



## **L3VPN Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 25.1.x, 25.2.x**

**First Published:** 2025-03-31

**Last Modified:** 2025-06-01

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2025 Cisco Systems, Inc. All rights reserved.



## CONTENTS

### Full Cisco Trademarks with Software License ?

---

#### PREFACE

##### Preface vii

Changes to This Document vii

Obtaining Documentation and Submitting a Service Request vii

---

#### CHAPTER 1

##### New and Changed Feature Information 1

New and Changed L3VPN Features 1

---

#### CHAPTER 2

##### YANG Data Models for L3VPN Features 3

Using YANG Data Models 3

---

#### CHAPTER 3

##### Implementing MPLS Layer 3 VPNs 5

MPLS Layer 3 VPN Overview 5

MPLS L3VPN Benefits 7

Major Components of MPLS L3VPN—Details 8

Virtual Routing and Forwarding Tables 9

VPN Routing Information: Distribution 10

BGP Distribution of VPN Routing Information 10

MPLS Forwarding 10

Automatic Route Distinguisher Assignment 11

Prerequisites for Implementing MPLS L3VPN 11

Restrictions for MPLS L3VPN 11

Configure the Core Network 12

Verify MPLS L3VPN Configuration 18

L3VPN over RSVP-TE 22

VRF-lite	23
Configure VRF-lite	23
MPLS L3VPN Services using Segment Routing	26
Configure MPLS L3VPN over Segment Routing	26
Configure Segment Routing in MPLS Core	27
Verify MPLS L3VPN Configuration over Segment Routing	30
Inter-AS Support for L3VPN	32
Inter-AS and ASBRs	33
MPLS VPN Inter-AS BGP Label Distribution	33
Exchanging IPv4 Routes with MPLS labels	34
Provide VPN Connectivity Across Multiple Autonomous Systems with MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels	36
Configure the Route Reflectors to Exchange VPN-IPv4 Routes	36
Configure the Route Reflectors to Reflect Remote Routes in its AS	38
Provide VPN Connectivity Across Multiple Autonomous Systems with MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses	39
Configure a Static Route to an ASBR Peer	39
Inter-AS Option B for L3VPN	40
Configure Inter-AS Option B for L3VPN	42
Carrier Supporting Carrier for L3VPN	62
Customer Carrier: ISP with MPLS Core	63
Customer Carrier: MPLS Service Provider	64
CSC Benefits	65
Configure Carrier Supporting Carrier for L3VPN	65

---

**CHAPTER 4**

<b>VXLAN Static Routing</b>	<b>69</b>
VXLAN Static Routing	69
Key Concepts	74
Configure VXLAN Static Routing	75

---

**CHAPTER 5**

<b>Implementing IPv6 VPN Provider Edge Transport over MPLS</b>	<b>81</b>
Overview of 6PE/6VPE	82
Benefits of 6PE/6VPE	83
Deploying IPv6 over MPLS Backbones	83

IPv6 on the Provider Edge and Customer Edge Routers	83
OSPFv3 (CE to PE)	84
Configuring 6PE/6VPE	85
Configuring OSPFv3 as the Routing Protocol Between the PE and CE Routers	89
Configure BGP as the Routing Protocol Between the PE and CE Routers	90





## Preface

This guide describes the Cisco 8000 Series Router configurations. The preface for the L3VPN Configuration Guide for Cisco 8000 Series Routers contains these sections:

- [Changes to This Document, on page vii](#)
- [Obtaining Documentation and Submitting a Service Request, on page vii](#)

## Changes to This Document

This table lists the technical changes made to this document since it was first released.

**Table 1: Changes to This Document**

Date	Summary
March 2025	Initial release of this document.

## Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly What's New in Cisco Product Documentation, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the What's New in Cisco Product Documentation as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.







# CHAPTER 1

## New and Changed Feature Information

This table summarizes the new and changed feature information for the L3VPN Configuration Guide for Cisco 8000 Series Routers, and tells you where they are documented.

- [New and Changed L3VPN Features, on page 1](#)

## New and Changed L3VPN Features

*Table 2: L3VPN Features Added or Modified in IOS XR Release 25.x.x*

Feature	Description	Changed in Release	Where Documented
None	No new features introduced	Not available	Not available





## CHAPTER 2

# YANG Data Models for L3VPN Features

---

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

- [Using YANG Data Models, on page 3](#)

## Using 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*.





## CHAPTER 3

# Implementing MPLS Layer 3 VPNs

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

This module provides the conceptual and configuration information for MPLS Layer 3 VPNs on router.

This chapter includes topics on:

- [MPLS Layer 3 VPN Overview, on page 5](#)
- [Configure the Core Network, on page 12](#)
- [Verify MPLS L3VPN Configuration, on page 18](#)
- [L3VPN over RSVP-TE, on page 22](#)
- [VRF-lite, on page 23](#)
- [MPLS L3VPN Services using Segment Routing, on page 26](#)
- [Inter-AS Support for L3VPN, on page 32](#)
- [Inter-AS Option B for L3VPN, on page 40](#)
- [Carrier Supporting Carrier for L3VPN, on page 62](#)
- [CSC Benefits, on page 65](#)
- [Configure Carrier Supporting Carrier for L3VPN, on page 65](#)

## MPLS Layer 3 VPN Overview

*Table 3: Feature History Table*

Feature Name	Release Information	Feature Description
MPLS Layer 3 VPN	Release 25.1.1	Introduced in this release on: Fixed Systems ( 8010 [ASIC: A100])  This feature is now supported on: <ul style="list-style-type: none"><li>• 8011-4G24Y4H-I</li></ul>
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	<p>Introduced in this release on: Fixed Systems (8200, 8700); Modular Systems (8800 [LC ASIC: P100]) (select variants only*)</p> <p>* The MPLS Layer 3 VPN functionality is now extended to:</p> <ul style="list-style-type: none"> <li>• 8212-48FH-M</li> <li>• 8711-32FH-M</li> <li>• 88-LC1-52Y8H-EM</li> <li>• 88-LC1-12TH24FH-E</li> </ul>
MPLS Layer 3 VPN	Release 24.2.11	<p>Introduced in this release on: Modular Systems (8800 [LC ASIC: P100]) (select variants only*)</p> <p>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.</p> <p>* This functionality is now extended to routers with the 88-LC1-36EH line cards.</p>

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

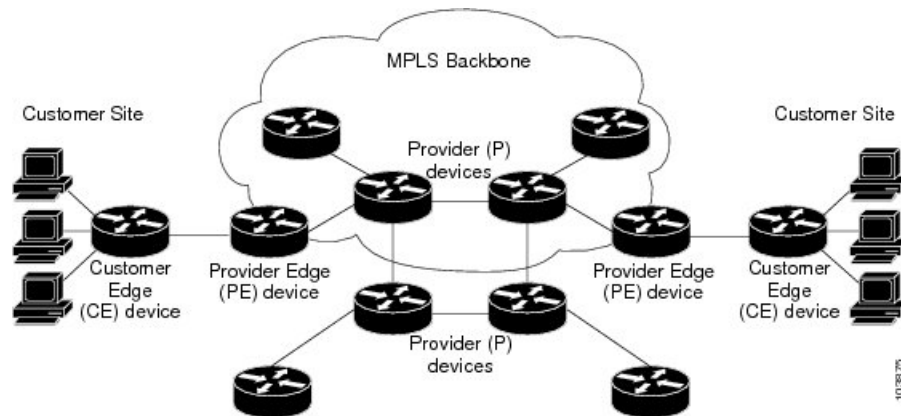
- An IP-based network delivering private network services over a public infrastructure
- A set of sites that are allowed to communicate with each other privately over the Internet or other public or private networks

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

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

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

The following figure depicts a basic MPLS VPN topology.

**Figure 1: Basic MPLS VPN Topology**

These are the basic components of MPLS VPN:

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

## MPLS L3VPN Benefits

MPLS L3VPN provides the following benefits:

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

## Major Components of MPLS L3VPN—Details

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

*Table 4: Feature History Table*

Feature Name	Release	Description
Virtual Routing and Forwarding Tables	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC:K100])(select variants only*).</p> <p>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.</p> <p>*Previously this feature was supported on Q200 and Q100. It is now extended to Cisco 8712-MOD-M routers.</p>

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

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

These components are collectively called a VRF instance.

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

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

## VPN Routing Information: Distribution

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

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

## BGP Distribution of VPN Routing Information

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

- A CE router by static configuration
- An eBGP session with the CE router
- Open Shortest Path First (OSPF) as Interior Gateway Protocol (IGP)

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

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

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

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

## MPLS Forwarding

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

A PE router binds a label to each customer prefix learned from a CE router and includes the label in the network reachability information for the prefix that it advertises to other PE routers. When a PE router forwards

a packet received from a CE router across the provider network, it labels the packet with the label learned from the destination PE router. When the destination PE router receives the labeled packet, it pops the label and uses it to direct the packet to the correct CE router. Label forwarding across the provider backbone is based on dynamic label switching. A customer data packet carries two levels of labels when traversing the backbone:

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

## Automatic Route Distinguisher Assignment

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

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

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

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

## Prerequisites for Implementing MPLS L3VPN

These are the prerequisites to configure MPLS L3VPN:

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

## Restrictions for MPLS L3VPN

Implementing MPLS L3VPN is subjected to these restrictions:

- Fragmentation of MPLS packets that exceed egress MTU is not supported. Fragmentation is not supported for IP->MPLS imposition as well. Hence, it is recommended to use Maximum MTU (9216) value on all interfaces in the MPLS core.
- L3VPN prefix lookup always yields a single path. In case of multiple paths at IGP or BGP level, path selection at each level is done using flow hash computed in data plane.
- Per VRF aggregate statistics are not supported.

### MPLS VPN Inter-AS with ASBRs exchanging IPv4 routes and MPLS labels

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

### Label assignments

- Local label allocation for every VRF on MPLS VPN.
- One VPN label for every VRF.
- Must have per VRF label mode across the VRF deployment.

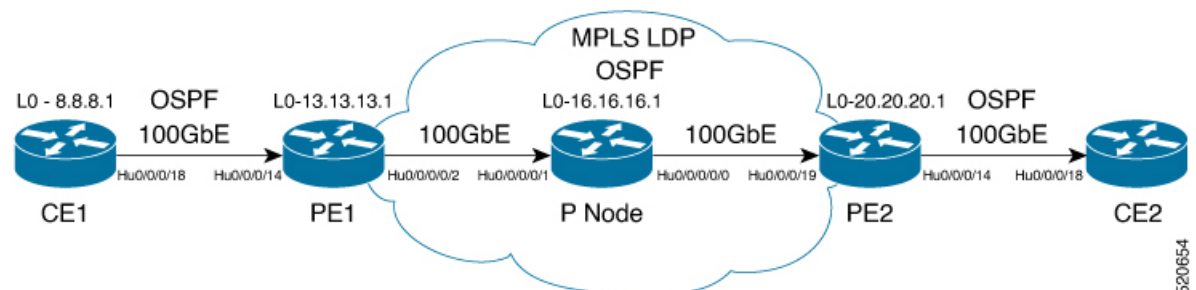
### L3 interface scale value on the Cisco modular router operating in P100 mode

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

## Configure the Core Network

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

**Figure 2: L3VPN over MPLS LDP**



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

### Assess the Needs of MPLS VPN Customers

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

- Identify the size of the network:

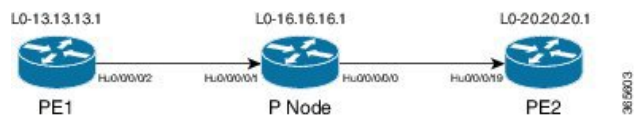
Identify the following to determine the number of routers and ports required:

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

### Configure Routing Protocols in the Core

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

**Figure 3: OSPF as Routing Protocol in the Core**



### Configuration Example

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

```

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

### Running Configuration

```

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

!
!
  
```

!

## Verification

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

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

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

Total neighbor count: 1
```

## Configure MPLS in the Core

You can also transport MPLS L3VPN services using segment routing in the core. For details, see .

