- Enhanced resilience: Ensures backup or standby PEs have synchronized Layer 3 routing information (such as ARP/ND and multicast routes), enabling seamless failover and minimizing traffic loss during link or node failures.
- Optimal load balancing: Facilitates efficient traffic distribution by ensuring all eligible PEs are prepared to forward Layer 3 traffic without delays caused by missing routing entries.
- Consistent routing: Leverages BGP-EVPN for control-plane synchronization of Layer 3 routes, providing a unified and consistent view of network reachability across redundant PEs.
- Faster convergence: Reduces convergence time by pre-populating ARP/ND tables and synchronizing multicast states, eliminating the need for on-demand learning after a failover.

Configure VRF only on subinterfaces in EVPN multihoming deployments

Ensure that Layer 3 route synchronization for EVPN multihoming is supported only when configuring VRF on subinterfaces.

For EVPN multihoming deployments, configure the VRF only on subinterfaces to enable Layer 3 route synchronization. Configuring the VRF on interfaces other than subinterfaces is not supported.

How Layer 3 route synchronization works

Layer 3 route synchronization is critical in EVPN networks to ensure high availability and prevent service interruptions. By coordinating routing information across multiple Provider Edge routers, the network maintains seamless traffic flow, even during failovers or topology changes.

Layer 3 route synchronization is critical in EVPN networks to ensure high availability and prevent service interruptions. By coordinating routing information across multiple Provider Edge routers, the network maintains seamless traffic flow, even during failovers or topology changes.

The key components involved in the process are:

- BGP-EVPN: The underlying protocol for route synchronization.
- Service PEs: The Provider Edge routers that participate in route learning and synchronization.
- Route types: Specific EVPN route types (MAC-IP route type, IP Prefixes route type, Multicast Report route types) used for different types of Layer 3 information.
- Synchronization attributes: EVPN attributes (EVI, L2-Route-Target, ESI, ETAG, ES Import, EVI-Route-Target) that tag routes for targeted distribution.

- Connected switches: Devices that update their MAC tables during failover events.

Layer 3 route synchronization ensures consistent routing information across redundant service Provider Edge (PE) routers by leveraging BGP-EVPN to propagate and install Layer 3 routing information.

These stages describe how Layer 3 route synchronization works:

1. Information learning and propagation: A service PE learns Layer 3 routing information (such as ARP/ND entries and multicast routes) and propagates it to other redundant service PEs by using BGP-EVPN.
2. Targeted route distribution: BGP imports Layer 3 synchronization routes based on matching ES Import route-policy filters and EVI route targets, ensuring distribution to relevant PEs.
3. Scenario application: The synchronization mechanism applies to Layer 3 gateway multihoming, L3VPN active-active bundles, and L3VPN port-active bundles, providing resilient services and efficient traffic handling.
4. If a PE fails or receives a route delete, the new primary service PE triggers a gratuitous ARP (GARP) or Neighbor Advertisement (NA) replay. Connected switches update their MAC tables, redirecting traffic to the new primary PE.

Layer 3 route synchronization keeps redundant service PEs ready to forward traffic and respond to failover events with consistent routing state.

Configure Layer 3 route synchronization with port-active redundancy

Provides instructions to configure Layer 3 route synchronization with port-active redundancy, enabling reliable failover and efficient routing in EVPN multihoming scenarios.

Enable consistent Layer 3 route synchronization for VRF instances across redundant PE routers in EVPN fabric with port-active multihoming.

Port-active mode ensures that only a single PE forwards traffic for a given Ethernet segment at any time, improving network redundancy and reliability.

- Ensure you have access to the command-line interface (CLI) of the PE router.
- Identify the VRF instances for which you want to enable Layer 3 route synchronization.
- Determine the interfaces that will participate in port-active multihoming.

1. Configure the BGP route-targets for VRFs to control IPv4/IPv6 route distribution and enable synchronization of Layer 3 routing information with the Layer-2 EVPN fabric.

```
Router# configure
Router(config)#vrf PA1
Router(config-vrf)#address-family ipv4 Unicast
Router(config-vrf-af)#import route-target 64600:10000
Router(config-vrf-af)#export route-target 64600:10000
Router(config-vrf-af)#evpn-route-sync 10000
Router(config-vrf)#vrf PA1
Router(config-vrf)#address-family ipv6 Unicast
Router(config-vrf-af)#import route-target 64600:10000
Router(config-vrf-af)#export route-target 64600:10000
Router(config-vrf)#vrf PA2
Router(config-vrf)#address-family ipv4 Unicast
Router(config-vrf-af)#import route-target 64600:10001
Router(config-vrf-af)#export route-target 64600:10001
Router(config-vrf-af)#evpn-route-sync 10001
Router(config-vrf)#vrf PA2
Router(config-vrf)#address-family ipv6 Unicast
Router(config-vrf-af)#import route-target 64600:10001
Router(config-vrf-af)#export route-target 64600:10001
```

```
Router(config-vrf-af)#commit
Router(config-vrf-af)#exit
Router(config-vrf)#exit
```

2. Configure VRF, encapsulation, and L3 addresses for the relevant subinterfaces.

```
Router(config)#interface Bundle-Ether200.2307
Router(config-subif)#vrf PA1
Router(config-subif)#ipv4 address 172.16.1.1/24
Router(config-subif)#ipv6 address 155:1:1:7::1/64
Router(config-subif)#encapsulation dot1q 2307
```

3. Configure Ethernet segment for port-active redundancy.

```
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether200
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 11.11.11.11.11.11.11.11.11
Router(config-evpn-ac-es)# load-balancing-mode port-active
Router(config-evpn-ac-es)# root
Router(config)# commit
```

4. Use the **show arp vrf PA1 detail** command to verify the ARP entries and their synchronization status for VRF PA1.

```
Router#show arp vrf PA1 detail

------------------------------------------------------------------------------
0/RP0/CPU0
------------------------------------------------------------------------------
Address         Age        Hardware Addr   State      Flag       Type
Interface
192.0.2.11       -          0000.0000.0200  Interface  Unknown    ARPA
Bundle-Ether200.2301
192.0.2.12      -          0010.9400.2f8e  EVPN_SYNC  EVPN-SYNC  ARPA
Bundle-Ether200.2301
192.0.2.13       -          0000.0000.0200  Interface  Unknown    ARPA
Bundle-Ether200.2303
192.0.2.14      00:00:36   0010.9400.2f90  Dynamic    Dynamic    ARPA
Bundle-Ether200.2303
192.0.2.15       -          0000.0000.0200  Interface  Unknown    ARPA
Bundle-Ether200.2305
192.0.2.16      00:00:36   0010.9400.2f92  Dynamic    Dynamic    ARPA
Bundle-Ether200.2305
192.0.2.17       -          0000.0000.0200  Interface  Unknown    ARPA
Bundle-Ether200.2307
192.0.2.18      -          0010.9400.2f94  EVPN_SYNC  EVPN-SYNC  ARPA
Bundle-Ether200.2307
```

5. Verify route synchronization status for the bridge-domain.

```
Router# show l2vpn bridge-domain bd-name rs#10000 detail

Legend: pp = Partially Programmed.
Bridge group: #ROUTE-SYNC, bridge-domain: rs#10000, id: 140, state: up, ShgId:
 0, MSTi: 0
  Coupled state: disabled
  VINE state: EVPN Native
  MAC learning: enabled
  MAC withdraw: enabled
```

```
    MAC withdraw for Access PW: enabled
    MAC withdraw sent on: bridge port up
    MAC withdraw relaying (access to access): disabled
  Flooding:
    Broadcast & Multicast: enabled
    Unknown unicast: enabled
  MAC aging time: 300 s, Type: inactivity
  MAC limit: 524287, Action: none, Notification: syslog
  MAC limit reached: no, threshold: 75%
  MAC port down flush: enabled
  MAC Secure: disabled, Logging: disabled
  Split Horizon Group: none
  E-Tree: Root
  Dynamic ARP Inspection: disabled, Logging: disabled
  IP Source Guard: disabled, Logging: disabled
  DHCPv4 Snooping: disabled
  DHCPv4 Snooping profile: none
  IGMP Snooping: disabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  Storm Control: disabled
  Bridge MTU: 1500
  MIB cvplsConfigIndex: 141
  Filter MAC addresses:
  P2MP PW: disabled
  Multicast Source: Not Set
  Create time: 11/06/2025 03:28:06 (00:05:13 ago)
  No status change since creation
  ACs: 0 (0 up), VFIs: 0, PWs: 0 (0 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
  List of EVPNs:
    EVPN, state: unresolved
      evi: 0 (N/A)
      XC ID 0x0
  List of ACs:
  List of Access PWs:
  List of VFIs:
  List of Access VFIs:
```

6. Verify EVPN EVI details, including its association with VRF for route synchronization.

```
Router# show evpn evi vpn-id 10001 detail

VPN-ID     Encap      Bridge Domain                 Type
---------- ---------- ----------------------------- -------------------
10001      ROUTE-SYNC rs#10001                      EVPN
   Stitching: Regular
   E-Tree: Root
   Forward-class: 0
   Advertise MACs: No
   Advertise BVI MACs: No
   Aliasing: Enabled
   UUF: Enabled
   Re-origination: Enabled
   Multicast:
     IGMP-Snooping Proxy: No
     MLD-Snooping Proxy : No
   BGP Implicit Import: Enabled
   VRF Name: PA2
   Preferred Nexthop Mode: Off
   BVI Coupled Mode: No
   BVI Subnet Withheld: ipv4 No, ipv6 No
   L3VRF Label Mode: Per-VRF
```

```
   RD Config: none
   RD Auto  : (auto) 192.168.10.1:10001
   RT Auto  : 64600:10001
   Route Targets in Use           Type
   ------------------------------ --------------------
   64600:10001                    Import
   64600:10001                    Export
```

7. Use the **show evpn ethernet-segment interface BE200** command to check the load-balancing mode.

```
Router# show evpn ethernet-segment interface BE200 carving detail
..

Ethernet Segment Id      Interface                           Originating IP
------------------------ ----------------------------------
--------------------
0011.1111.1111.1111.1111 BE200                               192.168.10.1
                                                             192.168.20.1
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether200
     Interface MAC  : 0000.0000.0200
     IfHandle       : 0x78000234
     State          : Standby
     Redundancy     : Not Defined
  ESI ID            : 1
  ESI type          : 0
     Value          : 0011.1111.1111.1111.1111
  ES Import RT      : 1111.1111.1111 (Local)
  Topology          :
     Operational    : MH
     Configured     : Port-Active
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       : Reroute
  Peering Details   : 2 Nexthops
     192.168.10.1 [MOD:P:00:T]
     192.168.20.1 [MOD:P:00:T]
```

Layer 3 route synchronization is successfully configured and verified, ensuring consistent routing information and L2VPN connectivity across the EVPN fabric for VRFs PA1 and PA2.

