



## **L3VPN Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 7.1.x**

**First Published:** 2020-01-01

**Last Modified:** 2020-01-27

### **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).

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

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)

© 2020 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### PREFACE

#### **Preface** vii

Changes to This Document vii

Communications, Services, and Additional Information vii

---

### CHAPTER 1

#### **New and Changed Feature Information** 1

New and Changed L3VPN Features 1

---

### CHAPTER 2

#### **Implementing MPLS Layer 3 VPNs** 3

MPLS L3VPN Overview 3

How MPLS L3VPN Works 4

Major Components of MPLS L3VPN 5

Restrictions for MPLS L3VPN 5

Inter-AS Support for L3VPN 6

Inter-AS Support: Overview 6

Inter-AS and ASBRs 6

Confederations 7

MPLS VPN Inter-AS BGP Label Distribution 8

Exchanging IPv4 Routes with MPLS labels 9

How to Implement MPLS Layer 3 VPNs 11

Prerequisites for Implementing MPLS L3VPN 11

Configure the Core Network 11

Assess the Needs of MPLS VPN Customers 11

Configure Routing Protocols in the Core 12

Configure MPLS in the Core 13

Determine if FIB is Enabled in the Core 14

Configure Multiprotocol BGP on the PE Routers and Route Reflectors 14

Connect MPLS VPN Customers	18
Define VRFs on PE Routers to Enable Customer Connectivity	18
Configure VRF Interfaces on PE Routers for Each VPN Customer	19
Configure Routing Protocol Between the PE and CE Routers	21
Verify MPLS L3VPN Configuration	28
Verify the L3VPN Traffic Flow	29
Verify the Underlay (transport)	29
Verify the Overlay (L3VPN)	31
Providing VPN Connectivity Across Multiple Autonomous Systems with MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels	33
Concept	33
Configuring ASBRs to Exchange IPv4 Routes and MPLS Labels	33
Configuring the Route Reflectors to Exchange VPN-IPv4 Routes	35
Configure the Route Reflectors to Reflect Remote Routes in its AS	37
Providing VPN Connectivity Across Multiple Autonomous Systems with MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses	38
Configuring the ASBRs to Exchange VPN-IPv4 Addresses for IP Tunnels	38
Configuring a Static Route to an ASBR Peer	41
Configuring EBGW Routing to Exchange VPN Routes Between Subautonomous Systems in a Confederation	42
Configuring MPLS Forwarding for ASBR Confederations	45
Configuring a Static Route to an ASBR Confederation Peer	46
VRF-lite	47
Configure VRF-lite	47
MPLS L3VPN Services using Segment Routing	51
Configure MPLS L3VPN over Segment Routing	51
Configure Segment Routing in MPLS Core	52
Verify MPLS L3VPN Configuration over Segment Routing	55
Single Pass GRE Encapsulation Allowing Line Rate Encapsulation	56
Configuration	56
Running Configuration	60
Verification	63
Implementing MPLS L3VPNs - References	66
MPLS L3VPN Benefits	66
Major Components of MPLS L3VPN—Details	66

Virtual Routing and Forwarding Tables	66
VPN Routing Information: Distribution	67
BGP Distribution of VPN Routing Information	67
MPLS Forwarding	67
Automatic Route Distinguisher Assignment	68
Layer 3 QinQ	68
Configure Layer 3 QinQ	69

**CHAPTER 3****Implementing IPv6 VPN Provider Edge Transport over MPLS 71**

Overview of 6PE/VPE	71
Benefits of 6PE/VPE	72
Deploying IPv6 over MPLS Backbones	72
IPv6 on the Provider Edge and Customer Edge Routers	72
OSPFv3 (CE to PE)	73
Restrictions for 6VPE	74
Configuring 6PE/VPE	74
Configuring OSPFv3 as the Routing Protocol Between the PE and CE Routers	78

**CHAPTER 4****Implementing DCI Layer 3 Gateway between MPLS-VPN and EVPN Data Center 79**

Data Center Interconnect between MPLS-VPN and EVPN-MPLS	79
DCI Layer 3 Gateway with EVPN-MPLS	79
VPNv4-Regular RT and EVPN-Stitching RT	81
EVPN-Regular RT and VPNv4-Stitching RT	93





## Preface

---



---

**Note** This product has reached end-of-life status. For more information, see the [End-of-Life and End-of-Sale Notices](#).