## Configuration Example

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

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

Repeat this configuration in PE2 and P routers as well.

## Running Configuration

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

!

## Verification

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

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
  TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
  LDP Discovery Sources:
    IPv4: (1)
      HundredGigE0/0/0/2
```

```

IPv6: (0)
Addresses bound to this peer:
IPv4: (6)
    10.64.98.32      87.0.0.2      88.88.88.14    50.50.50.50
    178.0.0.1       192.168.0.1
IPv6: (0)

```

### Determine if FIB is Enabled in the Core

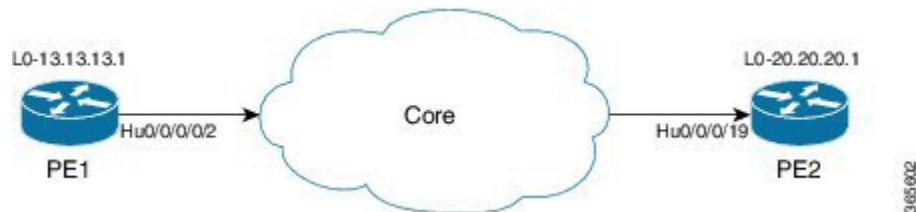
Forwarding Information Base (FIB) must be enabled on all routers in the core, including the provider edge (PE) routers. For information on how to determine if FIB is enabled, see the *Implementing Cisco Express Forwarding module in the IP Addresses and Services Configuration Guide for Cisco 8000 Series Routers*.

### Configure Multiprotocol BGP on the PE Routers and Route Reflectors

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

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

**Figure 4: Multiprotocol BGP on PE Routers**



### Configuration Example

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

```

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

## Running Configuration

```
router bgp 2001
  bgp router-id 10.0.0.1
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  neighbor 172.16.0.1
    remote-as 2001
    update-source Loopback0
    address-family vpnv4 unicast
    !
    address-family ipv4 unicast
    !
  !
  vrf vrf1601
    rd 2001:1601
    address-family ipv4 unicast
      label mode per-vrf
      redistribute connected
    !
  !
```

## Verification

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

```
Router-PE1#show bgp neighbor
```

```
BGP neighbor is 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

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

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

```

## Verify MPLS L3VPN Configuration

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

### Verify the L3VPN Traffic Flow

#### P node:

Router-P#**show mpls forwarding**

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24119	Pop	20.20.20.1/32	Hu0/0/0/0	191.31.1.90	2170204180148

#### PE2:

Router#**show mpls forwarding**

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24031	Aggregate	vrf1601: Per-VRF Aggr[V]	\	vrf1601	0

**Verify the Underlay (transport)**

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

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
  TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
  LDP Discovery Sources:
    IPv4: (1)
      TenGigE0/0/0/2
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.64.98.32      87.0.0.2      88.88.88.14      50.50.50.50
      178.0.0.1       192.1.1.1
    IPv6: (0)
```

Verify if the label update is received by the FIB:

```
Router-PE1#show mpls forwarding
Local  Outgoing  Prefix      Outgoing    Next Hop    Bytes
Label  Label      or ID       Interface   Interface   Switched
-----
24036  Pop        16.16.16.1/32  Hu0/0/0/2   191.22.1.2   293294
24037  24165     18.18.18.1/32  Hu0/0/0/2   191.22.1.2   500
24039  24167     20.20.20.1/32  Hu0/0/0/2   191.22.1.2   17872433
24039  24167     20.20.20.1/32  Hu0/0/0/2.1 191.22.3.2   6345
24041  Aggregate vrf1601: Per-VRF Aggr[V] \
                                vrf1601      0
```

Verify if label is updated in the hardware:

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

```
Local  Outgoing  Prefix      Outgoing    Next Hop    Bytes
Label  Label      or ID       Interface   Interface   Switched
-----
24039  24167     20.20.20.1/32  Hu0/0/0/2   191.22.1.2   N/A
24167  20.20.20.1/32  Hu0/0/0/2.1 191.22.3.2   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 the Overlay (L3VPN)

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

Router-PE1#**show bgp summary**

BGP router identifier 13.13.13.1, local AS number 2001

BGP generic scan interval 60 secs

Non-stop routing is enabled

BGP table state: Active

Table ID: 0xe0000000 RD version: 18003

BGP main routing table version 18003

BGP NSR Initial initsync version 3 (Reached)

BGP NSR/ISSU Sync-Group versions 0/0

BGP scan interval 60 secs

BGP is operating in STANDALONE mode.

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

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

Verify if BGP routes are advertised and learnt:

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

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
Route Distinguisher: 2001:1601 (default for vrf vrf1601)
*> 20.13.1.1/32    192.13.26.5                        0 7501 i
*> 20.13.1.2/32    192.13.26.5                        0 7501 i
*>i20.23.1.1/32    20.20.20.1                        100 0 6553700 11501 i
*>i20.23.1.2/32    20.20.20.1                        100 0 6553700 11501 i
```

Verify BGP labels:

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

Verify if the route is downloaded in the respective VRF:

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

**Disposition Path**

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

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

```

1/1: PopLkup-v4, 'vrf1601':4U, ipv4
24032, (VPN-VRF, 'vrf1602':4U), 1 Paths
1/1: PopLkup-v4, 'vrf1602':4U, ipv4

```

Verify if the label update is received by the FIB:

Router-PE2#**show mpls forwarding**

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24019	Pop	18.18.18.3/32	Hu0/0/0/19	191.31.1.89	11151725032
24030	24155	13.13.13.1/32	Hu0/0/0/19	191.31.1.89	3639895
24031	Aggregate	vrf1601: Per-VRF Aggr[V] \	vrf1601		0

## L3VPN over RSVP-TE

**Table 5: Feature History Table**

Feature Name	Release Information	Feature Description
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.

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

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

Perform these steps to configure L3VPN over RSVP-TE:

- Configure routing protocols in the core—To configure routing protocols in the core, see the *Routing Configuration Guide for Cisco 8000 Series Routers*.
- Enable MPLS on all routers in the core—To enable MPLS on all routers in the core, you must configure a Label Distribution Protocol (LDP). You can use either of the following as an LDP:
  - 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.

## VRF-lite

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

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



**Note** For VRF-lite using the BGP protocol, change the label allocation mode to per-VRF.

## Configure VRF-lite

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

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

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

### Configuration Example

- **Create VRF:**

```
Router#configure
Router(config)#vrf vrf1
Router(config-vrf)#address-family ipv4 unicast
/* You must create route-policy pass-all before this configuration */
Router(config-vrf-af)#import from default-vrf route-policy pass-all
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#100:100
Router(config-vrf-import-rt)#exit
Router(config-vrf-af)#export route-target
Router(config-vrf-import-rt)#100:100
Router(config-vrf-import-rt)#exit
Router(config-vrf-import-rt)#commit
```

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

- **Configure VRF under the interface:**

```
Router#configure
Router(config-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
```

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

- **Configure VRF under routing protocol:**

```
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
```

## 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
!

```

## Verification

```

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

```

## MPLS L3VPN Services using Segment Routing

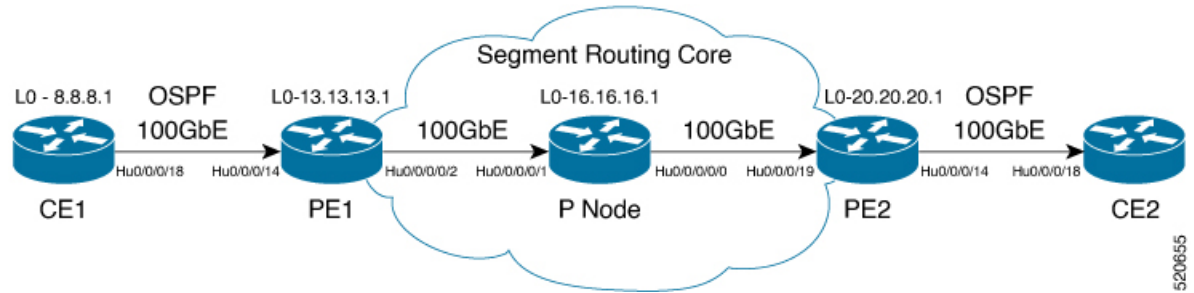
Currently, MPLS Label Distribution Protocol (LDP) is the widely used transport for MPLS L3VPN services. The user can achieve better resilience and convergence for the network traffic, by transporting MPLS L3VPN services using Segment Routing (SR), instead of MPLS LDP. Segment routing can be directly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

## Configure MPLS L3VPN over Segment Routing

### Topology

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

Figure 5: MPLS L3VPN over Segment Routing



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

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

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

### Configuration

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

- Configure Segment Routing in MPLS Core
- Configure protocol support on PE-CE (see Connect MPLS VPN Customers)
- Configure protocol support on PE-PE (see Configure Multiprotocol BGP on the PE Routers and Route Reflectors)

## Configure Segment Routing in MPLS Core

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

### Configuration Example

```
/* Configure Segment Routing using OSPF */

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

```

Router-PE1(config-ospf-ar-if)#exit
Router-PE1(config-ospf-ar)#interface Loopback0
Router-PE1(config-ospf-ar-if)#prefix-sid index 1
Router-PE1(config-ospf-ar-if)#commit

/* Configure segment routing global block */

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

/* Configure Segment Routing using ISIS */

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

```

## Running Configuration

### PE1:

```

router ospf dc-sr
router-id 13.13.13.1
segment-routing mpls
segment-routing forwarding mpls
mpls ldp sync
mpls ldp auto-config
segment-routing 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 16.16.16.1
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 20.20.20.1
  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
!

```

## Verify MPLS L3VPN Configuration over Segment Routing

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

**P node:**

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

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
-----	-----	-----	-----	-----	-----

```
64003 Pop SR Pfx (idx 0) Hu0/0/0/0 193.16.1.2 572842
```

- Verify the statistics in PE1 router:

**PE1:**

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

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

**PE2:**