Configure Layer 3 route synchronization for EVPN multihoming with all-active redundancy

Details the process for configuring Layer 3 route synchronization with all-active redundancy, supporting load balancing and seamless routing across multiple active links in EVPN multihomed environments.

Enable and verify Layer 3 route synchronization for VRF instances across redundant PE routers using the all-active EVPN multihoming model.

Layer 3 route synchronization for VRF instances across redundant PE routers with all-active multihoming allows multiple PEs to forward traffic for a given Ethernet segment simultaneously, ensuring efficient load balancing and L2/L3 consistency in the EVPN fabric.

- Ensure you have access to the CLI of the PE router.
- Identify the VRF instances for which you want to enable Layer 3 route synchronization.

- Determine the interfaces that will participate in all-active multihoming.

1. Configure the BGP route targets for VRFs.

   Control IPv4/IPv6 route distribution and enable synchronization of Layer 3 routing information with the Layer-2 EVPN fabric.

```
Router# configure
Router(config)#vrf PA1
Router(config-vrf)#address-family ipv4 Unicast
Router(config-vrf-af)#import route-target 64600:10000
Router(config-vrf-af)#export route-target 64600:10000
Router(config-vrf-af)#evpn-route-sync 10000
Router(config-vrf)#vrf PA1
Router(config-vrf)#address-family ipv6 Unicast
Router(config-vrf-af)#import route-target 64600:10000
Router(config-vrf-af)#export route-target 64600:10000
Router(config-vrf)#vrf PA2
Router(config-vrf)#address-family ipv4 Unicast
Router(config-vrf-af)#import route-target 64600:10001
Router(config-vrf-af)#export route-target 64600:10001
Router(config-vrf-af)#evpn-route-sync 10001
Router(config-vrf)#vrf PA2
Router(config-vrf)#address-family ipv6 Unicast
Router(config-vrf-af)#import route-target 64600:10001
Router(config-vrf-af)#export route-target 64600:10001
Router(config-vrf-af)#commit
Router(config-vrf-af)#exit
Router(config-vrf)#exit
```

2. Configure VRF, encapsulation, and Layer 3 addresses for subinterfaces.

```
Router(config)#interface Bundle-Ether200.2307
Router(config-subif)#vrf PA1
Router(config-subif)#ipv4 address 172.16.1.1/24
Router(config-subif)#ipv6 address 155:1:1:7::1/64
Router(config-subif)#encapsulation dot1q 2307
```

3. Configure Ethernet segment for all-active redundancy.

```
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether200
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 11.11.11.11.11.11.11.11.11
Router(config-evpn-ac-es)# root
Router(config)# commit
```

   All-active is the default mode.

   Repeat the configuration for other relevant interfaces.

4. Use the **show arp vrf PA1 detail** command to verify the ARP entries and their synchronization status for VRF PA1.

```
Router#show arp vrf PA1 detail

------------------------------------------------------------------------------
0/RP0/CPU0
------------------------------------------------------------------------------
Address         Age        Hardware Addr   State      Flag       Type
Interface
```

```
192.0.2.11       -          0000.0000.0200  Interface  Unknown    ARPA
Bundle-Ether200.2301
192.0.2.12       -          0010.9400.2f8e  EVPN_SYNC  EVPN-SYNC  ARPA
Bundle-Ether200.2301
192.0.2.13       -          0000.0000.0200  Interface  Unknown    ARPA
Bundle-Ether200.2303
192.0.2.14       00:00:36   0010.9400.2f90  Dynamic    Dynamic    ARPA
Bundle-Ether200.2303
192.0.2.15       -          0000.0000.0200  Interface  Unknown    ARPA
Bundle-Ether200.2305
192.0.2.16       00:00:36   0010.9400.2f92  Dynamic    Dynamic    ARPA
Bundle-Ether200.2305
192.0.2.17       -          0000.0000.0200  Interface  Unknown    ARPA
Bundle-Ether200.2307
192.0.2.18       -          0010.9400.2f94  EVPN_SYNC  EVPN-SYNC  ARPA
Bundle-Ether200.2307
```

5. Verify route synchronization status for the bridge-domain.

```
Router# show l2vpn bridge-domain bd-name rs#10000 detail

Legend: pp = Partially Programmed.
Bridge group: #ROUTE-SYNC, bridge-domain: rs#10000, id: 140, state: up, ShgId:
 0, MSTi: 0
  Coupled state: disabled
  VINE state: EVPN Native
  MAC learning: enabled
  MAC withdraw: enabled
    MAC withdraw for Access PW: enabled
    MAC withdraw sent on: bridge port up
    MAC withdraw relaying (access to access): disabled
  Flooding:
    Broadcast & Multicast: enabled
    Unknown unicast: enabled
  MAC aging time: 300 s, Type: inactivity
  MAC limit: 524287, Action: none, Notification: syslog
  MAC limit reached: no, threshold: 75%
  MAC port down flush: enabled
  MAC Secure: disabled, Logging: disabled
  Split Horizon Group: none
  E-Tree: Root
  Dynamic ARP Inspection: disabled, Logging: disabled
  IP Source Guard: disabled, Logging: disabled
  DHCPv4 Snooping: disabled
  DHCPv4 Snooping profile: none
  IGMP Snooping: disabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  Storm Control: disabled
  Bridge MTU: 1500
  MIB cvplsConfigIndex: 141
  Filter MAC addresses:
  P2MP PW: disabled
  Multicast Source: Not Set
  Create time: 11/06/2025 03:28:06 (00:05:13 ago)
  No status change since creation
  ACs: 0 (0 up), VFIs: 0, PWs: 0 (0 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
  List of EVPNs:
    EVPN, state: unresolved
      evi: 0 (N/A)
      XC ID 0x0
  List of ACs:
```

```
  List of Access PWs:
  List of VFIs:
  List of Access VFIs:
```

6. Verify EVPN EVI details, including its association with VRF for route synchronization.

```
Router# show evpn evi vpn-id 10001 detail

VPN-ID     Encap      Bridge Domain                 Type
---------- ---------- ----------------------------- --------------------
10001      ROUTE-SYNC rs#10001                      EVPN
   Stitching: Regular
   E-Tree: Root
   Forward-class: 0
   Advertise MACs: No
   Advertise BVI MACs: No
   Aliasing: Enabled
   UUF: Enabled
   Re-origination: Enabled
   Multicast:
     IGMP-Snooping Proxy: No
     MLD-Snooping Proxy : No
   BGP Implicit Import: Enabled
   VRF Name: PA2
   Preferred Nexthop Mode: Off
   BVI Coupled Mode: No
   BVI Subnet Withheld: ipv4 No, ipv6 No
   L3VRF Label Mode: Per-VRF
   RD Config: none
   RD Auto  : (auto) 192.168.10.1:10001
   RT Auto  : 64600:10001
   Route Targets in Use           Type
   ------------------------------ ---------------------
   64600:10001                    Import
   64600:10001                    Export
```

7. Use the **show evpn ethernet-segment interface BE200** command to check the load-balancing mode.

```
Router#show evpn ethernet-segment interface BE200 carving detail
..
Ethernet Segment Id      Interface                           Originating IP
-----------------------  ----------------------------------
--------------------
0011.1111.1111.1111.1111 BE200                               192.168.10.1
                                                             192.168.20.1
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether200
     Interface MAC  : 0000.0000.0200
     IfHandle       : 0x78000234
     State          : Up
     Redundancy     : Not Defined
  ESI ID            : 1
  ESI type          : 0
     Value          : 0011.1111.1111.1111.1111
  ES Import RT      : 1111.1111.1111 (Local)
  Topology          :
     Operational    : MH, All-active
     Configured     : All-active (AApF) (default)
  Service Carving   : Auto-selection
```

```
    Multicast      : Disabled
Convergence        : Reroute
Peering Details    : 2 Nexthops
    192.168.10.1 [MOD:P:00:T]
    192.168.20.1 [MOD:P:00:T]
```

Layer 3 route synchronization is successfully configured and verified. You can confirm consistent routing and L2VPN connectivity across the EVPN fabric for VRFs PA1 and PA2 by reviewing ARP entries, bridge-domain status, and Ethernet segment settings.

7 VXLAN Static Routing

Topics:

Outlines VXLAN static routing concepts, benefits, restrictions, topology operations, and configuration with and without Service Layer API, providing foundational knowledge and procedural guidance for deploying VXLAN static routing in network environments.

VXLAN static route services

Introduces VXLAN static route services, covering overlay networks, benefits of VXLAN and VXLAN static routing, static routing paths, restrictions, topology operations, and integration using Service Layer API to deliver comprehensive understanding of VXLAN static routing solutions.

A VXLAN static route service is a VXLAN routing method that

- defines the path from a source Virtual Tunnel Endpoint (VTEP) to a destination VTEP
- uses static routes in the Layer 3 underlay to direct VXLAN traffic, and
- can use the UDP header in VXLAN packets to support network load balancing.

Feature history

The feature history table lists release support for this feature.