---

This preface contains these sections:

- [Changes to This Document, on page vii](#)
- [Communications, Services, and Additional Information, 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
January 2020	Initial release of this document.

## Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit [Cisco Marketplace](#).
- To obtain general networking, training, and certification titles, visit [Cisco Press](#).
- To find warranty information for a specific product or product family, access [Cisco Warranty Finder](#).

### **Cisco Bug Search Tool**

[Cisco Bug Search Tool](#) (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.





# CHAPTER 1

## New and Changed Feature Information

This table summarizes the new and changed feature information for the L3VPN Configuration Guide for Cisco NCS 5500 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 7.1.x*

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





## CHAPTER 2

# Implementing MPLS Layer 3 VPNs

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

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



---

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

---

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

- [BGP](#)
- [MPLS](#)
- [Routing](#)
- [VPN and Ethernet Services](#)

This chapter includes topics on:

- [MPLS L3VPN Overview, on page 3](#)
- [How MPLS L3VPN Works, on page 4](#)
- [How to Implement MPLS Layer 3 VPNs, on page 11](#)
- [VRF-lite, on page 47](#)
- [MPLS L3VPN Services using Segment Routing, on page 51](#)
- [Single Pass GRE Encapsulation Allowing Line Rate Encapsulation, on page 56](#)
- [Implementing MPLS L3VPNs - References, on page 66](#)
- [Layer 3 QinQ , on page 68](#)

## MPLS L3VPN Overview

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

- An IP-based network delivering private network services over a public infrastructure

- A set of sites that are allowed to communicate with each other privately over the Internet or other public or private networks

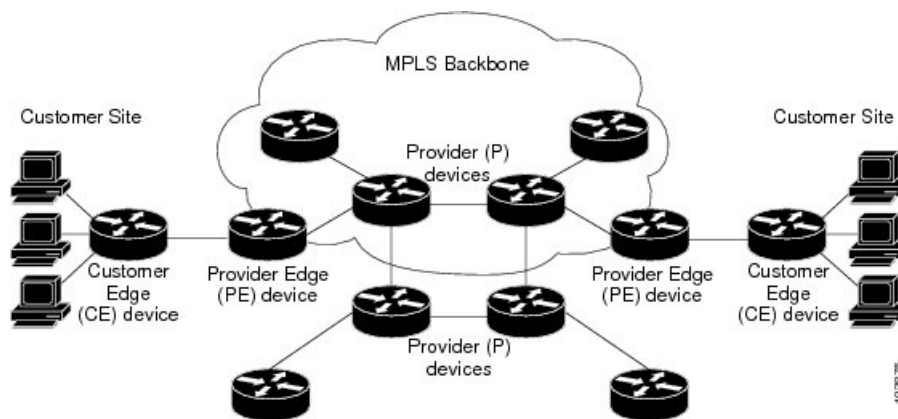
Conventional VPNs are created by configuring a full mesh of tunnels or permanent virtual circuits (PVCs) to all sites in a VPN. This type of VPN is not easy to maintain or expand, as adding a new site requires changing each edge device in the VPN.

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

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

The following figure depicts a basic MPLS VPN topology.

**Figure 1: Basic MPLS VPN Topology**



These are the basic components of MPLS VPN:

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

## How MPLS L3VPN Works

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

- Exchanges routing updates with the CE router

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

## Major Components of MPLS L3VPN

An MPLS-based VPN network has three major components:

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

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

Read more at [Major Components of MPLS L3VPN—Details, on page 66](#).

## Restrictions for MPLS L3VPN

Implementing MPLS L3VPN in Cisco NCS 5500 Series Routers 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 the prefix hash in control plane. The selected path is programmed in the data plane.
- TTL propagation cannot be disabled. TTL propagation always happens from IP->MPLS and MPLS->IP.

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

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

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



---

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

---

## Inter-AS Support for L3VPN

This section contains the following topics:

### Inter-AS Support: Overview

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

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

An MPLS VPN Inter-AS provides the following benefits:

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

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

- Allows a VPN to exist in different areas.

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

- Allows confederations to optimize iBGP meshing.