```
Router-PE2#show mpls forwarding
Local  Outgoing  Prefix          Outgoing  Next Hop        Bytes
Label  Label      or ID          Interface                Switched
-----
24031  Aggregate  vrf1601: Per-VRF Aggr[V] \
                                         vrf1601                0
```

# Inter-AS Support for L3VPN

*Table 6: Feature History Table*

Feature Name	Release	Description
Inter-AS Support for L3VPN	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC:K100])(select variants only*).</p> <p>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.</p> <p>*Previously this feature was supported on Q200 and Q100. It is now extended to Cisco 8712-MOD-M routers.</p>

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

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

## Benefits

An MPLS VPN Inter-AS provides the following benefits:

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

Service providers, running separate autonomous systems, can jointly offer MPLS VPN services to the same end customer. A VPN can begin at one customer site and traverse different VPN service provider



backbones before arriving at another site of the same customer. Previously, MPLS VPN could traverse only a single BGP autonomous system service provider backbone. This feature lets multiple autonomous systems form a continuous, seamless network between customer sites of a service provider.

- Allows a VPN to exist in different areas.

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

## Inter-AS and ASBRs

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

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

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

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

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



---

**Note** Inter-AS options A and C are supported and Inter AS option B is not supported.

---

## MPLS VPN Inter-AS BGP Label Distribution



---

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

---

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

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

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

## Exchanging IPv4 Routes with MPLS labels

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

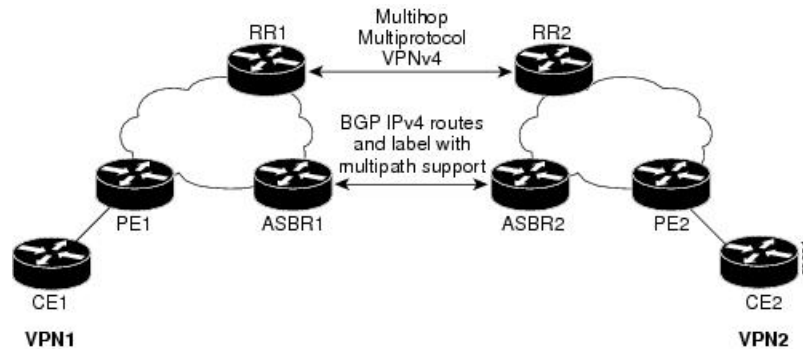
- Route reflectors exchange VPN-IPv4 routes by using multihop, multiprotocol eBGP. This configuration also preserves the next-hop information and the VPN labels across the autonomous systems.
- A local PE router (for example, PE1 in the figure below) needs to know the routes and label information for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

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

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

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

**Figure 6: VPNs Using eBGP and iBGP to Distribute Routes and MPLS Labels**



### BGP Routing Information

BGP routing information includes the following items:

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

### BGP Messages and MPLS Labels

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

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

### Sending MPLS Labels with Routes

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

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

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

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



**Note** This section contains instructions for the following tasks:

- Configuring the Route Reflectors to Exchange VPN-IPv4 Routes
- Configure the Route Reflectors to Reflect Remote Routes in its AS

### Configure the Route Reflectors to Exchange VPN-IPv4 Routes

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

#### Configuration Example

```
Router# configure
Router(config)# router bgp 500
Router(config-bgp)#
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
```

#### Running Configuration

```
Router#show run router bgp 500
router bgp 500
  bgp router-id 60.200.13.1
  address-family ipv4 labeled-unicast
    allocate-label all
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  neighbor 10.200.1.1
    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
!

```

## Verification

```

Router#show cef vrf vrf2001 ipv4 172.16.0.1/32 hardware egress location0/0/CPU0
111.1.1.2/32, version 39765, internal 0x5000001 0x0 (ptr 0x9f4d326c) [1], 0x0 (0xa0263058),
0x808 (0x899285b8)
Updated Oct 27 10:58:39.350
Prefix Len 32, traffic index 0, precedence n/a, priority 3
  via 10.200.1.1/32, 307 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0x89a59100 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 10.200.1.1/32 via 69263/0/21
      next hop 63.13.1.1/32 Te0/3/0/17/0 labels imposed {24007 64007 64023}

```

```

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

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

```

```

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

```

```

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

```

```

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

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

```

## Configure the Route Reflectors to Reflect Remote Routes in its AS

```

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

```

## Configure the Route Reflectors to Reflect Remote Routes in its AS

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

### Configuration Example

```

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

```

```

Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#route-reflector-client
Router(config-bgp-nbr)#address-family vpnv4 unicast
Router(config-bgp-nbr-af)#route-reflector-client

```

### Running Configuration

```

Router#show run router bgp 500
router bgp 500
  bgp router-id 60.200.13.1
  address-family ipv4 unicast
    allocate-label all
  !
  address-family vpnv4 unicast
  !
  neighbor 60.200.11.1
    remote-as 500
    update-source Loopback0
  !
  address-family ipv6 labeled-unicast
    route-reflector-client
  !
  address-family vpnv6 unicast
  !
  !
  neighbor 60.200.12.1
    remote-as 500
    update-source Loopback0
    address-family ipv4 labeled-unicast
      route-reflector-client
    !
    address-family vpnv4 unicast
      route-reflector-client
    !

```

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

This section contains instructions for the following task.

- Configure a Static Route to an ASBR Peer

### Configure a Static Route to an ASBR Peer

To configure a static route to an ASBR peer:

#### Configuration Example

```

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

```

# Inter-AS Option B for L3VPN

*Table 7: Feature History Table*

Feature Name	Release Information	Feature Description
Inter-AS Option B for L3VPN	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC:K100])  This feature support is now extended to the Cisco 8712-MOD-M routers.
Inter-AS Option B for L3VPN	Release 7.9.1	You can now provide inter-autonomous system (AS) support for L3VPN networks with two or more autonomous systems (ASes) that are using Border Gateway Protocol Multicast Virtual Private Network (BGP-MVPN). This provision allows administrators to control their ASes independently and facilitate efficient scalability by allowing ASes equipped with Autonomous System Boundary Routers (ASBRs) administered by different entities to communicate through a single BGP session, exchanging VPN-IP addresses, IP routes, and MPLS labels.

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

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

## Functions of Inter-AS Option B

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

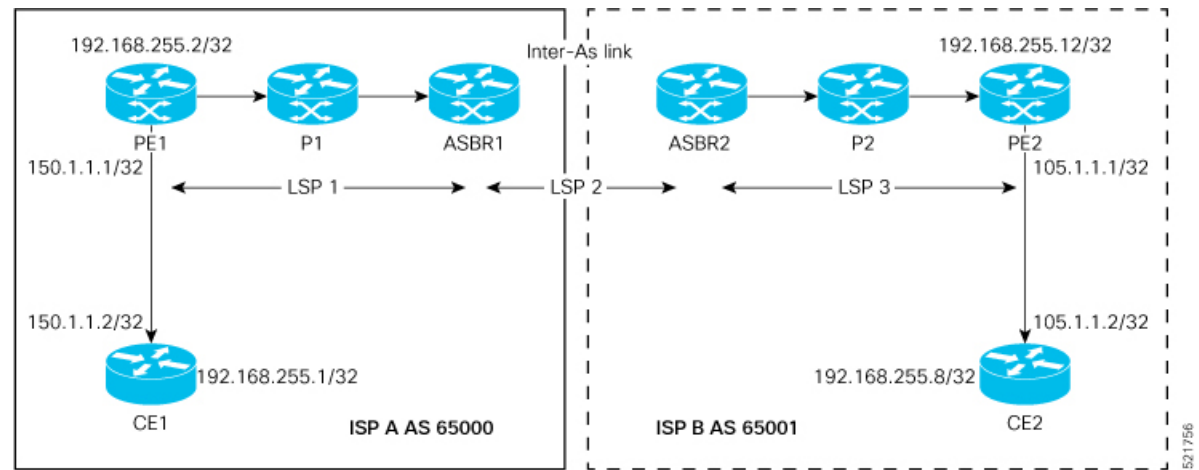
The label mapping information for a particular route is added to the same BGP update message that is used to distribute the route itself.

- When BGP is used to distribute a particular route, it also distributes an MPLS label which is mapped to that route. Many ISPs prefer this method of configuration since it ensures complete IGP isolation between different sites.



## Topology

Figure 7: L3VPN Inter-AS Option B



In this topology:

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



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

## Configure Inter-AS Option B for L3VPN

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



**Note** Though L3VPN Inter-AS Option B support per prefix mode, we recommend you use per-nexthop-label mode due to resource limitations.

### Configuration Example

#### PE1 Configuration

```
Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 192.168.255.2/32
Router(config-if)# ipv6 address 50:50:50::50/128
Router(config-if)# exit
Router(config)# interface Bundle-Ether25
Router(config-if)# description interface to R4
Router(config-if)# ipv4 address 172.16.0.1 255.240.0.0
Router(config-if)# exit
Router(config)# router isis access
Router(config-isis)# is-type level-1
Router(config-isis)# net 49.0001.0000.0000.0050.00
Router(config-isis)# nsr
Router(config-isis)# nsf ietf
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether25
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# interface Bundle-Ether25
Router(config-ldp-if)# exit
Router(config-ldp)# exit
Router(config)# vrf vrf1
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target 100:1
Router(config-vrf-af)# export route-target 100:1
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# import route-target 100:1
Router(config-vrf-af)# export route-target 100:1
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# router bgp 65000
```

```

Router(config-bgp)# nsr
Router(config-bgp)# bgp router-id 192.168.255.2
Router(config-bgp)# bgp graceful-restart
Router(config-bgp)# address-family vpnv4 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# address-family vpnv6 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 10.10.10.10
Router(config-bgp-nbr)# remote-as 65000
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# vrf vrf1
Router(config-bgp-vrf)# rd 100:1
Router(config-bgp-vrf)# bgp router-id 192.168.255.2
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# address-family ipv6 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# neighbor 150.1.1.2
Router(config-bgp-vrf-nbr)# remote-as 501
Router(config-bgp-vrf-nbr)# address-family ipv4 unicast
Router(config-bgp-vrf-nbr-af)# route-policy pass-all in
Router(config-bgp-vrf-nbr-af)# route-policy pass-all out
Router(config-bgp-vrf-nbr-af)# exit
Router(config-bgp-vrf-nbr)# exit
Router(config-bgp-vrf)# neighbor 150:1:1::2
Router(config-bgp-vrf-nbr)# remote-as 501
Router(config-bgp-vrf-nbr)# address-family ipv6 unicast
Router(config-bgp-vrf-nbr-af)# route-policy pass-all in
Router(config-bgp-vrf-nbr-af)# route-policy pass-all out
Router(config-bgp-vrf-nbr-af)# commit

```

## P1 Configuration

```

Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 172.16.0.1 255.255.255.255
Router(config-if)# exit
Router(config)# interface Bundle-Ether12
Router(config-if)# description interface to ASBR1
Router(config-if)# ipv4 address 10.20.1.2 255.240.0.0
Router(config-if)# exit
Router(config)# interface Bundle-Ether25
Router(config-if)# description interface to R50
Router(config-if)# ipv4 address 20.50.1.1 255.240.0.0
Router(config-if)# exit
Router(config)# router isis core
Router(config-isis)# is-type level-1
Router(config-isis)# net 49.0001.0000.0000.0020.00
Router(config-isis)# nsr
Router(config-isis)# nsf ietf
Router(config-isis)# log adjacency changes

```