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
VXLAN Static Routing	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: - 8711-48Z-M - 8011-32Y8L2H2FH - 8011-12G12X4Y-A/D
VXLAN Static Routing	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
VXLAN Static Routing	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The VXLAN Static Routing functionality is now extended to: - 8712-MOD-M - 8212-48FH-M - 8711-32FH-M - 88-LC1-52Y8H-EM - 88-LC1-12TH24FH-E - 88-LC1-36EH

Feature Name	Release Information	Feature Description

VXLAN Static Routing	Release 24.2.11	You can now configure the source and destination virtual tunnel endpoints (VTEPs) for a particular traffic flow, which is particularly useful for scenarios where your data center is connected to an enterprise network, so multiple servers in the data center provide cloud services to your customers and the enterprise edge router. These endpoints help provide rapid convergence in case of failure. Plus, using the UDP header in the VXLAN packet, the VXLAN static routing (also called unicast VXLAN) facilitates network balancing by preventing the transmission of replicated packets.
		
		Alternatively, you can use Service Layer API for faster provisioning of VXLAN static routing.
		
		This feature is supported only on the following PIDs:
		• 8202-32FH-M
		• 8101-32H
		• 8201-32FH
		
		This feature introduces these changes:
		• **CLI:**
		  • **host-reachability protocol static**
		  • **overlay-encapsulation**
		  • **hw-module profile cef vxlan ipv6-tnl-scale**
		• **YANG Data Model:** (see GitHub, YANG Data Models Navigator)
		  • `Cisco-IOS-XR-tunnel-nve-cfg`
		  • `Cisco-IOS-XR-ip-static-cfg`

VXLAN overlay networks

Explains how VXLAN stretches Layer 2 networks over a Layer 3 IP underlay, uses VTEPs for encapsulation, and identifies segments with VNIs.

A VXLAN overlay network is a Layer 2 tunneling network that

- stretches Layer 2 segments over an underlying Layer 3 IP network
- uses VTEPs to encapsulate and de-encapsulate Ethernet frames, and
- uses a 24-bit VNI to identify Layer 2 segments.

VXLAN behavior

Traditionally, Virtual Local Area Networks (VLANs) are used to partition a single physical network into multiple logical networks. With VLANs, every VLAN has a VLAN ID, which is added to a frame to keep traffic unique. The VLAN ID is 12-bits long, allowing around 4000 unique VLANs.

However, in current networks—such as data centers with extensive virtualization—there may be a need to isolate numerous virtual machines (VMs) from others, resulting in potential exhaustion of VLAN IDs. This drives the need for robust tunneling mechanisms to isolate and load-balance traffic inside the provider's network.

Virtual Extensible LAN (VXLAN) addresses several limitations of traditional VLANs, especially in large-scale and cloud-based environments. VXLAN is widely used in data center environments that require virtualized networks for cloud computing and virtualization technologies. It is also used in service provider networks to offer virtualized network services to customers.

VXLAN is a tunneling protocol that stretches Layer 2 networks over an underlying Layer 3 IP network. The VXLAN tunnel endpoint (VTEP) encapsulates and de-encapsulates Layer 2 traffic. The VTEP encapsulates Layer 2 Ethernet frames within Layer 4 User Datagram Protocol (UDP) and transports these encapsulated frames over a Layer 3 network.

VXLAN introduces an 8-byte VXLAN header, which includes a 24-bit VXLAN network identifier (VNI) and the original Ethernet frame in the UDP payload. The 24-bit VNI is used to identify Layer 2 segments and maintain Layer 2 isolation between segments. With all 24 bits, VXLAN can support up to 16 million LAN segments. The VNI designates individual VXLAN overlay networks, so virtual machines (VMs) in different VXLAN overlays cannot communicate with each other.

VXLAN connects multiple servers in a data center—including those providing cloud services to customers and the enterprise edge router. It automatically configures underlay tunnels between the router and servers, and overlay routing within those tunnels. VXLAN creates virtual networks on top of a physical (underlay) IP network, which can use either IPv4 or IPv6. Underlay and overlay networks are independent; changes in the underlay do not affect the overlay, meaning routers can be added or removed in the underlay without impacting the overlay.

VXLAN allows tunneling of Ethernet frames over IP transport using IP and UDP as the transport protocol. The tunnel extends a Layer 2 segment over a Layer 3 network using MAC-in-UDP encapsulation. A VXLAN header is added to the Layer 2 frame, and the entire packet is placed inside a UDP packet for delivery across the routed domain. The VXLAN tunnel endpoint (VTEP) is typically a router that handles the encapsulation and de-encapsulation of Layer 2 traffic.

When a host sends traffic:

- The VXLAN encapsulates the traffic in UDP and IP headers.
- VXLAN encodes flow information in the UDP source port, enabling routers to perform flow-based load balancing. Flow-based load balancing identifies different flows based on key fields such as source and destination IP addresses.
- VXLAN encapsulates these packets into the tunnel with an IPv4 or IPv6 outer header.
- When the traffic reaches the destination router, it is decapsulated and delivered to the destination host.
- VXLAN adds a custom source MAC address in the inner header, enabling internal devices to extract relevant information from the MAC address.

For more information on VXLAN, see Key Concepts.

Benefits of VXLAN

Lists the benefits of VXLAN overlay networking for data center and service provider environments.

VXLAN provides these key benefits:

- Enables high throughput through dedicated VPN connectivity between servers and enterprise edge routers.
- Allows creation of overlay networks independent of the underlying physical network, offering greater design and deployment flexibility.
- Provides flexible placement of multitenant segments with isolated virtual networks, improving security and separation for multiple tenants.
- Extends Layer 2 segments across the shared infrastructure to manage tenant workloads throughout the data center.
- Uses a 24-bit VXLAN Network Identifier (VNI), supporting up to 16 million unique virtual networks and greater scalability.
- Facilitates network load balancing using the source UDP port within the VXLAN outer header.

VXLAN static routing paths

Provides comprehensive details about VXLAN static route path behavior and route scale, including configuration requirements and supported scale options.

VXLAN static routing enables interconnection between non-VXLAN domains (such as MPLS) and VXLAN domains. It defines the path for VXLAN traffic from the source virtual tunnel endpoint (VTEP) to the destination VTEP by configuring static routes on the underlying Layer 3 network to direct traffic to the appropriate VTEPs.

Key facts about VXLAN static routing path behavior and scale:

- VXLAN static routing is used to connect VXLAN and non-VXLAN environments by manually defining Layer 3 forwarding paths for VXLAN traffic.
- Static routes are configured on the underlying network to control the flow of VXLAN traffic between VTEPs.
- By default, up to 160,000 static routes are supported for VXLAN. The route scale can be increased up to 1 million VXLAN static routes for IPv6 tunnel remote next-hop using the **hw-module profile cef vxlan ipv6-tnl-scale** command.

Benefits of VXLAN static routing

Lists the benefits of using manually configured static routes for VXLAN traffic.

The following are the primary advantages of manually configured static routes for VXLAN traffic:

- You can use static routes in scenarios where consistent routing decisions are required, as static routes are manually configured and the routing behavior is predictable and stable.
- You can specify the next hop for each destination using static routes, allowing for direct control over traffic.
- Static routes are useful for specific traffic engineering or policy requirements.
- You do not have to maintain dynamic routing tables for static routing, thereby reducing any overhead associated with routing protocols.

How VXLAN static routing works

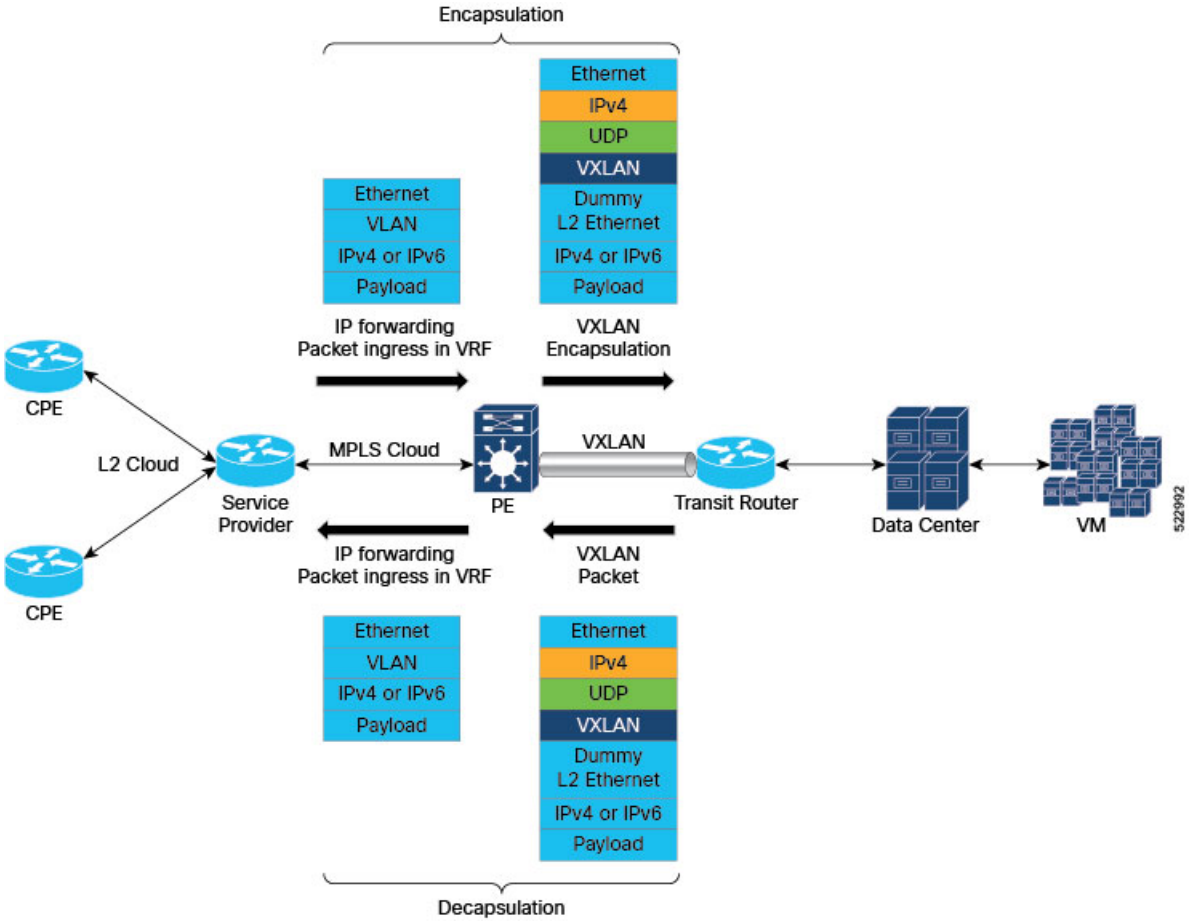
Describes how traffic flows through PE and transit routers in a VXLAN static routing topology, including encapsulation, decapsulation, and routing to and from VMs.