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

### Inter-AS and ASBRs

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

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

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

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

- Interprovider VPN—MPLS VPNs that include two or more autonomous systems, connected by separate border edge routers. The autonomous systems exchange routes using eBGP. No IGP or routing information is exchanged between the autonomous systems.
- BGP Confederations—MPLS VPNs that divide a single autonomous system into multiple subautonomous systems and classify them as a single, designated confederation. The network recognizes the confederation as a single autonomous system. The peers in the different autonomous systems communicate over eBGP sessions; however, they can exchange route information as if they were iBGP peers.



---

**Note** Inter-AS options A and C are supported.

---

## Confederations

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

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

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

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



---

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

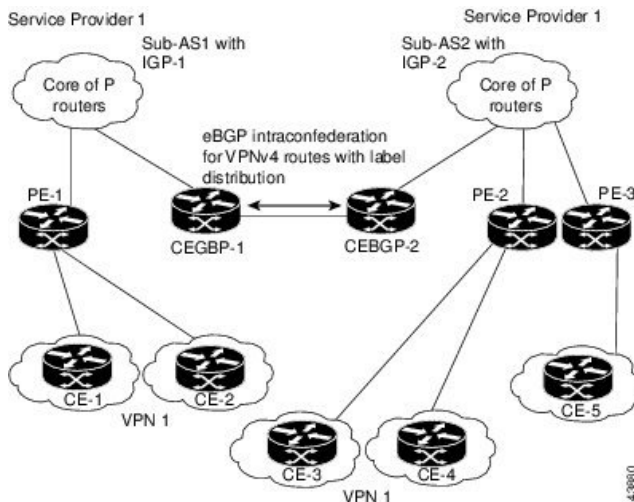
---

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

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

- The distributing router changes the next-hop addresses and labels and uses a next-hop-self address.
- IGP-1 and IGP-2 know the addresses of CEBGP-1 and CEBGP-2.

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



In this confederation configuration:

- CEBGP border edge routers function as neighboring peers between the subautonomous systems. The subautonomous systems use eBGP to exchange route information.
- Each CEBGP border edge router (CEBGP-1 and CEBGP-2) assigns a label for the router before distributing the route to the next subautonomous system. The CEBGP border edge router distributes the route as a VPN-IPv4 address by using the multiprotocol extensions of BGP. The label and the VPN identifier are encoded as part of the NLRI.
- Each PE and CEBGP border edge router assigns its own label to each VPN-IPv4 address prefix before redistributing the routes. The CEBGP border edge routers exchange IPv-IPv4 addresses with the labels. The next-hop-self address is included in the label (as the value of the eBGP next-hop attribute). Within the subautonomous systems, the CEBGP border edge router address is distributed throughout the iBGP neighbors, and the two CEBGP border edge routers are known to both confederations.
- For more information about how to configure confederations, see the “[Configuring MPLS Forwarding for ASBR Confederations, on page 45](#)”.

## MPLS VPN Inter-AS BGP Label Distribution



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

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



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

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

## Exchanging IPv4 Routes with MPLS labels



---

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

---

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

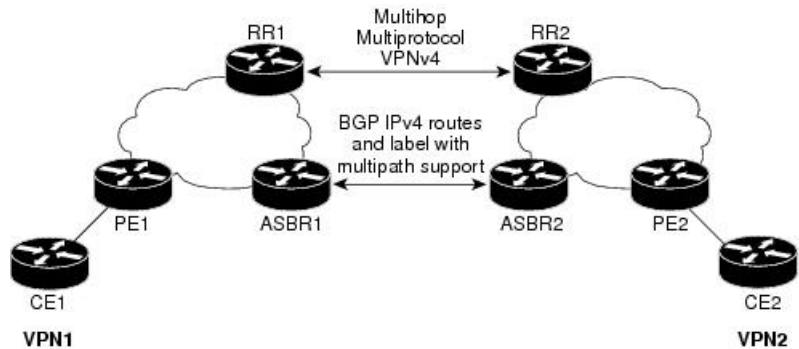
- Route reflectors exchange VPN-IPv4 routes by using multihop, multiprotocol eBGP. This configuration also preserves the next-hop information and the VPN labels across the autonomous systems.
- A local PE router (for example, PE1 in the figure below) needs to know the routes and label information for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

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

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

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

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



## BGP Routing Information

BGP routing information includes the following items:

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

## BGP Messages and MPLS Labels

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

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

## Sending MPLS Labels with Routes

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

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

## How to Implement MPLS Layer 3 VPNs

Implementing MPLS L3VPNs involves these main tasks:

### Prerequisites for Implementing MPLS L3VPN

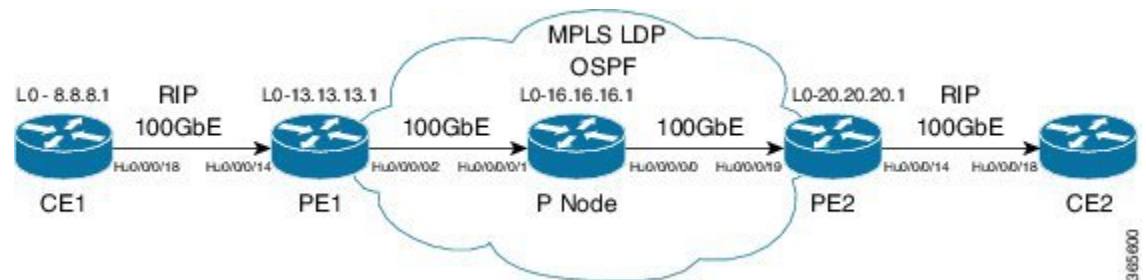
These are the prerequisites to configure MPLS L3VPN:

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

### Configure the Core Network

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

**Figure 4: L3VPN over MPLS LDP**



Configuring the core network involves these main tasks:

### Assess the Needs of MPLS VPN Customers

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

- Identify the size of the network:

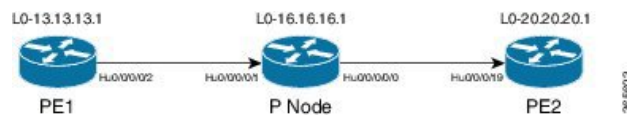
Identify the following to determine the number of routers and ports required:

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

Figure 5: 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-ar-if)#commit
```

### Running Configuration

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

### Verification

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

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

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

```
Total neighbor count: 1
```

### Related Topics

- [How to Implement MPLS Layer 3 VPNs, on page 11](#)

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

### Associated Commands

- [router-id](#)
- [router ospf](#)

## Configure MPLS in the Core

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

You can also transport MPLS L3VPN services using segment routing in the core. For details, see [Configure Segment Routing in MPLS Core, on page 52](#).

### Configuration Example

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

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

Repeat this configuration in PE2 and P routers as well.

### Running Configuration

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

### Verification

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

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
```

```

Graceful Restart: No
Session Holdtime: 180 sec
State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
Up time: 2w2d
LDP Discovery Sources:
  IPv4: (1)
    HundredGigE0/0/0/2
  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)

```

### Related Topics

- [How to Implement MPLS Layer 3 VPNs, on page 11](#)

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

### Associated Commands

- [mpls ldp](#)
- [show mpls ldp neighbor](#)

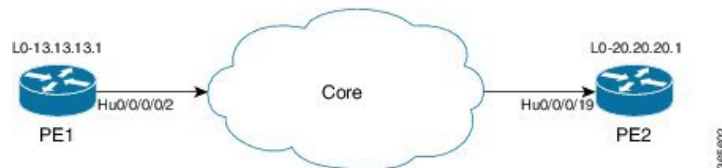
## Determine if FIB is Enabled in the Core

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

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

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

**Figure 6: Multiprotocol BGP on PE Routers**



### Configuration Example

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

```
Router-PE1#configure
```

```

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

```

### Running Configuration

```

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

```

### Verification

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

```

Router-PE1#show bgp neighbor

BGP neighbor is 20.20.20.1
  Remote AS 2001, local AS 2001, internal link
  Remote router ID 20.20.20.1
  BGP state = Established, up for 1d19h
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00