```

Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether12
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Bundle-Ether25
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# nsr
Router(config-ldp)# interface Bundle-Ether12
Router(config-ldp-if)# exit
Router(config-ldp)# interface Bundle-Ether25
Router(config-ldp-if)# commit

```

### ASBR1 Configuration

```

Router# configure
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 10.10.10.10 255.255.255.255
Router(config-if)# ipv6 address 10:10:10::10/128
Router(config-if)# exit
Router(config)# interface Bundle-Ether12
Router(config-if)# description interface to 172.16.0.1
Router(config-if)# ipv4 address 10.20.1.1 255.240.0.0
Router(config-if)# monitor-session Test ethernet port-level
Router(config-if-mon)# exit
Router(config-if)# exit
Router(config)# interface Bundle-Ether11
Router(config-if)# description interface to ASBR2
Router(config-if)# ipv4 address 10.1.0.1 255.255.255.0
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

```

## ASBR2 Configuration

```

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::1/128
Router(config-if)# exit
Router(config)# interface Bundle-Ether12
Router(config-if)# description interface to P2
Router(config-if)# ipv4 address 1.2.1.1 255.240.0.0

```

```

Router(config-if)# monitor-session Test ethernet port-level
Router(config-if-mon)# exit
Router(config-if)# exit
Router(config)# interface Bundle-Ether11
Router(config-if)# description interface to ASBR1
Router(config-if)# ipv4 address 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 5.5.5.5
Router(config-bgp-nbr)# remote-as 65001
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# neighbor 10.10.10.10
Router(config-bgp-nbr)# remote-as 65000
Router(config-bgp-nbr)# ebgp-multihop 2
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# route-policy pass-all out

```

```

Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# address-family vpnv6 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# maximum-prefix 4500000 90
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# exit
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 10.10.10.10/32 Bundle-Ether11 10.1.0.1
Router(config-static-afi)# commit

```

## P2 Configuration

```

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

```

## PE2 Configuration

```

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

```



```

Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# exit
Router(config-isis)# interface Bundle-Ether25
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit
Router(config-isis)# exit
Router(config)# mpls ldp
Router(config-ldp)# interface Bundle-Ether25
Router(config-ldp-if)# exit
Router(config-ldp)# exit
Router(config)# vrf vrf1
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target 100:1
Router(config-vrf-af)# export route-target 100:1
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# import route-target 100:1
Router(config-vrf-af)# export route-target 100:1
Router(config-vrf-af)# commit

```

### Running Configuration

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

The following is the running configuration of PE1:

```

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

```

```

!
!
mpls ldp
interface Bundle-Ether25
!
!

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

```

!  
!

The following is the running configuration of P1:

```
interface Loopback0
  ipv4 address 172.16.0.1 255.255.255.255
!
interface Bundle-Ether12
  description interface to ASBR1
  ipv4 address 10.20.1.2 255.240.0.0
!
interface Bundle-Ether25
  description interface to R50
  ipv4 address 20.50.1.1 255.240.0.0
!
router isis core
  is-type level-1
  net 49.0001.0000.0000.0020.00
  nsr
  nsf ietf
  log adjacency changes
  address-family ipv4 unicast
  metric-style wide
!
interface Bundle-Ether12
  point-to-point
  address-family ipv4 unicast
!
!
interface Bundle-Ether25
  point-to-point
  address-family ipv4 unicast
!
!
interface Loopback0
  point-to-point
  address-family ipv4 unicast
!
!
!

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

The following is the running configuration of ASBR1:

```
interface Loopback0
  ipv4 address 10.10.10.10 255.255.255.255
  ipv6 address 10:10:10::10/128
!
interface Bundle-Ether12
  description interface to 172.16.0.1
  ipv4 address 10.20.1.1 255.240.0.0
  monitor-session Test ethernet port-level
!
!
interface Bundle-Ether11
```

```

description interface to ASBR2
ipv4 address 10.1.0.1 255.255.255.0
!
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
!
!

```

The following is the running configuration of ASBR2:

```

interface Loopback0
  ipv4 address 10.0.0.1 255.255.255.255
  ipv6 address 1::1::1/128
!
interface Bundle-Ether12
  description interface to P2
  ipv4 address 1.2.1.1 255.240.0.0
!
interface Bundle-Ether11
  description interface to ASBR1
  ipv4 address 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 5.5.5.5
    remote-as 65001
    update-source Loopback0
    address-family vpnv4 unicast
      next-hop-self
    !
    address-family vpnv6 unicast
      next-hop-self
    !
  !
  neighbor 10.10.10.10
    remote-as 65000
    ebgp-multihop 2
    update-source Loopback0
    address-family vpnv4 unicast
      route-policy pass-all in
      maximum-prefix 4500000 90
      route-policy pass-all out
    !
    address-family vpnv6 unicast
      route-policy pass-all in
      maximum-prefix 4500000 90
      route-policy pass-all out
    !
  !

router static
  address-family ipv4 unicast
    10.10.10.10/32 Bundle-Ether11 10.1.0.1
  !

```

The following is the running configuration of P2:

```

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

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

```

```
point-to-point
address-family ipv4 unicast
!
!
interface Loopback0
point-to-point
address-family ipv4 unicast
!
!
!
mpls ldp
interface Bundle-Ether12
!
interface Bundle-Ether25
!
!
```

The following is the running configuration of PE2:

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

```

!
!

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

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

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

mpls ldp
 interface Bundle-Ether25
!
!

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

```

## Verification

Verification on PE1.

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



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

```
Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
        U - per-user static route, o - ODR, L - local, G - DAGR, l - LISP
        A - access/subscriber, a - Application route
        M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path
```

Gateway of last resort is not set

```
B   105.1.1.0/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:04:43
C   150.1.1.0/24 is directly connected, 01:14:27, TenGigE0/0/0/22/0.1
L   150.1.1.1/32 is directly connected, 01:14:27, TenGigE0/0/0/22/0.1
B   202.1.0.0/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:00:08
B   202.1.1.0/24 [200/0] via 10.10.10.10 (nexthop in vrf default), 00:00:08
```

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

```
Router:PE1# show bgp vpnv4 unicast rd 100:1 202.1.0.0/24
```

```
Sun Jun  6 23:12:12.140 UTC
```

BGP routing table entry for 202.1.0.0/24, Route Distinguisher: 100:1

Versions:

Process	bRIB/RIB	SendTblVer
Speaker	2844	2844

Last Modified: Jun 6 23:08:30.194 for 00:03:42

Paths: (1 available, best #1)

Not advertised to any peer

Path #1: Received by speaker 0

Not advertised to any peer

200 501

10.10.10.10 (metric 30) from 10.10.10.10 (10.10.10.10)

**Received Label 24521**

Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate, imported

Received Path ID 0, Local Path ID 1, version 2844

Extended community: RT:100:1

Source AFI: VPNv4 Unicast, Source VRF: vrf1, Source Route Distinguisher: 100:1

```
Router:PE1# show cef vrf vrf1 202.1.0.0
```

```
Mon Jun  7 02:07:39.841 UTC
```

202.1.0.0/24, version 513583, internal 0x5000001 0x30 (ptr 0xa3f8bac8) [1], 0x0 (0x0), 0x208 (0x8f505928)

Updated Jun 7 01:50:33.710

Prefix Len 24, traffic index 0, precedence n/a, priority 3

gateway array (0x8f2d20e8) reference count 252, flags 0x2038, source rib (7), 0 backups  
[1 type 1 flags 0x48441 (0x8ad86708) ext 0x0 (0x0)]

LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]

gateway array update type-time 1 Jun 6 23:20:45.951

LDI Update time Jun 6 23:20:45.951

via 10.10.10.10/32, 5 dependencies, recursive [flags 0x6000]

path-idx 0 NHID 0x0 [0xa25ff9d8 0x0]

recursion-via-/32

next hop VRF - 'default', table - 0xe0000000

next hop 10.10.10.10/32 via 24003/0/21

next hop 20.50.1.1/32 BE25 labels imposed {24004 24521}

```
Load distribution: 0 (refcount 1)
```

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

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

```
Router:PE1# show mpls forwarding prefix 10.10.10.10/32
```

```
Mon Jun 7 02:06:40.845 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24003	24004	10.10.10.10/32	BE25	20.50.1.1	141107

```
Router:PE1# show cef vrf vrf1 202.1.0.0
```

```
Mon Jun 7 02:07:39.841 UTC
```

```
202.1.0.0/24, version 513583, internal 0x5000001 0x30 (ptr 0xa3f8bac8) [1], 0x0 (0x0), 0x208 (0x8f505928)
```

```
Updated Jun 7 01:50:33.710
```

```
Prefix Len 24, traffic index 0, precedence n/a, priority 3
```

```
gateway array (0x8f2d20e8) reference count 252, flags 0x2038, source rib (7), 0 backups [1 type 1 flags 0x48441 (0x8ad86708) ext 0x0 (0x0)]
```

```
LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
```

```
gateway array update type-time 1 Jun 6 23:20:45.951
```

```
LDI Update time Jun 6 23:20:45.951
```

```
via 10.10.10.10/32, 5 dependencies, recursive [flags 0x6000]
```

```
path-idx 0 NHID 0x0 [0xa25ff9d8 0x0]
```

```
recursion-via-/32
```

```
next hop VRF - 'default', table - 0xe0000000
```

```
next hop 10.10.10.10/32 via 24003/0/21
```

```
next hop 20.50.1.1/32 BE25 labels imposed {24004 24521}
```

```
Load distribution: 0 (refcount 1)
```

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

Verification on P1.

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

```
Router:P1#d show mpls forwarding prefix 10.10.10.10/32
```

```
Mon Jun 7 02:34:55.293 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24004	Pop	10.10.10.10/32	BE12	10.20.1.1	28804

Verification on ASBR1.