VXLAN static routing is used to extend Layer 2 networks across IP infrastructure. In this topology, traffic from customer devices passes through provider network elements using encapsulation mechanisms that maintain network segmentation and forwarding.

The key components involved in the process are:

- PE router: Receives customer traffic and encapsulates or decapsulates packets using VXLAN.
- VXLAN tunnel: Connects the PE router to the transit router, carrying customer traffic with added headers.
- Transit router: Terminates the VXLAN tunnel, decapsulates packets, and routes them to customer VMs.

In a VXLAN static routing topology, routers use encapsulation and decapsulation to enable traffic flow between customer edge devices and virtual machines across different domains.



These stages describe how VXLAN static routing works:

1. The PE router receives Layer 3 traffic at the VRF interface from the customer edge (CPE).
2. The PE router encapsulates each customer packet with VXLAN headers and applies relevant VLAN tags, mapping VLANs to VRF and VXLAN network identifiers (VNIs).
3. The VXLAN tunnel begins at the PE router and carries encapsulated packets over the network to the transit router or servers behind it.
4. A BGP session is established between PE and transit routers over the VXLAN tunnel to exchange routing information.
5. The PE router distributes VXLAN-encapsulated traffic using a UDP source port (value typically between 49152 and 65535).
6. The transit router receives these packets, terminates (decapsulates) the VXLAN tunnel, and performs an IP lookup.
7. The transit router forwards the traffic to the appropriate customer VM.
8. For return traffic from the VM, the packet is similarly encapsulated as VXLAN with an additional Layer 2 header. Both the VXLAN and inner Layer 2 headers are terminated at the PE router.

The PE and transit routers ensure reliable delivery of traffic between customer networks and virtual machines by encapsulating and decapsulating packets as needed within a VXLAN static routing topology.

VXLAN static routing using the Service Layer API

Provides source details for using the Service Layer API to provision and manage VXLAN static routing.

VXLAN static routing can be provisioned and managed using the Service Layer API, providing several key advantages:

- Enables faster provisioning, easier scaling, and improved overall management of VXLAN networks.

- Allows large cloud providers to dynamically provision tunneling mechanisms at scale to isolate end customer traffic.
- Serves as an efficient alternative to traditional router configuration via CLI, offering granular control over network traffic on the forwarding plane.
- Leverages Google's gRPC to generate client and server bindings so users can program the forwarding plane in a variety of programming languages.

For more information on the Service Layer API, see the *Use Service Layer API to Bring your Controller on Cisco IOS XR Router* chapter in the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

VXLAN key concepts

Explains core principles of VXLAN, including packet format, tunnel endpoints, and load sharing mechanisms, enabling users to grasp essential components and operational behavior of VXLAN technology.

A VXLAN key concept is a foundational component or behavior in VXLAN networking that

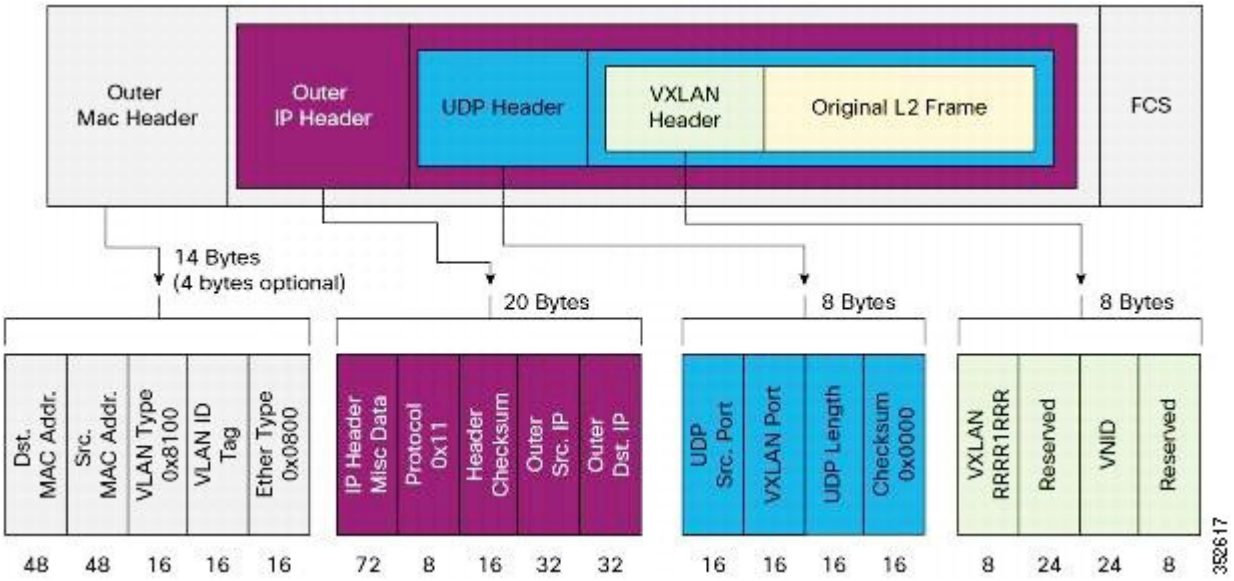
- defines the structure and encapsulation of VXLAN packets
- enables VTEPs to bridge overlay and underlay networks, and
- utilizes source UDP port mapping to achieve efficient load sharing.

VXLAN packet format

Shows the source VXLAN packet format figure.

Here is the VXLAN packet format.

Figure 11: VXLAN Packet Format



VXLAN tunnel endpoints

Explains how VXLAN tunnel endpoints connect overlay and underlay networks, use unique loopback IP addresses for identification, and encapsulate Ethernet frames for reliable transport across an IP network. VXLAN tunnel endpoints facilitate scalable network virtualization and enable efficient communication between distributed hosts.

A VXLAN tunnel endpoint is a physical or virtual router that

- connects overlay and underlay networks

- uses a unique loopback interface IP address in the transport network, and
- encapsulates Ethernet frames for transmission through an IP interface.

Additional reference information
A VXLAN tunnel endpoint (VTEP) can be a physical or virtual router that connects the overlay and underlay networks. A VTEP device is identified in the IP transport network by a unique loopback IP address. The VTEP uses this address to encapsulate Ethernet frames and transmits the encapsulated packets through the IP interface. Source and destination VTEPs create a stateless tunnel to deliver traffic from one host to another. When a frame destined for a remote host arrives, it is encapsulated in IP and UDP headers. Each VTEP can support up to 8,000 VXLAN tunnel interfaces.

How load sharing with VXLANs works

Describes how transport networks use ECMP and VXLAN source UDP ports to load-share VXLAN packet flows.

Most data center transport networks are designed with multiple redundant paths and employ multipath load-sharing technologies to distribute traffic loads efficiently. VXLAN encapsulated packets traverse these paths based on the underlying network's forwarding decisions.

The key components involved in the process are:

- VTEPs: Serve as the source and destination for VXLAN packets in the network.
- ECMP: Allows simultaneous use of multiple best paths in the transport network by balancing traffic loads.
- UDP port numbers (source and destination): Used in VXLAN packet headers; the source UDP port uniquely identifies flows for load-sharing.

Load sharing with VXLANs enables efficient traffic distribution across multiple redundant paths in transport networks. The process utilizes ECMP and the variable VXLAN source UDP port to differentiate flows and optimize path usage.

The process involves these stages:

1. Multipath design and ECMP deployment: Transport networks are configured as IP-routing networks that use ECMP to balance traffic load among multiple best paths.
2. VXLAN packet flow creation: VTEPs encapsulate packets with identical destination IP and UDP port numbers. The source UDP port is varied for each VXLAN flow, creating unique flow identifiers.
3. Load-share hashing and flow differentiation: The transport network uses the VXLAN source UDP port for load-share hashing, ensuring that flows are distributed across available ECMP paths. This helps avoid out-of-sequence packet forwarding and maximizes path utilization.
4. Packet forwarding and distribution: As VXLAN packets enter the transport network, the source UDP port guides their distribution across multiple redundant paths, ensuring efficient load balancing.

The transport network can distinguish VXLAN flows by source UDP port and distribute them across ECMP paths, maximizing bandwidth utilization and minimizing congestion.

Configure VXLAN static routing

Provides instructions for configuring VXLAN static routing, detailing the procedural steps required to implement static routing within a VXLAN network.

Set up static routing over VXLAN on a provider edge router to enable efficient and secure connectivity between network segments

Perform the following tasks on the provider edge (PE) router to configure VXLAN static routing:

- Configure VRF
- Configure interface NVE for decapsulation

- Configure static routing
- Configure customized UDP destination port and UDP source port range

Plan the VRFs, NVE interface, VNIs, remote next hops, source MAC address, and UDP port values before you begin.