```

```

Hold time is 60, keepalive interval is 20 seconds
Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
Last write 00:00:16, attempted 19, written 19
Second last write 00:00:36, attempted 19, written 19
Last write before reset 00:00:00, attempted 0, written 0
Second last write before reset 00:00:00, attempted 0, written 0
Last write pulse rcvd Apr 12 10:31:20.739 last full not set pulse count 27939
Last write pulse rcvd before reset 00:00:00
Socket not armed for io, armed for read, armed for write
Last write thread event before reset 00:00:00, second last 00:00:00
Last KA expiry before reset 00:00:00, second last 00:00:00
Last KA error before reset 00:00:00, KA not sent 00:00:00
Last KA start before reset 00:00:00, second last 00:00:00
Precedence: internet
Non-stop routing is enabled
Multi-protocol capability received
Neighbor capabilities:
  Route refresh: advertised (old + new) and received (old + new)
  Graceful Restart (GR Awareness): received
  4-byte AS: advertised and received
  Address family IPv4 Unicast: advertised and received
  Address family VPNv4 Unicast: advertised and received
Received 25595 messages, 0 notifications, 0 in queue
Sent 8247 messages, 0 notifications, 0 in queue
Minimum time between advertisement runs is 0 secs
Inbound message logging enabled, 3 messages buffered
Outbound message logging enabled, 3 messages buffered

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

For Address Family: VPNv4 Unicast
BGP neighbor version 798487
Update group: 0.2 Filter-group: 0.1 No Refresh request being processed
AF-dependent capabilities:
  Graceful Restart capability received

```



```

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

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

```

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

```

Router-PE1#show bgp vpnv4 unicast

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

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 2001:1601 (default for vrf vrf1601)
*> 20.13.1.1/32      192.13.26.5              0 7501 i
*> 20.13.1.2/32      192.13.26.5              0 7501 i
*> 20.13.1.3/32      192.13.26.5              0 7501 i
*> 20.13.1.4/32      192.13.26.5              0 7501 i
*> 20.13.1.5/32      192.13.26.5              0 7501 i
**>i20.14.1.1/32     14.14.14.1              100 0 8501 i
**>i20.14.1.2/32     14.14.14.1              100 0 8501 i
**>i20.14.1.3/32     14.14.14.1              100 0 8501 i
**>i20.14.1.4/32     14.14.14.1              100 0 8501 i
**>i20.14.1.5/32     14.14.14.1              100 0 8501 i

```

## Related Topics

- [Configure the Core Network, on page 11](#)
- [Define VRFs on PE Routers to Enable Customer Connectivity, on page 18](#)

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

#### Associated Commands

- [neighbor](#)
- [router-bgp](#)
- [update-source](#)
- [vrf](#)
- [show bgp](#)

## Connect MPLS VPN Customers

Connecting MPLS VPN customers involves these main tasks:

- [Define VRFs on PE Routers to Enable Customer Connectivity, on page 18](#)
- [Configure VRF Interfaces on PE Routers for Each VPN Customer, on page 19](#)
- Configure the Routing Protocol between the PE and CE Routers

Use any of these options:

- [Configure BGP as the Routing Protocol Between the PE and CE Routers, on page 21](#)
- [Configure RIPv2 as the Routing Protocol Between the PE and CE Routers, on page 25](#)
- [Configure Static Routes Between the PE and CE Routers, on page 26](#)
- [Configure OSPF as the Routing Protocol Between the PE and CE Routers, on page 27](#)

## Define VRFs on PE Routers to Enable Customer Connectivity

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

#### Configuration Example

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

```
Router-PE1#configure
Router-PE1 (config)#vrf vrf1601
Router-PE1 (config-vrf)#address-family ipv4 unicast
Router-PE1 (config-vrf-af)#import route-target
```

```

Router-PE1 (config-vrf-af-import-rt) #2001:1601
Router-PE1 (config-vrf-af-import-rt) #exit
Router-PE1 (config-vrf-af) #export route-target
Router-PE1 (config-vrf-af-export-rt) #2001:1601
Router-PE1 (config-vrf-af-export-rt) #commit

```

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

### Running Configuration

```

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

```

### Verification

Verify the import and export route targets.

```

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

```

### Related Topics

- [Configure VRF Interfaces on PE Routers for Each VPN Customer, on page 19](#)
- [Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 14](#)

### Associated Commands

- [import route-policy](#)
- [import route-target](#)
- [export route-policy](#)
- [export route-target](#)
- [vrf](#)

## Configure VRF Interfaces on PE Routers for Each VPN Customer

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



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

### Configuration Example

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

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

### Running Configuration

```
interface HundredGigE0/0/0/14.1601
 vrf vrf1601
 ipv4 address 192.13.26.6 255.255.255.252
 encapsulation dot1q 1601
!
```