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

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

```

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

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

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

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

Processed 3 prefixes, 3 paths

Router:ASBR1# show mpls forwarding labels 24521
Sun Jun  6 23:05:49.323 EDT
Local  Outgoing  Prefix          Outgoing  Next Hop  Bytes
Label  Label      or ID          Interface
-----
24521  25516  100:1:202.1.0.0/24  10.1.0.1  0
-----

```

#### Verification on ASBR2.

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

```

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

Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.3
  Path #1: Received by speaker 0

```

```

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

```

```
Router:ASBR2# show bgp vpnv4 unicast rd 100:1 advertised neighbor 10.10.10.10 summary
```

```

Sun Jun  6 23:07:05.617 EDT
Network          Next Hop          From          AS Path
Route Distinguisher: 100:1
105.1.1.0/24      10.0.0.1          5.5.5.5       200?
150.1.1.0/24      10.0.0.1          10.10.10.10   200 100?
202.1.0.0/24      10.0.0.1          5.5.5.5       200 501i
202.1.1.0/24      10.0.0.1          5.5.5.5       200 501i

```

```
Processed 4 prefixes, 4 paths
```

```
Router:ASBR2# show mpls forwarding labels 25516
```

```

Sun Jun  6 23:07:32.394 EDT
Local  Outgoing  Prefix      Outgoing  Next Hop  Bytes
Label  Label        or ID      Interface
-----
25516  24002        No ID      5.5.5.5   654
-----

```

#### Verification on P2.

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

```
Router:P2# show mpls forwarding prefix 5.5.5.5/32
```

```

Mon Jun  7 03:09:11.532 UTC
Local  Outgoing  Prefix      Outgoing  Next Hop  Bytes
Label  Label        or ID      Interface
-----
24005  Pop          5.5.5.5/32  BE25      2.5.1.2   11921958
-----

```

#### Verification on PE2.

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

```
Router:PE2# show route vrf vrf1
```

```

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
U - per-user static route, o - ODR, L - local, G - DAGR, l - LISP
A - access/subscriber, a - Application route
M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

```

```
Gateway of last resort is not set
```

```

C    105.1.1.0/24 is directly connected, 6w4d, TenGigE0/0/0/30.1
L    105.1.1.1/32 is directly connected, 6w4d, TenGigE0/0/0/30.1
B    150.1.1.0/24 [200/0] via 10.0.0.1 (nexthop in vrf default), 03:58:16

```

```

B    160.1.0.0/24 [200/0] via 10.0.0.1 (nexthop in vrf default), 04:30:07
B    202.1.0.0/24 [20/0] via 105.1.1.2, 01:30:05
B    202.1.1.0/24 [20/0] via 105.1.1.2, 01:30:05

```

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

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

```
BGP routing table entry for 202.1.0.0/24, Route Distinguisher: 100:1
```

```
Versions:
```

```

Process          bRIB/RIB    SendTblVer
Speaker          1070062     1070062

```

```
Local Label: 24002
```

```
Last Modified: Jun  7 01:30:56.657 for 01:31:29
```

```
Paths: (1 available, best #1)
```

```
Advertised to peers (in unique update groups):
```

```
10.0.0.1
```

```
Path #1: Received by speaker 0
```

```
Advertised to peers (in unique update groups):
```

```
10.0.0.1
```

```
501
```

```
105.1.1.2 from 105.1.1.2 (194.0.0.1)
```

```
Origin IGP, localpref 100, valid, external, best, group-best, import-candidate
```

```
Received Path ID 0, Local Path ID 1, version 1070062
```

```
Extended community: RT:100:1
```

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

```
Router:PE2# show mpls forwarding labels 24002
```

```
Mon Jun  7 03:02:53.255 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24002	Aggregate	vrf1: Per-VRF Aggr[V]	\		
			vrf1		138

```
Router:PE2# show cef vrf vrf1 202.1.0.0
```

```
Mon Jun  7 03:04:08.268 UTC
```

```
202.1.0.0/24, version 3477, internal 0x1000001 0x30 (ptr 0x97f75328) [1], 0x0 (0x0), 0x0 (0x0)
```

```
Updated Jun  7 01:30:57.120
```

```
Prefix Len 24, traffic index 0, precedence n/a, priority 3
```

```
gateway array (0x8c820f38) reference count 2, flags 0x2010, source rib (7), 0 backups [1 type 3 flags 0x48441 (0x8a79cd88) ext 0x0 (0x0)]
```

```
LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
```

```
gateway array update type-time 1 Jun  7 01:30:57.120
```

```
LDI Update time Jun  7 01:30:57.120
```

```
Level 1 - Load distribution: 0
```

```
[0] via 105.1.1.2/32, recursive
```

```
via 105.1.1.2/32, 3 dependencies, recursive, bgp-ext [flags 0x6020]
```

```
path-idx 0 NHID 0x0 [0x8d575b80 0x0]
```

```
next hop 105.1.1.2/32 via 105.1.1.2/32
```

```
Load distribution: 0 (refcount 1)
```

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

## Carrier Supporting Carrier for L3VPN

**Table 8: Feature History Table**

Feature Name	Release Information	Feature Description
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.
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: <ul style="list-style-type: none"><li>• 8212-48FH-M</li><li>• 8711-32FH-M</li><li>• 88-LC1-52Y8H-EM</li><li>• 88-LC1-12TH24FH-E</li></ul>
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 Concepts

The following terminology is used in the context of Carrier Supporting Carrier (CSC):

*backbone carrier*—Service provider that provides the segment of the backbone network to the other provider. A backbone carrier offers BGP and MPLS VPN services.

*customer carrier*—Service provider that uses the segment of the backbone network. The customer carrier may be an Internet service provider (ISP) or a BGP/MPLS VPN service provider.

**CSC-CE router**—A customer edge router is part of a customer network and interfaces to a CSC provider edge (PE) router. In this document, the CSC-CE router sits on the edge of the customer carrier network.

**CSC-PE router**—A provider edge router is part of a service provider's network connected to a CSC customer edge (CE) router. In this document, the CSC-PE router sits on the edge of the backbone carrier network.

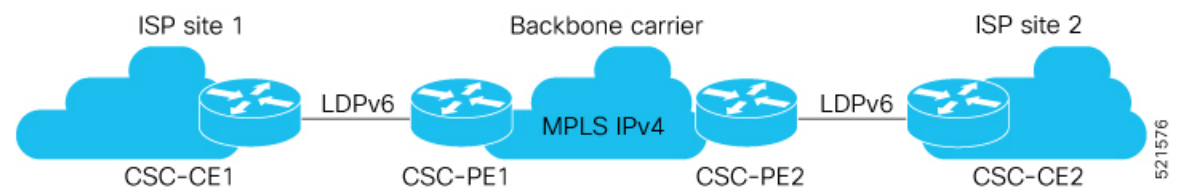
There can be two types of customer carriers:

- An Internet service provider (ISP)
- A BGP/MPLS VPN service provider

## Customer Carrier: ISP with MPLS Core

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.

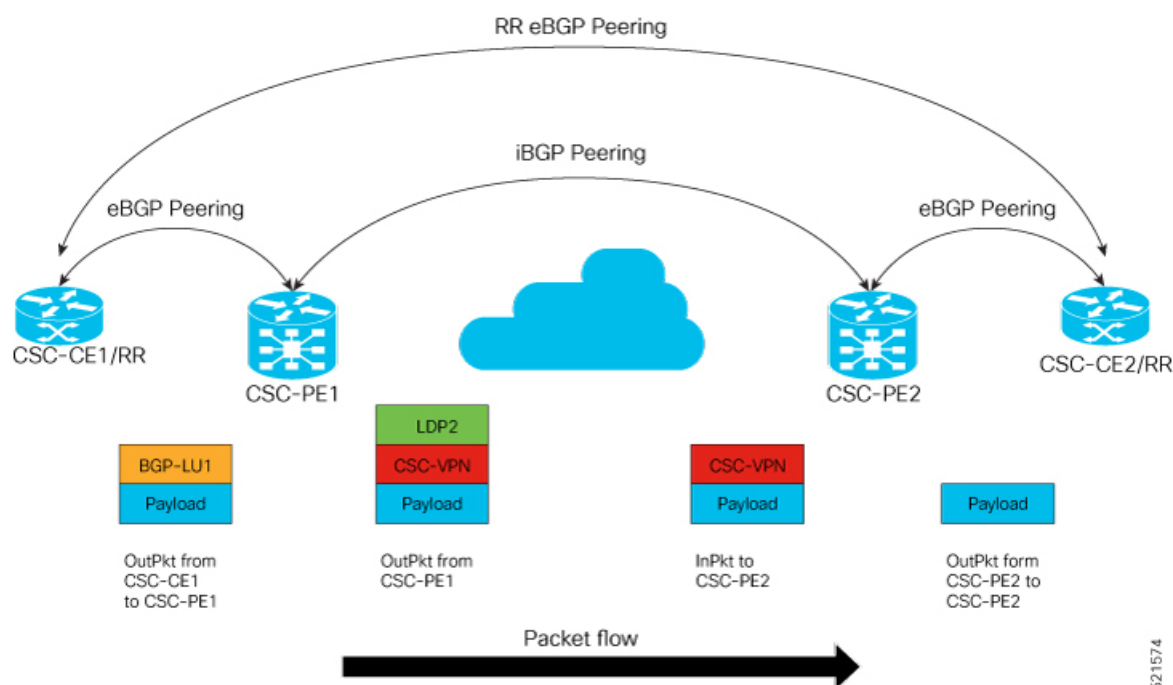
**Figure 8: Customer Carrier Is an ISP**



The links between the CE and PE routers use eBGP to distribute IPv4 routes and MPLS labels. Between the links, the PE routers use multiprotocol iBGP to distribute VPNv4 routes.

### MPLS Single-Label Packet Flow into CSC-CE (CSC-CE/RR eBGP Peering)

The following illustration shows how the packet flows into CSC-CE.

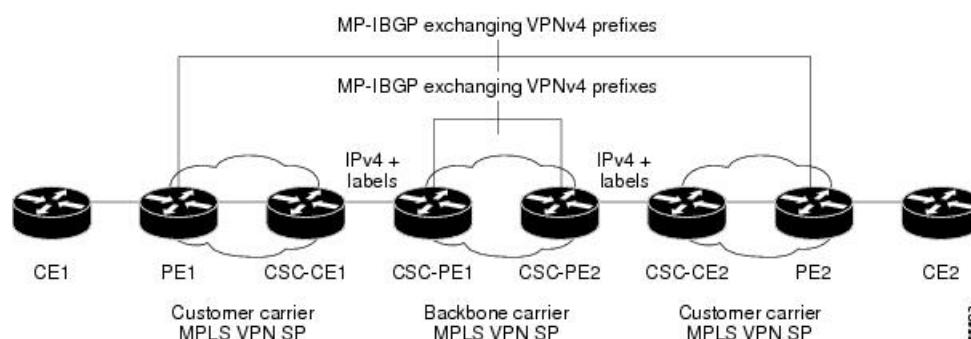


521574

## Customer Carrier: MPLS Service Provider

The following topology shows a network configuration where the backbone carrier and the customer carrier are BGP/MPLS VPN service providers. The customer carrier has two sites. The customer carrier uses MPLS in its network while the backbone carrier may use MPLS or IP tunnels in its network.

**Figure 9: Customer Carrier is an MPLS VPN Service Provider**

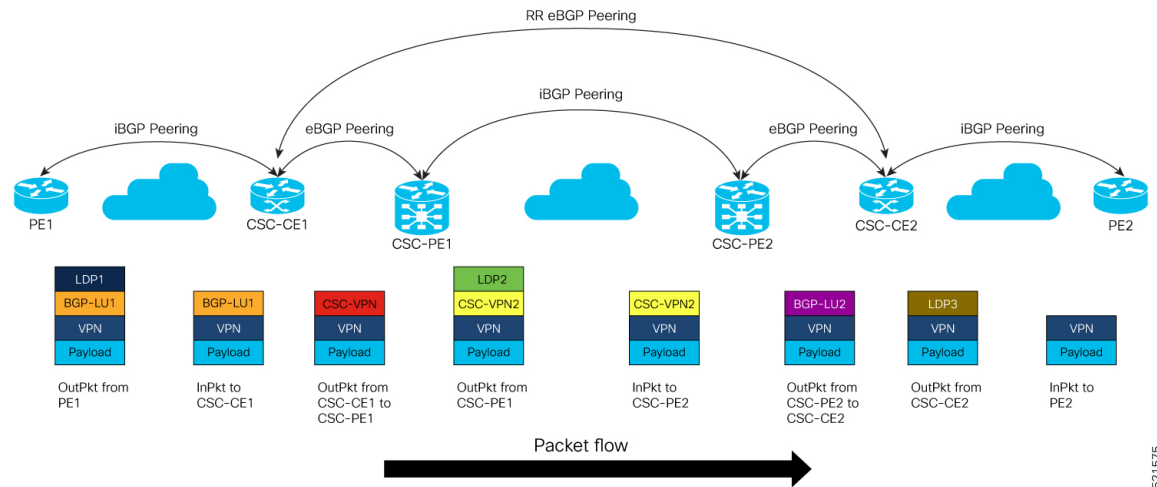


Customer Carrier is an MPLS VPN service provider, the customer carrier can run BGP-LU and LDP in its core network. In this case, the CSC-CE1 router in the customer carrier redistributes the eBGP routes it learns from the CSC-PE1 router of the backbone carrier to an IGP.

### MPLS Two-Labels Packet Flow into CSC-CE (CSC-CE/RR BGP Peering)

The following illustration shows how the packet flows into CSC-CE.





521575

## CSC Benefits

This section describes the benefits of CSC to the backbone carrier and customer carriers.

### Benefits to the Backbone Carrier

- The MPLS VPN carrier supporting carrier feature is scalable.
- The MPLS VPN carrier supporting carrier feature is a flexible solution.

### Benefits to the Customer Carriers

- Customer carriers who use the VPN services provided by the backbone carrier receive the same level of security that Frame Relay or ATM-based VPNs provide.
- Customer carriers can use any link layer technology to connect the CE routers to the PE routers.
- The customer carrier can use any addressing scheme and still be supported by a backbone carrier.

### Benefits of Implementing MPLS VPN CSC Using BGP

The benefits of using BGP to distribute IPv4 routes and MPLS label routes are:

- BGP takes the place of an IGP and LDP in a VPN forwarding and routing instance (VRF) table.
- BGP is the preferred routing protocol for connecting two ISPs.

## Configure Carrier Supporting Carrier for L3VPN

Perform this task on CSC-PE to configure Carrier Supporting Carrier for L3VPN.

## Configuration Example