1. Configure the VRF.

```
Router# configure
Router(config)# vrf vrf1
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target
Router(config-vrf-import-rt)# 1:1
Router(config-vrf-import-rt)# exit
Router(config-vrf-af)# export route-target
Router(config-vrf-export-rt)# 1:1
Router(config-vrf-export-rt)# exit
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# import route-target
Router(config-vrf-import-rt)# 1:1
Router(config-vrf-import-rt)# exit
Router(config-vrf-af)# export route-target
Router(config-vrf-export-rt)# 1:1
Router(config-vrf-export-rt)# root
Router(config)# vrf vrf2
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target
Router(config-vrf-import-rt)# 1:2
Router(config-vrf-import-rt)# exit
Router(config-vrf-af)# export route-target
Router(config-vrf-export-rt)# 1:2
Router(config-vrf-export-rt)# exit
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# import route-target
Router(config-vrf-import-rt)# 1:2
Router(config-vrf-import-rt)# exit
Router(config-vrf-af)# export route-target
Router(config-vrf-export-rt)# 1:2
Router(config-vrf-export-rt)# commit
```

2. Configure the NVE interface for decapsulation.

```
Router(config)# interface nve1
Router(config-if)# member vni 2
Router(config-nve-vni)# vrf vrf1
Router(config-nve-vni)# host-reachability protocol static
Router(config-nve-vni)# exit
Router(config-if)# member vni 6
Router(config-nve-vni)# vrf vrf2
Router(config-nve-vni)# host-reachability protocol static
Router(config-nve-vni)# exit
Router(config-if)# overlay-encapsulation vxlan
Router(config-nve-encap-vxlan)# peer-ip lookup disable
Router(config-nve-encap-vxlan)# exit
Router(config-if)# source-interface Loopback1
Router(config-if)# commit
```

3. Configure static routing.

```
Router# configure
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 10.10.10.10/32 10.151.11.2
Router(config-static-afi)# 10.10.10.11/32 10.151.11.2
Router(config-static-afi)# exit
Router(config-static)# vrf VRF1
Router(config-static-vrf)# address-family ipv4 unicast
Router(config-static-vrf-afi)# 192.0.2.10/32 remote-next-hop 10.10.10.10
tunnel VXLAN index 1 nve 1 evni 1 src-mac aaa1.bbb1.ccc1 ->  IPv4 over IPv4
Router(config-static-vrf-afi)# 192.0.2.11/32 remote-next-hop 10:10:10::10
tunnel VXLAN index 3 nve 1 evni 3 src-mac aaa1.bbb1.ccc1 -> IPv4 over IPv6
Router(config-static-vrf-afi)# exit
Router(config-static-vrf)# exit
Router(config-static)# address-family ipv6 unicast
Router(config-static-afi)# 10:10:10::10/128 10:151:11::2
Router(config-static-afi)# 10:10:10::11/128 10:151:11::2
Router(config-static-afi)# exit
Router(config-static)# vrf VRF1
Router(config-static-vrf)# address-family ipv6 unicast
Router(config-static-vrf-afi)# 11:1:1::1/128 remote-next-hop 10.10.10.11
tunnel VXLAN index 2 nve 1 evni 2 src-mac aaa1.bbb1.ccc1 ->  IPv6 over IPv4
Router(config-static-vrf-afi)# 11:1:1::2/128 remote-next-hop 10:10:10::11
tunnel VXLAN index 4 nve 2 evni 4 src-mac aaa1.bbb1.ccc1 -> IPv6 over IPv6
Router(config-static-vrf-afi)# commit
```

4. Configure UDP destination port and source port range for VXLAN.

```
Router# configure
Router(config)# nve
Router(config-nve)# overlay-encap vxlan
Router(config-vxlan)# udp-port destination 65330
Router(config-vxlan)# udp-port src-port start 1024
Router(config-vxlan)# commit
```

5. Review the VXLAN static routing running configuration.

```
vrf vrf1
 address-family ipv4 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
 address-family ipv6 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
vrf vrf2
 address-family ipv4 unicast
  import route-target
   1:2
  !
```

```
  export route-target
   1:2
  !
 !
 address-family ipv6 unicast
  import route-target
   1:2
  !
  export route-target
   1:2

interface nve1
 member vni 2
  vrf vrf1
  host-reachability protocol static
 !

 member vni 6
  vrf vrf2
  host-reachability protocol static
 !
 overlay-encapsulation vxlan
 peer-ip lookup disable
!
 source-interface Loopback0


router static
address-family ipv4 unicast
  10.10.10.10/32 10.151.11.2
  10.10.10.11/32 10.151.11.2
!
address-family ipv6 unicast
  10:10:10::10/128 10:151:11::2
  10:10:10::11/128 10:151:11::2

vrf vrf1
  address-family ipv4 unicast
   192.0.2.10/32 remote-next-hop 10.10.10.10 tunnel VXLAN index 1 nve 1 evni
 1 src-mac aaa1.bbb1.ccc1. ====> IPv4oIPv4

   192.0.2.11/32 remote-next-hop 10:10:10::10 tunnel VXLAN index 3 nve 1 evni
 3 src-mac aaa1.bbb1.ccc1 ====> IPv4oIPv6
  !
  address-family ipv6 unicast
   11:1:1::1/128 remote-next-hop 10.10.10.11 tunnel VXLAN index 2 nve 1 evni
 2 src-mac aaa1.bbb1.ccc1 ====> ipv6 over ipv4

   11:1:1::2/128 remote-next-hop 10:10:10::11 tunnel VXLAN index 4 nve 2 evni
 4 src-mac aaa1.bbb1.ccc1 ====> ipv6 over ipv6


configure
 nve
  overlay-encap vxlan
   udp-port destination 65330
   udp-port src-port start 1024
  !
 !
```

This section shows VXLAN static routing running configuration.

6. Verify the NVE interface and static route state.

```
Router# show nve interface nve 1
Interface: nve1 State: Up Encapsulation: VxLAN
  Source Interface: Loopback0 (primary: v4: 192.0.2.12 v6: 1:1:1::1)

Router# show nve global

 NVE Global details
   VNI Scope Local : No
   VxLAN Src Port  : 1024
   VxLAN Destination Port : 65330
   VxLAN interfaceless l3vni bring up: TRUE
   Count of NVE interfaces with mpls-udp encap: 0
   Global system mac: 9c54.1643.f900
```

```
Router# show route vrf VRF1 192.0.2.10 detail

Routing entry for 192.0.2.10/32
  Known via "static", distance 1, metric 0
  Installed Nov 30 17:44:37.003 for 02:22:09
  Routing Descriptor Blocks
    10.10.10.10, via Bundle-Ether13.5
      Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id:
0xe0000000
      Route metric is 0
      Label: None
      Tunnel ID: None
      Binding Label: None
      Extended communities count: 0
      NHID:0x0(Ref:0)
      IP Tunnel Info: Auto create: 1 Tunnel Type: vxlan-l3 eVNI: 0x1 Tunnel
ID: 0x1 RTEP ID: 0x100000000000001
      MPLS eid:0xfffffffffffffffd
  Route version is 0x1 (1)
  No local label
  IP Precedence: Not Set
  QoS Group ID: Not Set
  Flow-tag: Not Set
  Fwd-class: Not Set
  Route Priority: RIB_PRIORITY_RECURSIVE (9) SVD Type RIB_SVD_TYPE_LOCAL
  Download Priority 3, Download Version 12
  Route eid: 0xfffffffffffffffd
  No advertising protos.
```

VXLAN static routing is configured and operational on the provider edge router. The NVE interface should be up, and the route detail output must display the expected VXLAN tunnel information.

Configure VXLAN static routing using the Service Layer API

Details configuration procedures for VXLAN static routing using Service Layer API, guiding users through the steps to automate and manage static routes in VXLAN deployments via programmatic interfaces.

Set up and verify Service Layer API and gRPC for VXLAN static routing, ensuring the service-layer state is correct.

Use the Service Layer API for environments where you require faster provisioning, easier scaling, and improved management of VXLAN static routes.

Plan the gRPC port, address family, and service-layer security settings in advance.

1. Configure gRPC and enable Service Layer API.

```
Router# configure
Router(config)# grpc
Router(config-grpc)# port 57777
Router(config-grpc)# address-family ipv4
Router(config-grpc)# service-layer
Router(config-grpc)# no-tls
Router(config-grpc)# commit
```

2. Review the Service Layer API running configuration.

```
grpc
 port 57777
 address-family ipv4
 service-layer
 no-tls
!
```

3. Use the **show service-layer state** command to verify the Service Layer API state.

```
Router# show service-layer state

------------service layer state----------
config on:                    NO
connected to RIB for IPv4:    YES
connected to RIB for IPv6:    YES
Initialization state:         initialized
pending requests:             0
BFD Connection:               DOWN
MPLS Connection:              DOWN
Interface Connection:         UP
Objects accepted:             NO
interface registered:         NO
bfd registered for IPv4:      NO
bfd registered for IPv6:      NO
```

The Service Layer API is configured. The VXLAN static routing configuration is complete when the service-layer state output confirms the expected RIB and interface connections.

8 IPv6 VPN Provider Edge Transport over MPLS

Topics:

Outlines IPv6 VPN provider edge transport mechanisms over MPLS, including deployment considerations, configuration procedures, protocol integration, and verification steps for efficient and secure networking across MPLS infrastructures.

IPv6 VPN provider edge transport services

Introduces IPv6 VPN provider edge transport services using MPLS, detailing 6PE and 6VPE architectures, service integration options, inter-AS requirements, solution benefits, operational workflows, edge and customer device roles, OSPFv3 route exchange, and configuration procedures for BGP and OSPFv3 connectivity.

IPv6 VPN provider edge transport services are network transport methods that

- use the existing MPLS IPv4 core infrastructure to support IPv6 communication
- enable IPv6 sites to communicate across MPLS label switched paths (LSPs), and
- leverage multiprotocol BGP extensions to exchange IPv6 reachability information and MPLS labels.