### Verification

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

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

### Related Topics

- [Define VRFs on PE Routers to Enable Customer Connectivity, on page 18](#)

## Configure Routing Protocol Between the PE and CE Routers

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

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

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



### Configuration Example

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

#### PE1:

```
Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 13.13.13.1
Router-PE1(config-bgp)#address-family ipv4 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#address-family vpnv4 unicast
Router-PE1(config-bgp-af)#exit
/* VRF configuration */
Router-PE1(config-bgp)#vrf vrf1601
Router-PE1(config-bgp-vrf)#rd 2001:1601
Router-PE1(config-bgp-vrf)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-af)#label mode per-vrf
Router-PE1(config-bgp-vrf-af)#redistribute connected
Router-PE1(config-bgp-vrf-af)#exit
Router-PE1(config-bgp-vrf)#neighbor 192.13.26.5
Router-PE1(config-bgp-vrf-nbr)#remote-as 7501
Router-PE1(config-bgp-vrf-nbr)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-nbr-af)#route-policy pass-all in
Router-PE1(config-bgp-vrf-nbr-af)#route-policy pass-all out
Router-PE1(config-bgp-vrf-nbr-af)#commit
```

#### CE1:

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

## Running Configuration

### PE1:

```

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

```

### CE1:

```

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

```

## Verification

### • PE1:

```

Router-PE1#show bgp neighbor
BGP neighbor is 192.13.26.5
  Remote AS 6553700, local AS 2001, external link
  Administratively shut down
  Remote router ID 192.13.26.5
  BGP state = Established
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00
  Hold time is 60, keepalive interval is 20 seconds
  Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
  Last write 00:00:16, attempted 19, written 19
  Second last write 00:00:36, attempted 19, written 19
  Last write before reset 00:00:00, attempted 0, written 0
  Second last write before reset 00:00:00, attempted 0, written 0
  Last write pulse rcvd Apr 12 10:31:20.739 last full not set pulse count 27939

```

```

Last write pulse rcvd before reset 00:00:00
Socket not armed for io, armed for read, armed for write
Last write thread event before reset 00:00:00, second last 00:00:00
Last KA expiry before reset 00:00:00, second last 00:00:00
Last KA error before reset 00:00:00, KA not sent 00:00:00
Last KA start before reset 00:00:00, second last 00:00:00
Precedence: internet
Non-stop routing is enabled
Graceful restart is enabled
Restart time is 120 seconds
Stale path timeout time is 360 seconds
Enforcing first AS is enabled
Multi-protocol capability not received
Received 0 messages, 0 notifications, 0 in queue
Sent 0 messages, 0 notifications, 0 in queue
Minimum time between advertisement runs is 30 secs
Inbound message logging enabled, 3 messages buffered
Outbound message logging enabled, 3 messages buffered

For Address Family: IPv4 Unicast
BGP neighbor version 0
Update group: 0.2 Filter-group: 0.0 No Refresh request being processed
Inbound soft reconfiguration allowed
AF-dependent capabilities:
  Outbound Route Filter (ORF) type (128) Prefix:
    Send-mode: advertised
    Receive-mode: advertised
  Graceful Restart capability advertised
    Local restart time is 120, RIB purge time is 600 seconds
    Maximum stalepath time is 360 seconds
Route refresh request: received 0, sent 0
Policy for incoming advertisements is pass-all
Policy for outgoing advertisements is pass-all
0 accepted prefixes, 0 are bestpaths
Cumulative no. of prefixes denied: 0.
Prefix advertised 0, suppressed 0, withdrawn 0
Maximum prefixes allowed 1048576
Threshold for warning message 75%, restart interval 0 min
An EoR was not received during read-only mode
Last ack version 1, Last synced ack version 0
Outstanding version objects: current 0, max 0
Additional-paths operation: None
Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with stitching-RT
option
Advertise VPNv6 routes is enabled with default option

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

```

• **CE1:**

```

Router-CE1#show bgp neighbor
BGP neighbor is 192.13.26.6
Remote AS 2001, local AS 6553700, external link
Remote router ID 192.13.26.6

```

```

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

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

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

```

## Related Topics

- [Connect MPLS VPN Customers, on page 18](#)