```

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
Router(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

```

## Running Configuration

This section shows 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

```

## Verification

Verify the Carrier Supporting Carrier configuration.

```

Router:CSC-PE1# show cef vrf vpnl 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 100.12.1.2/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
    path-idx 0 NHID 0x0 [0xa09e9fa8 0x0]
    recursion-via-/32
    next hop 100.12.1.2/32 via 100516/0/21
      local label 100927
      next hop 100.12.1.2/32 Hu0/0/0/5.1 labels imposed {ImplNull 200013}
  via 100.12.101.2/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
    path-idx 1 NHID 0x0 [0xa09ealb8 0x0]
    recursion-via-/32
    next hop 100.12.101.2/32 via 100520/0/21
      local label 100927
      next hop 100.12.101.2/32 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 100.25.1.5/32, HundredGigE0/0/0/1.1, 3 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8ac12a48 0x0]
    next hop 100.25.1.5/32
      local adjacency
      local label 200013 labels imposed {ExpNullv4}

Load distribution: 0 (refcount 2)

```

Hash	OK	Interface	Address
0	Y	HundredGigE0/0/0/1.1	100.25.1.5



## CHAPTER 4

# VXLAN Static Routing

VXLAN static routing provides a method for connecting multiple servers in a data center to an enterprise edge router. This chapter covers these topics:

- [VXLAN Static Routing, on page 69](#)
- [Key Concepts, on page 74](#)
- [Configure VXLAN Static Routing, on page 75](#)

## VXLAN Static Routing

*Table 9: Feature History Table*

Feature Name	Release Information	Feature Description
VXLAN Static Routing	Release 24.4.1	Introduced in this release on: Fixed Systems (8700) (select variants only*)  *VXLAN Static Routing is now supported on Cisco 8712-MOD-M routers.

Feature Name	Release Information	Feature Description
VXLAN Static Routing	Release 24.2.11	<p>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.</p> <p>Alternatively, you can use Service Layer API for faster provisioning of VXLAN static routing.</p> <p>This feature is supported only on the following PIDs:</p> <ul style="list-style-type: none"> <li>• 8202-32FH-M</li> <li>• 8101-32H</li> <li>• 8201-32FH</li> </ul> <p>This feature introduces these changes:</p> <ul style="list-style-type: none"> <li>• <b>CLI:</b> <ul style="list-style-type: none"> <li>• <a href="#">host-reachability protocol static</a></li> <li>• <a href="#">overlay-encapsulation</a></li> <li>• <a href="#">hw-module profile cef vxlan ipv6-tnl-scale</a></li> </ul> </li> <li>• <b>YANG Data Model:</b> (see <a href="#">GitHub</a>, <a href="#">YANG Data Models Navigator</a>) <ul style="list-style-type: none"> <li>• <code>Cisco-IOS-XR-tunnel-nve-cfg</code></li> <li>• <code>Cisco-IOS-XR-ip-static-cfg</code></li> </ul> </li> </ul>

### Introduction to VXLAN

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.

But in today's networks, you might have a data center with lots of virtualization and need to isolate several virtual machines (VMs) from other VMs where you could easily run out of VLANs. So, there is a need to provide robust tunneling mechanisms to isolate and load-balance traffic inside the provider's network.

Virtual Extensible LAN (VXLAN) addresses some of the limitations of traditional VLANs in large-scale and cloud-based environments. VXLAN is widely used in data center environments where there is a need for virtualized networks to support cloud computing and virtualization technologies. It is also used in service provider networks to provide 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. VTEP encapsulates Layer 2 Ethernet frames within the Layer 4 User Datagram Protocol (UDP) and transports the encapsulated frames over a Layer 3 network.

VXLAN introduces an 8-byte VXLAN header that consists of a 24-bit VXLAN network identifier (VNI) with the original Ethernet frame added in the UDP payload. The 24-bit VNI is used to identify Layer 2 segments and maintain Layer 2 isolation between the segments.

With all 24 bits in VNI, VXLAN can support 16 million LAN segments. The VNI is used to designate the individual VXLAN overlay network on which the communicating virtual machines (VMs) are situated. VMs in different VXLAN overlay networks cannot communicate with each other.

VXLAN is a Layer 2 tunneling protocol that connects multiple servers in a data center that provide cloud services to customers and the enterprise edge router. VXLAN automatically configures underlay tunnels between the router and servers and overlay routing within those tunnels. VXLAN creates virtual networks on top of an underlay network. The underlay network is typically a physical IP network. VXLAN underlay can be IPv4 or IPv6 packets. The underlay and overlay networks are independent, and changes in the underlay don't affect the overlay. You can add or remove a router in the underlay network without affecting the overlay network.

VXLAN allows you to tunnel Ethernet frames over IP transport that uses IP and UDP as the transport protocol. A tunnel is created that enables you to extend 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 placed inside a UDP packet to send to the routed domain. The VXLAN adds Layer 2 header, and the remote endpoint can ignore this header. The VXLAN tunnel endpoint (VTEP) is a router that encapsulates and de-encapsulates Layer 2 traffic.

When a host sends traffic:

- The VXLAN encapsulates the traffic in UDP and IP headers.
- VXLAN encodes the flow information in the UDP source port to enable routers to perform flow-based load balancing.  
  
Flow-based load balancing identifies different flows of traffic based on the key fields in the data packet. For example, IPv4 source and destination IP addresses can be used to identify a flow.
- VXLAN encapsulates these packets into the tunnel with an IPv4 or IPv6 outer header.
- After the traffic reaches the destination router, the router decapsulates the packet and sends it to the destination host.
- VXLAN adds the custom source MAC address in the inner header that encodes the information in the MAC address where your internal network devices can extract the required information.

For more information on VXLAN, see [Key Concepts, on page 74](#).

### Benefits of VXLAN

VXLAN provides the following benefits:

- High throughput through dedicated VPN connectivity between servers and enterprise edge routers.
- Allows the creation of overlay networks independent of the underlying physical network, which provides greater flexibility in network design and deployment.
- Flexible placement of multitenant segments throughout the data center with the creation of isolated virtual networks for multiple tenants, providing greater security and separation between different users.

- Extends Layer 2 segments over the underlying shared network infrastructure to manage tenant workloads across physical pods in the data center.
- Uses a 24-bit VXLAN Network Identifier (VNI) which enables the creation of up to 16 million unique virtual networks, providing greater scalability.
- Facilitates network load balancing using the source UDP port within the VXLAN outer header.

### VXLAN Static Routing

You can use VXLAN static routing to interconnect non-VXLAN, such as MPLS and VXLAN domains. VXLAN static routing defines the path for VXLAN traffic from the source VTEP to reach the destination VTEP and involves configuring static routes on the underlying Layer 3 network to direct the VXLAN traffic to the appropriate VTEPs.

VXLAN supports up to 160000 static routes by default. However, you can increase the scale value up to one million VXLAN static routes for IPv6 tunnel remote nexthop using the [hw-module profile cef vxlan ipv6-tnl-scale](#) command.

### Benefits of VXLAN Static Routing

- You can use static routes in scenarios where consistent routing decisions are required. Because the 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 and thereby have direct control over traffic.
- Static routes are useful for specific traffic engineering or policy requirements.
- You do not have to maintain routing tables for static routing, hence reduces any overhead associated with routing protocols.