Implementation context

IPv6 Provider Edge (6PE) and IPv6 VPN Provider Edge (6VPE) use the existing MPLS IPv4 core infrastructure to transport IPv6 traffic. These services enable IPv6 sites to communicate with each other over an MPLS IPv4 core network by establishing MPLS label switched paths (LSPs).

The feature uses multiprotocol Border Gateway Protocol (BGP) extensions configured on the provider edge (PE) routers within the IPv4 network. These extensions allow the exchange of IPv6 reachability information and assign an MPLS label for each IPv6 address prefix.

Edge routers are configured as dual-stack, running both IPv4 and IPv6 protocols. They use IPv4-mapped IPv6 addresses to facilitate the exchange of IPv6 prefix reachability.

To implement 6PE or 6VPE, familiarity with MPLS and BGP4 configuration and troubleshooting is essential.

6PE and 6VPE services

Explains 6PE and 6VPE service fundamentals, outlining integration options for IPv6 services over MPLS, and clarifies Inter-AS and next-hop behavior requirements for scalable network designs.

6PE and 6VPE services are provider edge deployment techniques that

- integrate IPv6 services over existing service provider backbones
- preserve MPLS IPv4 infrastructure while adding IPv6 service reachability, and
- use dual-stack edge routers and multiprotocol BGP extensions to exchange reachability information.

Feature history

The feature history table lists release support for this feature.

Table 9: Feature History Table

Feature Name	Release	Description
6PE/6VPE	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: - 8711-48Z-M - 8011-32Y8L2H2FH - 8011-12G12X4Y-A/D

Feature Name	Release	Description
6PE/6VPE	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature support is now supported on Cisco 8011-4G24Y4H-I routers.
6PE/6VPE	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The router now enables seamless integration of IPv6 networks over an MPLS backbone, allowing service providers to offer IPv6 services without changing the core MPLS infrastructure. This feature facilitates IPv6 routing by leveraging existing IPv4 MPLS paths, ensuring efficient and scalable network expansion. IPv6 prefixes are advertised as VPNv4 routes, which simplifies deployment and reduces operational complexity. This approach requires no IPv6 capabilities in the MPLS core, leading to cost savings and a smoother transition to IPv6. *Previously this feature was supported on Q200 and Q100. It is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

IPv6 service integration options

Lists source options and operational context for integrating IPv6 services across service provider backbones.

Multiple techniques are available to integrate IPv6 services over service provider core backbones:

- Dedicated IPv6 network: Runs over various underlying data link layers.
- Dual-stack IPv4-IPv6 backbone: Operates both IPv4 and IPv6 protocols in parallel.
- MPLS backbone leverage: Uses an existing MPLS backbone to carry IPv6 traffic.

These solutions are deployed on service provider backbones when the volume of IPv6 traffic and revenue justifies the investment and associated risks. Favorable conditions allow for the introduction of native IPv6 services from the network

edge in a scalable way, avoiding IPv6 addressing constraints and preserving a stable IPv4 backbone. Maintaining backbone stability is critical, especially for service providers that have recently stabilized their IPv4 infrastructure.

Service providers operating an MPLS/IPv4 infrastructure often use similar approaches, as multiple scenarios for offering IPv6 services over MPLS are possible. Cisco Systems, for example, developed the Cisco 6PE (IPv6 Provider Edge Router over MPLS) solution to meet these requirements.

6PE and 6VPE Inter-AS and next-hop behavior

Outlines requirements for Inter-AS support and next-hop display behavior in 6PE and 6VPE deployments.

Ensure that your deployment follows these requirements for 6PE and 6VPE Inter-AS operations:

- Support Border Gateway Protocol (BGP) to enable relevant address families and allocate and distribute PE and ASBR labels for Inter-AS 6PE deployments.
- Cisco IOS XR must display actual IPv4 next-hop addresses for IPv6 labeled-unicast and VPNv6 prefixes. IPv4-mapped-to-IPv6 format is not supported.

Benefits of 6PE and 6VPE

Details the advantages offered by 6PE and 6VPE, highlighting improvements in IPv6 service delivery, operational efficiency, and MPLS backbone integration.

Service providers who currently deploy MPLS experience these benefits of Cisco 6PE/6VPE:

- Minimal operational cost and risk—No impact on existing IPv4 and MPLS services.
- Provider edge routers upgrade only—A 6PE/6VPE router can be an existing PE router or a new one dedicated to IPv6 traffic.
- No impact on IPv6 customer edge routers—The ISP can connect to any customer CE running Static, IGP or EGP.
- Production services ready—An ISP can delegate IPv6 prefixes.
- IPv6 introduction into an existing MPLS service—6PE/6VPE routers can be added at any time

How IPv6 over MPLS backbones works

Describes the operational workflow of transporting IPv6 over MPLS backbones, illustrating encapsulation, traffic flow, and protocol interactions within core networks.

The 6PE mechanism allows operators to introduce IPv6 services without disrupting or upgrading their existing MPLS IPv4 backbone. Label-based forwarding ensures efficient packet transit and preserves network stability.

The key components involved in the process are:

- IPv6 domain: A network area using IPv6 addresses and protocols requiring connectivity across the backbone.
- MPLS IPv4 core network: The underlying infrastructure that forwards packets based on MPLS labels, not IP headers.
- 6PE Provider Edge Router: A specialized edge router that encapsulates IPv6 packets within MPLS labels for transit across the IPv4 backbone.

IPv6 over MPLS backbones enables seamless communication between IPv6 domains by leveraging an existing MPLS IPv4 core network. This implementation avoids reconfiguring core routers and requires no infrastructure upgrades, providing a cost-effective solution for IPv6 deployment.

These stages describe how IPv6 over MPLS backbones works:

1. IPv6 packet origination: An IPv6 domain generates an IPv6 packet destined for another IPv6 domain across the backbone.
2. Provider Edge router encapsulation: The 6PE-enabled provider edge router encapsulates the IPv6 packet within an MPLS label, preparing it for transit.

3. Transit across MPLS IPv4 core: The MPLS IPv4 core network transports the labeled packet solely based on the MPLS label, bypassing the need to interpret IPv6 headers.
4. Provider Edge router decapsulation: At the destination edge, another 6PE router removes the MPLS label and routes the packet into the target IPv6 domain.
5. Reachability exchange: Provider edge routers share reachability information using BGP extensions to ensure end-to-end IPv6 connectivity.

IPv6 domains successfully communicate across the MPLS IPv4 backbone by utilizing MPLS label-switched paths (LSPs) and reachability exchange. The solution leverages the existing MPLS infrastructure, offering a scalable and low-impact IPv6 deployment.

IPv6 on provider edge and customer edge routers

Outlines IPv6 functionality and deployment on provider edge and customer edge routers, including required hardware roles, tunneling mechanisms for carrying IPv6, and multipath operations in provider edge devices.

A provider edge and customer edge router are network devices that

- interconnect IPv6 islands over an MPLS IPv4 core as part of 6PE and 6VPE deployments
- exchange routes between each other to enable seamless IPv6 communication, and
- support multipath behavior to provide load sharing and redundancy.

Roles of service provider edge routers in MPLS IPv6 deployments

Provides a summary of the roles, features, and advantages of service provider edge routers in MPLS-based IPv6 services (6PE and 6VPE).

Service provider edge routers play a crucial part in enabling IPv6 services over an MPLS network using technologies such as 6PE (IPv6 Provider Edge) and 6VPE (IPv6 VPN Provider Edge). These routers offer several key roles and advantages:

- Protocol support: Enable delivery of IPv6 services (both global routing and VPN), leveraging existing MPLS infrastructure.
- Infrastructure efficiency: Do not require hardware, software, or configuration upgrades in the core network; minimize impact on ongoing revenue-generating IPv4 traffic.
- Service versatility: Support multiservice capabilities including Layer 3 VPNs, QoS, traffic engineering, fast re-routing, and seamless integration of ATM and IP switching.

These features allow service providers to deploy IPv6 and VPN services efficiently without disrupting current operations or requiring large-scale upgrades. Edge routers thus act as the main interface for MPLS-based IPv6 service delivery to customers.

How customer edge router tunnels carry IPv6

Describes how customer edge router tunnels can carry IPv6 traffic across an MPLS IPv4 network.

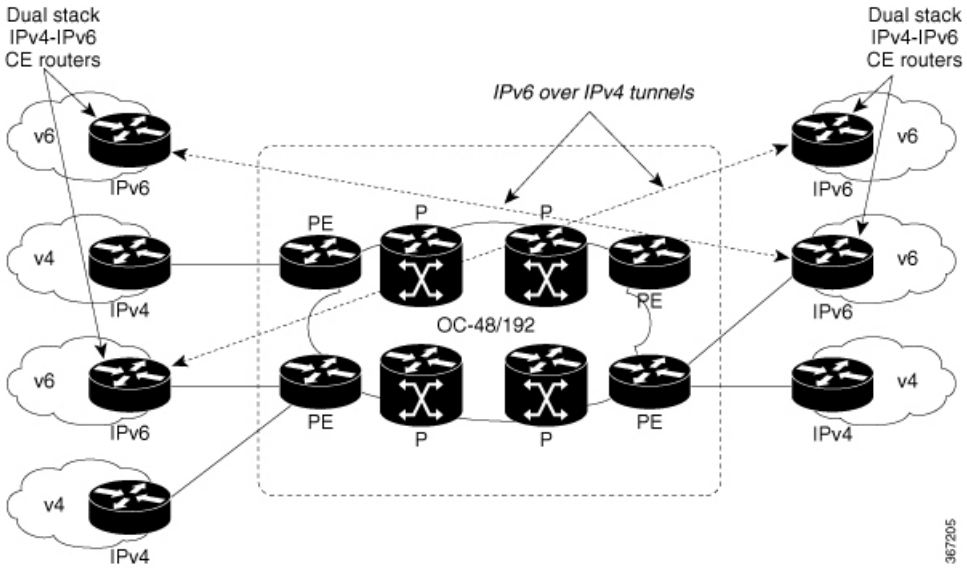
Tunnel meshing is required as the number of CEs to connect increases. Additionally, delegating a global IPv6 prefix for an ISP can be challenging in this topology.