- [Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 14](#)

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

### Associated Commands

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

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

Figure 8: RIPv2 as the Routing Protocol between PE and CE Routers



### Configuration Example

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

#### PE1:

```

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

#### CE1:

```

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

## Running Configuration

### PE1:

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

### CE1:

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

## Related Topics

- [Connect MPLS VPN Customers, on page 18](#)

## Associated Commands

- [redistribute](#)
- [router rip](#)

## Configure Static Routes Between the PE and CE Routers

### Configuration Example

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

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

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

## Running Configuration

### PE1:

```

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

```

**CE1:**

```

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

```

**Related Topics**

- [Connect MPLS VPN Customers, on page 18](#)

**Associated Commands**

- router static

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

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

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

**Configuration Example**

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

**PE1:**

```

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

```

Repeat this configuration at PE2 node as well.

**CE1:**

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

**Running Configuration****PE1:**

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

**CE1:**

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

**Related Topics**

- [Connect MPLS VPN Customers, on page 18](#)

**Associated Commands**

- [router ospf](#)

## Verify MPLS L3VPN Configuration

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

## Verify the L3VPN Traffic Flow

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

### P node:

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

### PE2:

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

## 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)

  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  Hop           Switched
-----
24036  Pop       16.16.16.1/32   Hu0/0/0/2  191.22.1.2    293294

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

24039  24167    20.20.20.1/32   Hu0/0/0/2  191.22.1.2    17872433
      24167    20.20.20.1/32   Hu0/0/0/2.1 191.22.3.2    6345

24041  Aggregate vrf1601: Per-VRF Aggr[V] \
                                         vrf1601      7950400999
```

- Verify if label is updated in the hardware:

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

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes 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	<b>Programmed</b>

```
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  Outgoing  Prefix          Outgoing  Next Hop      Bytes
Label  Label     or ID           Interface  Interface     Switched
-----
24019  Pop       18.18.18.3/32  Hu0/0/0/19  191.31.1.89  11151725032

```



```

24030 24155      13.13.13.1/32      Hu0/0/0/19      191.31.1.89      3639895
24031 Aggregate  vrf1601: Per-VRF Aggr[V] \
                                     vrf1601                                     32167647049

```

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



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

This section contains instructions for the following tasks:

### Concept

.

### Configuring ASBRs to Exchange IPv4 Routes and MPLS Labels

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

#### Configuration Example

```

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

```

#### Running Configuration

```

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

```

#### Verification

```

Router#show bgp ipv4 labeled-unicast

```

```

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

```

```

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

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.200.1.1/32	16.1.1.1	0		0	100 ?
*	66.161.1.1	0		0	100 ?
*> 10.200.2.1/32	16.1.1.1	5		0	100 ?
*	66.161.1.1	5		0	100 ?
*> 10.200.5.1/32	16.1.1.1	11		0	100 ?
*	66.161.1.1	11		0	100 ?
*> 10.200.6.1/32	16.1.1.1	4		0	100 ?
*	66.161.1.1	4		0	100 ?
*> 60.200.11.1/32	0.0.0.0	0		32768	?
*>i60.200.12.1/32	60.200.12.1	0	100		0 ?
*>i60.200.13.1/32	60.200.13.1	0	100		0 ?

```

Router#show bgp ipv4 labeled-unicast 10.200.1.1

```

```

BGP routing table entry for 10.200.1.1/32

```

```

Versions:

```

Process	bRIB/RIB	SendTblVer
Speaker	31	31

Local Label: 64006

```

Paths: (2 available, best #1)

```

```

  Advertised to peers (in unique update groups):
    60.200.12.1

```

```

  Path #1: Received by speaker 0

```

```

  Advertised to peers (in unique update groups):
    60.200.12.1

```

```

  100

```

```

    16.1.1.1 from 16.1.1.1 (10.200.1.1)

```

```

      Received Label 3

```

```

      Origin incomplete, metric 0, localpref 100, valid, external, best, group-best,
multipath, labeled-unicast

```

```

      Received Path ID 0, Local Path ID 0, version 31

```

```

      Origin-AS validity: not-found

```

```

Router#show cef vrf default ipv4 10.200.1.1

```

```

10.200.1.1/32, version 161, internal 0x5000001 0x0 (ptr 0x8910c440) [1], 0x0 (0x87f73bc0),
0xa00 (0x88f40118)