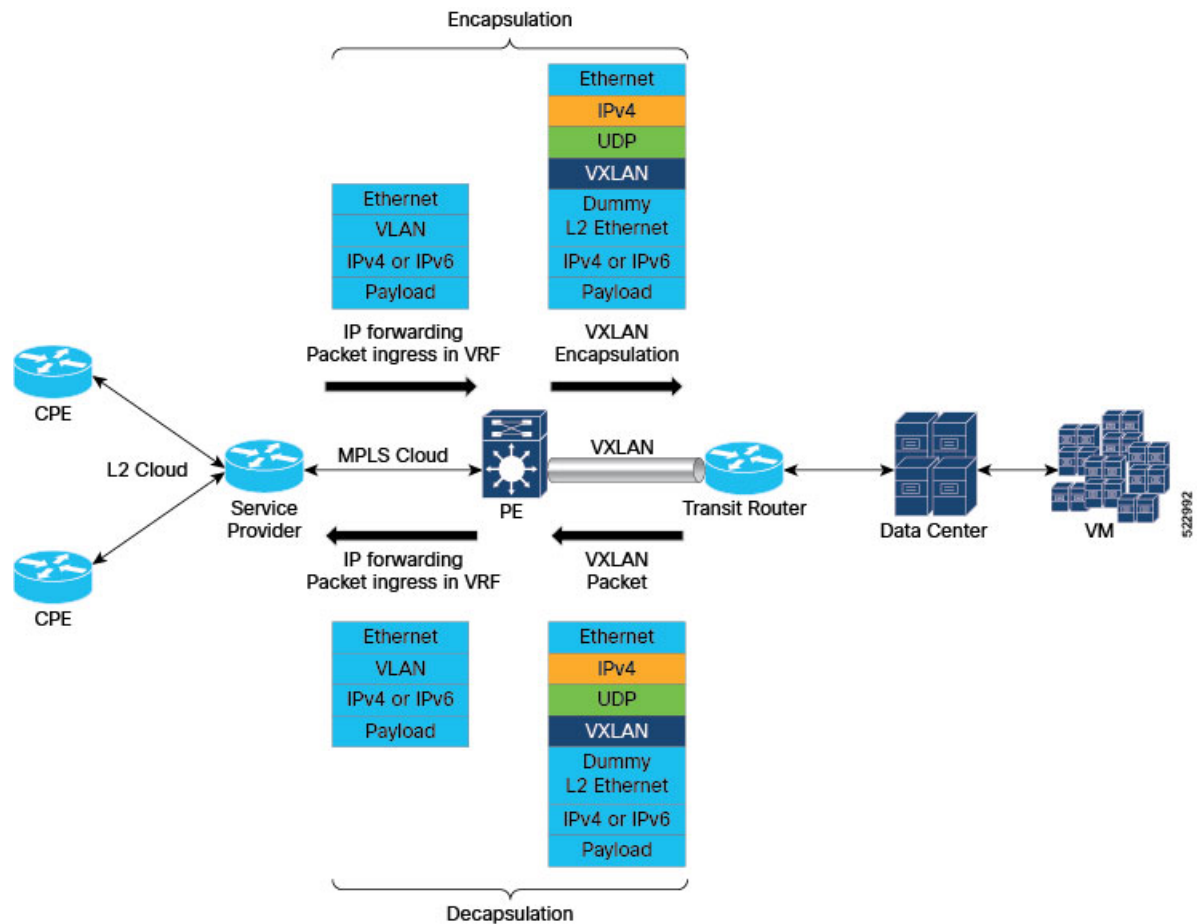
### Restrictions for VXLAN Static Routing

- Identical remote next hops with the same NVE with different VNI is not supported.
- Two same remote next hops in the same VRF, when the VNI is different, are not supported. However, same next hops with different NVE is supported.
- The index can be the same within a single VRF for the same VNI. However, the same VNI for different VRFs, must have a different index.
- Source MAC address in the static encapsulation route cannot be a multicast address.

### Topology

Let's understand how VXLAN static routing works using this topology.





In this topology,

- The PE router receives Layer 3 traffic on the VRF interface.
- The VXLAN tunnel starts at the PE router and terminates on the transit router or the servers behind the transit router.
- A BGP session is established over the tunnel.
- When Layer 3 traffic enters the PE router from the CPE, the PE router encapsulates the packet into the VXLAN tunnel and sends the traffic to the transit routers. Traffic that enters the PE has the VLAN tags for tracking customers within your network domain. VLANs map to VRF on the PE and to the VXLAN VNI. VXLAN encapsulation can be IPv4 or IPv6.
- The traffic is distributed based on your configured UDP source port. UDP source port value ranges from 49152 to 65535.
- The VXLAN tunnel terminates at the transit router.
- The transit router decapsulates the packet and performs an IP lookup to route traffic to the customer VM.

Similarly, the traffic from the VM is encapsulated as VXLAN. There will be an additional L2 header in the packet that must be terminated. Both VXLAN and the inner L2 header are terminated on the PE.

### VXLAN Static Routing using Service Layer API

You can configure VXLAN static routing using Service Layer API, which results in faster provisioning, easier scaling, and improved overall management of VXLAN networks.

Large cloud providers often need to provision tunneling mechanisms quickly and at a large scale to isolate end customer traffic dynamically. However, the traditional method of configuring the router through CLI might not be efficient. We provide Service Layer API as an alternate way to manage routers to address this challenge. With Service Layer API, you can have granular control of network traffic over the forwarding plane. It leverages Google's gRPC to generate client and server bindings, which allows users to program the forwarding plane using a variety of programming languages.

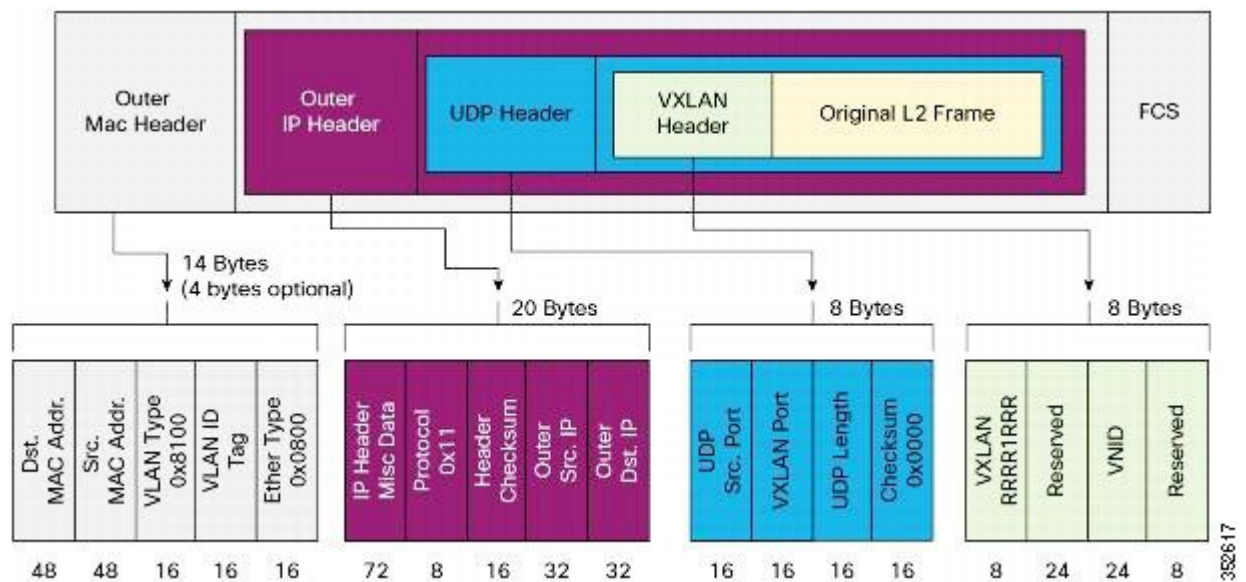
For more information on Service Layer API, see the *Use Service Layer API to Bring your Controller on Cisco IOS XR Router* chapter in the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

## Key Concepts

### VXLAN Packet Format

Here is the VXLAN packet format.

**Figure 10: VXLAN Packet Format**



### VXLAN Tunnel Endpoint

A VXLAN tunnel endpoint (VTEP) can be a physical or virtual router that connects the overlay and the underlay networks. A VTEP device is identified in the IP transport network using a unique IP address, which is a loopback interface IP address. The VTEP device uses this IP address to encapsulate Ethernet frames and transmits the encapsulated packets to the transport network through the IP interface. A source and destination VTEP creates a stateless tunnel to deliver traffic from one host to another. When a frame for a remote host reaches a device, the frame is encapsulated in IP and UDP headers. A maximum of 8k VXLAN tunnel interface per VTEP is supported.

### Load Sharing with VXLANs

Most data center transport networks are designed and deployed with multiple redundant paths that utilize various multipath load-sharing technologies to distribute traffic loads on all available paths. Encapsulated VXLAN packets are forwarded between VTEPs based on the native forwarding decisions of the transport network.

A typical VXLAN transport network is an IP-routing network that uses the standard IP equal cost multipath (ECMP) to balance the traffic load among multiple best paths. To avoid out-of-sequence packet forwarding, flow-based ECMP is commonly deployed. An ECMP flow is defined by the source and destination IP addresses.

All the VXLAN packet flows between a pair of VTEPs have the same outer source and destination IP addresses. All VTEP devices must use one identical destination UDP port, either the Internet Assigned Numbers Authority (IANA)-allocated UDP port 4789 or a customer-configured port. The source UDP port is the only variable element in the ECMP flow definition that can differentiate VXLAN flows from the transport network standpoint. The VXLAN outer-packet header uses source UDP port for link load-share hashing, which is the only element that can uniquely identify a VXLAN flow. A VXLAN flow is unique as the VXLAN inner frame header considers the VXLAN source UDP port for load balancing.

## Configure VXLAN Static Routing

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

```
/* Configure 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

/* Configure interface NVE 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 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 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

/* 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)# 11.1.1.1/32 remote-next-hop 10.10.10.10 tunnel VXLAN index
1 nve 1 evni 1 src-mac aaal.bbb1.ccc1 -> IPv4 over IPv4
Router(config-static-vrf-afi)# 11.1.1.2/32 remote-next-hop 10:10:10::10 tunnel VXLAN index
3 nve 1 evni 3 src-mac aaal.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 aaal.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 aaal.bbb1.ccc1 -> IPv6 over IPv6
Router(config-static-vrf-afi)# commit

/* Configure customized UDP destination port and UDP source port range */
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

```

## Running Configuration

This section shows VXLAN static routing running configuration.

```
/* Configure VRF */
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

/* NVE and Encapsulation*/
interface nve1
  member vni 2
    vrf vrf1
    host-reachability protocol protocol static
  !

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

/* Static Routing */
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
    11.1.1.1/32 remote-next-hop 10.10.10.10 tunnel VXLAN index 1 nve 1 evni 1 src-mac
    aaal.bbb1.ccc1. ==> IPv4oIPv4

    11.1.1.2/32 remote-next-hop 10:10:10::10 tunnel VXLAN index 3 nve 1 evni 3 src-mac
    aaal.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
    aaal.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
    aaal.bbb1.ccc1 ==> ipv6 over ipv6

/* Decapsulation */
configure
nve
  overlay-encap vxlan
  udp-port destination 65330
  udp-port src-port start 1024
  !
  !

```

## Verification

Verify that the NVE interface is Up.

```

Router# show nve interface nve 1
Interface: nve1 State: Up Encapsulation: VxLAN
  Source Interface: Loopback0 (primary: v4: 1.1.1.1 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

```

Verify the static routing configuration.

```

Router# show route vrf VRF1 11.1.1.1 detail

Routing entry for 11.1.1.1/32
  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: 0x1000000000000001
      MPLS eid:0xfffffffffffffd
      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: 0xfffffffffffffd
      No advertising protos.