The key components involved in the process are:

- Customer Edge (CE) router: Acts as the tunnel endpoint and encapsulates IPv6 packets for transport over an MPLS IPv4 network.
- Provider Edge (PE) router: Maintains standard MPLS connectivity and does not require configuration changes for IPv6 tunneling.
- Provider (P) router: Remains unchanged and forwards packets using the existing MPLS IPv4 infrastructure.

Using tunnels on customer edge (CE) routers is the simplest way to deploy IPv6 over MPLS networks. This approach does not affect MPLS operation or infrastructure and requires no changes to provider (P) routers or provider edge (PE) routers.

Figure 12: IPv6 Using Tunnels on the CE Routers



These stages describe how customer edge router tunnels carry IPv6:

1. The CE router encapsulates IPv6 packets inside IPv4 tunnels.
2. Encapsulated packets traverse the service provider's MPLS IPv4 core network without requiring changes to P or PE routers.
3. On reaching the destination CE router, IPv6 packets are decapsulated and forwarded to the customer network.

Customer edge tunnel endpoints successfully carry IPv6 traffic while provider edge routers maintain standard IPv4 MPLS connectivity, enabling IPv6 deployment without major core network changes.

How IPv6 provider edge multipath works

Describes how IPv6 provider edge (6PE) multipath uses internal and external BGP to provide load sharing and redundancy by distributing IPv6 routes across multiple MPLS-labeled paths in a provider edge network.

Internal and external BGP multipath for IPv6 allows the IPv6 router to balance load between several paths (for example, the same neighboring autonomous system (AS) or sub-AS, or the same metrics) to reach its destination. The 6PE multipath feature uses multiprotocol internal BGP (MP-IBGP) to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route.

The key components involved in the process are:

- IPv6 router: Selects multiple available BGP paths to forward IPv6 traffic.
- Internal and external BGP multipath: Supports load sharing by distributing traffic across several paths, potentially within the same neighboring autonomous system (AS) or sub-AS.
- Multiprotocol internal BGP (MP-IBGP): Distributes IPv6 routes as MPLS-labeled packets across the IPv4 core network.

IPv6 provider edge multipath enables load balancing and redundancy in service provider networks by allowing routers to use multiple internal or external BGP paths for IPv6 traffic.

These stages describe how IPv6 provider edge multipath works:

1. Path selection: The IPv6 provider edge (6PE) router evaluates available BGP paths to the destination, considering internal and external multipath scenarios.

2. Label distribution: MP-IBGP distributes IPv6 routes and attaches an MPLS label to each route across the IPv4 MPLS core.
3. Forwarding table installation: When multipath is enabled, all MPLS-labeled paths are installed in the forwarding table, enabling load balancing.
4. Traffic distribution: IPv6 traffic is distributed across multiple paths based on the available MPLS label information.

IPv6 traffic is efficiently load-shared and benefits from redundant paths, improving network resiliency and overall performance.

OSPFv3 CE-to-PE route exchange services

Introduces OSPFv3 protocol support between customer edge and provider edge routers, emphasizing multiple VRF integration, PE-CE protocol extensions, and requirements for VRF lite behavior within the routing architecture.

OSPFv3 CE-to-PE route exchange services are routing capabilities that

- support multiple VRFs per OSPFv3 routing process
- use OSPFv3 PE-CE extensions in the VPN environment, and
- support VRF lite deployment without a BGP or MPLS based backbone.

Additional reference information

The Open Shortest Path First version 3 (OSPFv3) IPv6 VPN Provider Edge (6VPE) feature adds VPN routing and forwarding (VRF) and provider edge-to-customer edge (PE-CE) routing support to Cisco IOS XR OSPFv3 implementations. This feature enables:

- Multiple VRF support per OSPFv3 routing process.
- OSPFV3 PE-CE extensions.

Multiple VRF support in OSPFv3

Provides source details for multiple VRF support in a single OSPFv3 routing process.

OSPFv3 supports multiple VRFs in a single routing process, which enables scaling to tens or hundreds of VRFs without overloading route processor resources. Multiple OSPFv3 processes can be configured on a single router, allowing for partitioned VRF processing across several route processors in large-scale VRF deployments. This capability can also be used to isolate the default routing table or high-impact VRFs from regular VRFs. It is recommended to use a single process for all VRFs. If needed, a second OSPFv3 process should be configured for IPv6 routing.

> **Note**
>
> A maximum of four OSPFv3 processes are supported.

OSPFv3 PE-CE extension requirements for IPv6 VPN environments

Provides source details for OSPFv3 PE-CE extensions in IPv6 VPN environments.

The following facts outline key requirements and features of OSPFv3 PE-CE extensions in the context of IPv6 VPN:

- Service Providers increasingly deploy IPv6 protocol in modern customer networks.
- VPN services must support both IPv4 and IPv6 protocols for customer environments.
- To support IPv6, routing protocols require additional extensions to operate in VPN environments.
- OSPFv3 must be extended specifically for operation at Provider Edge (PE) and Customer Edge (CE) links in IPv6 VPNs.

OSPFv3 VRF lite behavior
Outlines the VRF lite behavior and restrictions that apply to OSPFv3 deployments.

To comply with the requirements for OSPFv3 VRF lite deployment, ensure that you:

- Disable DN bit processing in the VRF lite environment.
- Prevent automatic ABR status assignment in VRF contexts (except default VRF), regardless of connectivity to area 0. In the VRF lite environment, automatic ABR status setting must be disabled.

You must run the **capability vrf-lite** command in the OSPFv3 VRF configuration submode to enable VRF lite.

Configure 6PE and 6VPE

Provides instructions for configuring 6PE and 6VPE services, guiding users through setup steps to enable IPv6 VPN solutions over MPLS backbones.

Enable the transport of IPv6 prefixes across an IPv4 backbone by configuring 6PE and 6VPE protocols on PE routers.

Use 6PE and 6VPE to allow IPv6 reachability and VPN functionality across an IPv4 cloud. Supported routing protocols vary:

- For 6PE: BGP, OSPF, IS-IS, and Static protocols are supported to learn routes from both clouds.
- For 6VPE: Only BGP and Static protocols are supported for learning routes from IPv4 and IPv6 clouds. OSPFv3 is supported between PE and CE routers.
- Plan which PE routers will participate in both the IPv4 and IPv6 clouds.
- Configure route policies before attaching them to routers.
- Set label allocation mode to "per-vrf" on all routers, including peer routers.

Follow these steps to configure 6PE and 6VPE:

1. Configure BGP settings and enable required address families.

```
Router#configure
Router(config)#router bgp 10
Router(config-bgp)#bgp router-id 192.0.2.10
Router(config-bgp)#graceful-restart
Router(config-bgp)#log neighbor changes detail
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#redistribute connected
Router(config-bgp-af)#redistribute ospfv3 7
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#commit
Router(config-bgp)#neighbor 66:1:2::2
Router(config-bgp-nbr)#remote-as 102
Router(config-bgp-nbr)#address-family ipv6 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
Router(config-bgp)#neighbor 192.0.2.11
Router(config-bgp-nbr)#remote-as 10
Router(config-bgp-nbr)#update-source Loopback0
Router(config-bgp-nbr)#address-family vpnv4 unicast
Router(config-bgp-nbr-af)#address-family ipv6 labeled-unicast
Router(config-bgp-nbr-af)#address-family vpnv6 unicast
Router(config-bgp-nbr-af)#commit
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit
Router(config-bgp)#vrf red
```

```
Router(config-bgp-vrf)#rd 500:1
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)#redistribute static
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)#redistribute static
Router(config-bgp-vrf-af)#commit
Router(config)#interface HundredGigE0/0/1/0
Router(config-if)#vrf red
Router(config-if)#ipv6 address 4002:110::1/128
Router(config-if)#exit
Router(config)#vrf red
Router(config-vrf)#address-family ipv4 unicast
Router(config-vrf-af)#label mode per-vrf
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#500:1
Router(config-vrf-import-rt)#!
Router(config-vrf-import-rt)#export route-target
Router(config-vrf-export-rt)#500:1
Router(config-vrf-export-rt)#!
Router(config-vrf-export-rt)#!
Router(config-vrf-export-rt)#address-family ipv6 unicast
Router(config-vrf-af)#label mode per-vrf
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#500:1
Router(config-vrf-import-rt)#!
Router(config-vrf-import-rt)#export route-target
Router(config-vrf-export-rt)#500:1
Router(config-vrf-export-rt)#commit
```

2. Configure BGP neighbors for IPv6 and VPN connectivity.

```
Router(config-bgp)#neighbor 66:1:2::2
Router(config-bgp-nbr)#remote-as 102
Router(config-bgp-nbr)#address-family ipv6 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
Router(config-bgp)#neighbor 192.0.2.11
Router(config-bgp-nbr)#remote-as 10
Router(config-bgp-nbr)#update-source Loopback0
Router(config-bgp-nbr)#address-family vpnv4 unicast
Router(config-bgp-nbr-af)#address-family ipv6 labeled-unicast
Router(config-bgp-nbr-af)#address-family vpnv6 unicast
Router(config-bgp-nbr-af)#commit
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit
```

3. Configure VRF settings and label modes.

```
Router(config-bgp)#vrf red
Router(config-bgp-vrf)#rd 500:1
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)#redistribute static
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)#redistribute static
Router(config-bgp-vrf-af)#commit
```

4. Configure interface and VRF address settings.

```
Router(config)#interface HundredGigE0/0/1/0
Router(config-if)#vrf red
Router(config-if)#ipv6 address 4002:110::1/128
Router(config-if)#exit
Router(config)#vrf red
Router(config-vrf)#address-family ipv4 unicast
Router(config-vrf-af)#label mode per-vrf
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#500:1
Router(config-vrf-import-rt)#exit
Router(config-vrf-import-rt)#export route-target
Router(config-vrf-export-rt)#500:1
Router(config-vrf-export-rt)#exit
Router(config-vrf-export-rt)#address-family ipv6 unicast
Router(config-vrf-af)#label mode per-vrf
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#500:1
Router(config-vrf-import-rt)#exit
Router(config-vrf-import-rt)#export route-target
Router(config-vrf-export-rt)#500:1
Router(config-vrf-export-rt)#commit
```

5. Set route-target import and export for VRF.

```
Router(config)#vrf red
Router(config-vrf)#address-family ipv4 unicast
Router(config-vrf-af)#label mode per-vrf
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#500:1
Router(config-vrf-import-rt)#exit
Router(config-vrf-import-rt)#export route-target
Router(config-vrf-export-rt)#500:1
Router(config-vrf-export-rt)#exit
Router(config-vrf-export-rt)#address-family ipv6 unicast
Router(config-vrf-af)#label mode per-vrf
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#500:1
Router(config-vrf-import-rt)#exit
Router(config-vrf-import-rt)#export route-target
Router(config-vrf-export-rt)#500:1
Router(config-vrf-export-rt)#commit
```

This example shows how to configure 6PE on PE routers to transport the IPv6 prefixes across the IPv4 cloud. Ensure that you configure 6PE on PE routers participating in both the IPv4 cloud and IPv6 clouds. Pointers:

- For 6PE, all routing protocols supported on Cisco IOS XR (BGP, OSPF, IS-IS, Static) can be used for route learning.
- For 6VPE, only BGP and Static protocols are supported for route learning, and OSPFv3 can be used between PE and CE routers.

- It is mandatory to configure **per-vrf** label allocation mode on all routers (including peers).
- Route policies must be created prior to configuring 6PE/6VPE.
- Starting from Cisco IOS XR Release 7.5.3, BGP assigns a label value of 2 (IPv6 Explicit NULL Label) to IPv6 prefixes, replacing earlier random label assignment.

6. Review the running configuration.

```
router bgp 10
bgp router-id 192.0.2.10
bgp graceful-restart
bgp log neighbor changes detail
!
address-family ipv6 unicast
 redistribute connected
  redistribute ospfv3 7
  allocate-label all
!
!
neighbor 66:1:2::2
  remote-as 201
  address-family ipv6 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
!
neighbor 192.0.2.11
  remote-as 10
  update-source Loopback0
  address-family vpnv4 unicast
  !
  address-family ipv6 labeled-unicast
  !
  address-family vpnv6 unicast
!
vrf red
  rd 500:1
  address-family ipv4 unicast
   label mode per-vrf
   redistribute connected
   redistribute static
  !
  address-family ipv6 unicast
   label mode per-vrf
   redistribute connected
   redistribute static
  !
 !
!
interface HundredGigE0/0/1/0
vrf red
Ipv6 address 4002:110::1/128
!
exit
vrf red
address-family ipv4 unicast
import route-target
500:1
!
export route-target
500:1
```

```
!
!
address-family ipv6 unicast
import route-target
500:1
!
export route-target
500:1
!
```

7. Verify the configuration.

```
Router# show route ipv6
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, (!) - FRR Backup path
Gateway of last resort is not set

L    ::ffff:127.0.0.0/104
      [0/0] via ::, 02:10:49
C    66:1:2::/64 is directly connected,
      02:09:39, TenGigE0/0/0/10.2
L    66:1:2::1/128 is directly connected,
      02:09:39, TenGigE0/0/0/10.2
C   66:1:3::/64isdirectlyconnected,
[20/0] via fe80::200:2cff:fe64:99e2, 02:07:38, TenGigE0/0/0/10.2
B    2000:0:0:1c::/64
      [20/0] via fe80::200:2cff:fe64:99e2, 02:07:38, TenGigE0/0/0/10.2
B    2000:0:0:1d::/64
Local PE :
Router# show bgp ipv6 labeled-unicast 2000:0:0:1c::/64
BGP routing table entry for 2000:0:0:1c::/64
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               5033        5033
    Local Label: 66313
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.1
  Advertised to peers (in unique update groups):
    192.0.2.11
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.1
  Advertised to peers (in unique update groups):
    192.0.2.11
  201
    66:1:2::2 from 66:1:2::2 (192.0.2.12)
      Origin IGP, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 0, version 5033
      Origin-AS validity: not-found

Remote PE
Router# show bgp ipv6 labeled-unicast 2000:0:0:1c::/64
BGP routing table entry for 2000:0:0:1c::/64
```

```
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker             139679      139679
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.2
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.2
  201
    192.0.2.10 (metric 5) from 192.0.2.11 (192.0.2.10)
      Received Label 66313
      Origin IGP, localpref 100, valid, internal, best, group-best,
labeled-unicast
      Received Path ID 0, Local Path ID 0, version 139679
      Originator: 192.0.2.10, Cluster list: 192.0.2.13
```

Configure OSPFv3 between PE and CE routers

Configure PE-to-CE routing sessions that use Open Shortest Path First version 3 (OSPFv3).

Establish OSPFv3 routing sessions between provider edge (PE) and customer edge (CE) routers for IPv6 connectivity within a VRF.

Configure PE-to-CE routing sessions that use Open Shortest Path First version 3.Use OSPFv3 to enable dynamic routing between PE and CE devices over an MPLS VPN, leveraging VRF for traffic segmentation.

Plan the required OSPFv3 process ID, router ID, VRF name, OSPF area, and identify the relevant interfaces that will participate in routing.

1. Configure OSPFv3 between PE and CE routers.

```
Router# configure
Router(config)# router ospfv3 7
Router(config-ospfv3)# router-id 10.200.1.7
Router(config-ospfv3)# vrf vrf1
Router(config-ospfv3-vrf)# area 7
Router(config-ospfv3-vrf-ar)# interface Loopback7
Router(config-ospfv3-vrf-ar-if)# exit
Router(config-ospfv3-vrf-ar-if)# interface TenGigE0/7/0/0/3.7
Router(config-ospfv3-vrf-ar-if)# commit
```

   This example shows how to configure provider edge (PE)-to-customer edge (CE) routing sessions that use Open Shortest Path First version 3 (OSPFv3).

2. Review the running configuration.

```
router ospfv3 7
router-id 10.200.1.7
vrf vrf1
  area 7
   interface Loopback7
   !
   interface TenGigE0/7/0/0/3.7
   !
  !
!
```

3. Use the **show ospfv3 7 vrf vrf1 neighbor** command to verify OSPFv3 neighbor relationships.

```
Router# show ospfv3 7 vrf vrf1 neighbor
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPFv3 7, VRF vrf1

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
10.201.7.1      0     FULL/DROTHER    00:00:36    0
TenGigE0/7/0/0/3.7
    Neighbor is up for 1w0d

Total neighbor count: 1
```

The configuration is complete when the router successfully joins the OSPFv3 area and neighbor verification shows the expected FULL state.

Configure BGP between PE and CE routers

Configure BGP as the routing protocol between provider edge (PE) and customer edge (CE) routers to enable VPN-IPv6 prefix distribution across provider and customer sites.

Configure BGP as the routing protocol between PE and CE routers.

BGP distributes reachability information for VPN-IPv6 prefixes for each VPN. PE to PE or PE to route reflector (RR) sessions use iBGP, while PE to CE sessions use eBGP. PE to CE eBGP sessions can be directly or indirectly connected (eBGP multihop).

Configure the route policy, such as pass-all, before you attach it to the PE-to-CE neighbor.

1. Configure BGP between PE and CE routers.
   On the PE router.

```
Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 192.0.2.14
Router-PE1(config-bgp)#address-family ipv6 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#address-family vpnv6 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 ipv6 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 2002:1::3
Router-PE1(config-bgp-vrf-nbr)#remote-as 7501
Router-PE1(config-bgp-vrf-nbr)#address-family ipv6 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
```

   On the CE router.

```
Router-CE1#configure
Router-CE1(config)#router bgp 2001
Router-CE1(config-bgp)#bgp router-id 192.0.2.15
Router-CE1(config-bgp)#address-family ipv6 unicast
Router-CE1(config-bgp-af)#exit
```

```
Router-CE1(config-bgp)#address-family vpnv6 unicast
Router-CE1(config-bgp-af)#exit
Router-CE1(config-bgp)#neighbor  2001:1::1
Router-CE1(config-bgp-nbr)#remote-as 2001
Router-CE1(config-bgp-nbr)#address-family ipv6 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
```

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:

CE1:

2. Review the running configuration.
   On PE1.

```
router bgp 2001
 bgp router-id 192.0.2.14
 address-family ipv6 unicast
 !
 address-family vpnv6 unicast
 !
 vrf vrf1601
  rd 2001:1601
  address-family ipv6 unicast
   label mode per-vrf
   redistribute connected
  !
  neighbor 2002:1::3
   remote-as 7501
   address-family ipv6 unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
 !
```

On CE1.

```
router bgp 7501
 bgp router-id 192.0.2.15
 address-family ipv6 unicast
 !
 address-family vpnv6 unicast
 !
 neighbor 2002:1::1
 remote-as 2001
  address-family ipv6 unicast
   route-policy pass-all in
   route-policy pass-all out
 !
!
```

3. Use the **show bgp neighbor** command to verify the BGP neighbor status on both routers.

- PE1:

```
Router-PE1# show bgp neighbor
BGP neighbor is 2002:1::3
 Remote AS 6553700, local AS 2001, external link
 Administratively shut down
 Remote router ID 2002:1::2
  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: IPv6 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 VPNv6 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: 2002:1::3, Local port: 23456, IF Handle: 0x00000000
  Foreign host: 2002:1::1, 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 2001:1::1
 Remote AS 2001, local AS 6553700, external link
 Remote router ID 2002:1::1
  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: IPv6 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: 2002:1::1, Local port: 179, IF Handle: 0x00000000
Foreign host: 2001:1::3, Foreign port: 23456
Last reset 00:00:00
External BGP neighbor not directly connected.
```