```

## VXLAN Static Routing using Service Layer API

### Configuration Example

```

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

```

### Running Configuration

```

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

```

### Verification

Verify the service-layer 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

```







## CHAPTER 5

# Implementing IPv6 VPN Provider Edge Transport over MPLS

---

IPv6 Provider Edge or IPv6 VPN Provider Edge (6PE/6VPE) uses the existing MPLS IPv4 core infrastructure for IPv6 transport. 6PE/6VPE enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS label switched paths (LSPs).

This feature relies heavily on multiprotocol Border Gateway Protocol (BGP) extensions in the IPv4 network configuration on the provider edge (PE) router to exchange IPv6 reachability information (in addition to an MPLS label) for each IPv6 address prefix. Edge routers are configured as dual-stack, running both IPv4 and IPv6, and use the IPv4 mapped IPv6 address for IPv6 prefix reachability exchange.

Familiarity with MPLS and BGP4 configuration and troubleshooting is required for implementing 6PE/6VPE.

- [Overview of 6PE/6VPE, on page 82](#)
- [Benefits of 6PE/6VPE, on page 83](#)
- [Deploying IPv6 over MPLS Backbones, on page 83](#)
- [IPv6 on the Provider Edge and Customer Edge Routers, on page 83](#)
- [OSPFv3 \(CE to PE\), on page 84](#)
- [Configuring 6PE/6VPE, on page 85](#)
- [Configuring OSPFv3 as the Routing Protocol Between the PE and CE Routers, on page 89](#)
- [Configure BGP as the Routing Protocol Between the PE and CE Routers, on page 90](#)

# Overview of 6PE/6VPE

Table 10: Feature History Table

Feature Name	Release	Description
6PE/6VPE	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC:K100])(select variants only*).</p> <p>The router now enable seamless integration of IPv6 networks over an MPLS backbone, allowing service providers to offer IPv6 services without changing the core MPLS infrastructure. 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.</p> <p>*Previously this feature was supported on Q200 and Q100. It is now extended to Cisco 8712-MOD-M routers.</p>

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

- Dedicated IPv6 network running over various data link layers
- Dual-stack IPv4-IPv6 backbone
- Existing MPLS backbone leverage

These solutions are deployed on service providers' backbones when the amount of IPv6 traffic and the revenue generated are in line with the necessary investments and the agreed-upon risks. Conditions are favorable for the introduction of native IPv6 services, from the edge, in a scalable way, without any IPv6 addressing restrictions and without putting a well-controlled IPv4 backbone in jeopardy. Backbone stability is essential for service providers that have recently stabilized their IPv4 infrastructure.

Service providers running an MPLS/IPv4 infrastructure follow similar trends because several integration scenarios that offer IPv6 services on an MPLS network are possible. Cisco Systems has specially developed Cisco 6PE or IPv6 Provider Edge Router over MPLS, to meet all those requirements.

Inter-AS support for 6PE requires support of Border Gateway Protocol (BGP) to enable the address families and to allocate and distribute PE and ASBR labels.



**Note** Cisco IOS XR displays actual IPv4 next-hop addresses for IPv6 labeled-unicast and VPNv6 prefixes. IPv4-mapped-to-IPv6 format is not supported.

## Benefits of 6PE/6VPE

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

## Deploying IPv6 over MPLS Backbones

Backbones enabled by 6PE (IPv6 over MPLS) allow IPv6 domains to communicate with each other over an MPLS IPv4 core network. This implementation requires no backbone infrastructure upgrades and no reconfiguration of core routers, because forwarding is based on labels rather than on the IP header itself. This provides a very cost-effective strategy for IPv6 deployment.

## IPv6 on the Provider Edge and Customer Edge Routers

### Service Provider Edge Routers

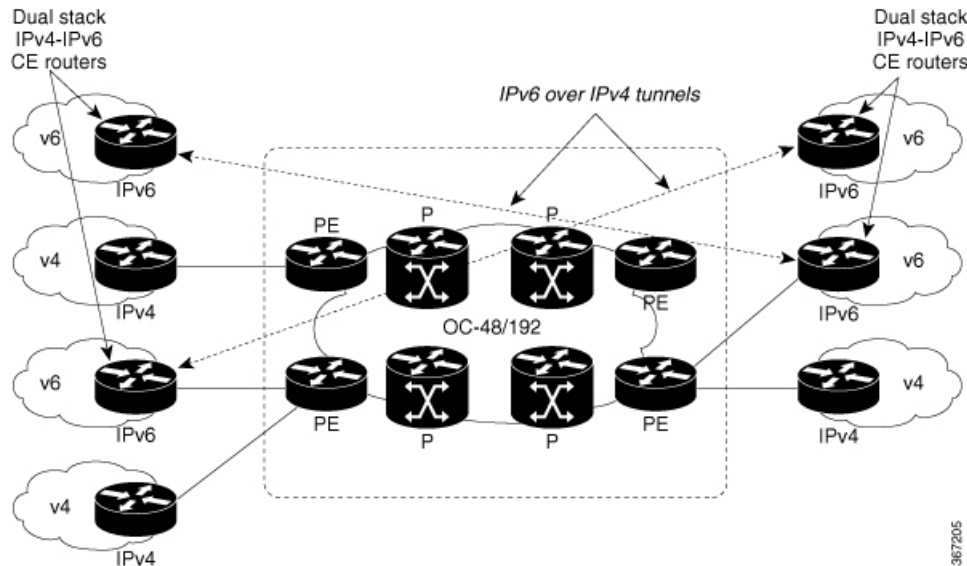
6PE is particularly applicable to service providers who currently run an MPLS network. One of its advantages is that there is no need to upgrade the hardware, software, or configuration of the core network, and it eliminates the impact on the operations and the revenues generated by the existing IPv4 traffic. MPLS is used by many service providers to deliver services to customers. MPLS as a multiservice infrastructure technology is able to provide layer 3 VPN, QoS, traffic engineering, fast re-routing and integration of ATM and IP switching.

### Customer Edge Routers

Using tunnels on the CE routers is the simplest way to deploy IPv6 over MPLS networks. It has no impact on the operation or infrastructure of MPLS and requires no changes to the P routers in the core or to the PE routers. However, tunnel meshing is required as the number of CEs to connect increases, and it is difficult to delegate a global IPv6 prefix for an ISP.

The following figure illustrates the network architecture using tunnels on the CE routers.

Figure 11: IPv6 Using Tunnels on the CE Routers



### IPv6 Provider Edge Multipath

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.

When MP-IBGP multipath is enabled on the 6PE router, all labeled paths are installed in the forwarding table with available MPLS information (label stack). This functionality enables 6PE to perform load balancing.

## OSPFv3 (CE to PE)

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 implementation. This feature allows:

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

### Multiple VRF Support

OSPFv3 supports multiple VRFs in a single routing process that allows scaling to tens and hundreds of VRFs without consuming too much route processor (RP) resources. Multiple OSPFv3 processes can be configured on a single router. In large-scale VRF deployments, this allows partition VRF processing across multiple RPs. It is also used to isolate default routing table or high impact VRFs from the regular VRFs. It is recommended to use a single process for all the VRFs. If needed, a second OSPFv3 process must be configured for IPv6 routing.




---

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

---

### OSPFv3 PE-CE Extensions

IPv6 protocol is being vastly deployed in today's customer networks. Service Providers (SPs) need to be able to offer Virtual Private Network (VPN) services to their customers for supporting IPv6 protocol, in addition to the already offered VPN services for IPv4 protocol.

In order to support IPv6, routing protocols require additional extensions for operating in the VPN environment. Extensions to OSPFv3 are required in order for OSPFv3 to operate at the PE-CE links.

### VRF Lite

VRF lite feature enables VRF deployment without BGP or MPLS based backbone. In VRF lite, the PE routers are directly connected using VRF interfaces. For OSPFv3, the following needs to operate differently in the VRF lite scenario, as opposed to the deployment with BGP or MPLS backbone:

- DN bit processing—In VRF lite environment, the DN bit processing is disabled.
- ABR status—In VRF context (except default VRF), OSPFv3 router is automatically set as an ABR, regardless to its connectivity to area 0. This automatic ABR status setting is disabled in the VRF lite environment.




---

**Note** To enable VRF Lite, issue the **capability vrf-lite** command in the OSPFv3 VRF configuration submenu.

---

## Configuring 6PE/6VPE

### Configuration Example

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, you can use all routing protocols supported on Cisco IOS XR software such as BGP, OSPF, IS-IS, and Static to learn routes from both clouds. However, for 6VPE, you can use only the BGP, and Static routing protocols to learn routes. Also, 6VPE supports OSPFv3 routing protocol between PE and CE routers.
- While configuring 6PE/6VPE on the router, it is mandatory to configure label allocation mode, per-vrf for all routers including peer routers.
- Route policies must be configured prior to configuring 6PE/6VPE.
- BGP uses the **per-vrf** label mode for transporting local and redistributed IP prefixes. Before IOS XR Release 7.5.3, BGP assigned a random label for the prefixes. Starting from Release 7.5.3, BGP assigns a label value of 2, the IPv6 Explicit NULL Label, for the same prefixes.

```
Router#configure
Router(config)#router bgp 10
```

```

Router(config-bgp)#bgp router-id 11.11.11.11
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 13.13.13.13
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-bgp-vrf-af)#!
!
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

```

## Running Configuration

```
router bgp 10
  bgp router-id 11.11.11.11
  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 13.13.13.13
    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
!
```

## Verification

```
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::/64 is directly connected,
[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):
  13.13.13.13
```

```
Path #1: Received by speaker 0
```

```
Advertised to update-groups (with more than one peer):
  0.1
```

```
Advertised to peers (in unique update groups):
  13.13.13.13
```

```
201
```

```
66:1:2::2 from 66:1:2::2 (39.229.0.1)
  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
```



```

11.11.11.11 (metric 5) from 13.13.13.13 (11.11.11.11)
  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: 11.11.11.11, Cluster list: 5.5.5.5

```

## Configuring OSPFv3 as the Routing Protocol Between the PE and CE Routers

### Configuration Example

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

```

Router#config
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)#
Router(config-ospfv3-vrf-ar-if)#interface TenGigE0/7/0/0/3.7
Router(config-ospfv3-vrf-ar-if)#

```

### Running Configuration

```

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

```

### Verification

```

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	<b>FULL</b> /DROTHER	00:00:36	0	TenGigE0/7/0/0/3.7
Neighbor is up for 1w0d					

Total neighbor count: 1

# Configure BGP as the Routing Protocol Between the 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 are iBGP sessions, and PE to CE sessions are eBGP sessions. PE to CE eBGP sessions can be directly or indirectly connected (eBGP multihop).

## Configuration Example

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

### PE1:

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

### CE1:

```
Router-CE1#configure
Router-CE1 (config) #router bgp 2001
Router-CE1 (config-bgp) #bgp router-id 8.8.8.1
Router-CE1 (config-bgp) #address-family 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
```

## Running Configuration

### PE1:

```

router bgp 2001
  bgp router-id 13.13.13.1
  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
    !
  !
  !

```

### CE1:

```

router bgp 7501
  bgp router-id 8.8.8.1
  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
  !
  !

```

### Verification

#### • 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.
```