```

```

Updated May  3 18:10:47.034

```

```

Prefix Len 32, traffic index 0, precedence n/a, priority 4

```

```

Extensions: context-label:64006

```

```

  via 16.1.1.1/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]

```

```

    path-idx 0 NHID 0x0 [0x889e55a0 0x87b494b0]

```

```

    recursion-via-/32

```

```

    next hop 16.1.1.1/32 via 16.1.1.1/32

```

```

      local label 64006

```

```

      next hop 16.1.1.1/32 Te0/0/1/4/2 labels imposed {ImplNull ImplNull}

```

```

  via 66.161.1.1/32, 3 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]

```

```

    path-idx 1 NHID 0x0 [0x89113870 0x87b493e8]

```

```

    recursion-via-/32

```

```

    next hop 66.161.1.1/32 via 66.161.1.1/32

```

```

    local label 64006
    next hop 66.161.1.1/32 BE161          labels imposed {ImplNull ImplNull}
Router#

```

### Associated Commands

- allocate-label all
- address-family ipv4 labeled-unicast

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

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

### Configuration Example

```

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

```

### Running Configuration

```

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

```

## Verification

```

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

```

### Associated Commands

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

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

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

### Configuration Example

```

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

```

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

### Running Configuration

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

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

This section contains instructions for the following tasks:

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

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

#### SUMMARY STEPS

1. **configure**
2. **router bgp** *autonomous-system-number*
3. **address-family** { **ipv4 tunnel** }
4. **address-family** { **vpnv4 unicast** }
5. **neighbor** *ip-address*
6. **remote-as** *autonomous-system-number*
7. **address-family** { **vpnv4 unicast** }
8. **route-policy** *route-policy-name* { **in** }
9. **route-policy** *route-policy-name* { **out** }

10. **neighbor** *ip-address*
11. **remote-as** *autonomous-system-number*
12. **update-source** *type interface-path-id*
13. **address-family** { **ipv4 tunnel** }
14. **address-family** { **vpn4 unicast** }
15. Use the **commit** or **end** command.

## DETAILED STEPS

---

### Step 1 **configure**

**Example:**

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

Enters the XR Config mode.

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

**Example:**

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

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

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

**Example:**

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

Configures IPv4 tunnel address family.

### Step 4 **address-family** { **vpn4 unicast** }

**Example:**

```
RP/0/RP0/CPU0:router(config-bgp-af)# address-family vpn4 unicast
```

Configures VPNv4 address family.

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

**Example:**

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

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

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

**Example:**

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

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

**Step 7** **address-family** { *vpn4 unicast* }

**Example:**

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family vpn4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)#
```

Configures VPNv4 address family.

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

**Example:**

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

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

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

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

**Example:**

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

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

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

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

**Example:**

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

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

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

**Example:**

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

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

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

**Example:**

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

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



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

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

Configures IPv4 tunnel address family.

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

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

Configures VPNv4 address family.

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

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

**end** - Prompts user to take one of these actions:

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

---

## Configuring a Static Route to an ASBR Peer

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

### SUMMARY STEPS

1. **configure**
2. **router static**
3. **address-family ipv4 unicast**
4. **A.B.C.D/length next-hop**
5. Use the **commit** or **end** command.

### DETAILED STEPS

**Step 1**     **configure****Example:**

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

Enters the XR Config mode.

**Step 2**     **router static****Example:**

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

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

Enters router static configuration mode.

### Step 3 address-family ipv4 unicast

#### Example:

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

Enables an IPv4 address family.

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

#### Example:

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

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

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

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

**end** - Prompts user to take one of these actions:

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

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

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



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

### SUMMARY STEPS

1. **configure**
2. **router bgp** *autonomous-system-number*
3. **bgp confederation peers** *peer autonomous-system-number*
4. **bgp confederation identifier** *autonomous-system-number*

5. **address-family vpnv4 unicast**
6. **neighbor ip-address**
7. **remote-as autonomous-system-number**
8. **address-family vpnv4 unicast**
9. **route-policy route-policy-name in**
10. **route-policy route-policy-name out**
11. **next-hop-self**
12. Use the **commit** or **end** command.

## DETAILED STEPS

### Step 1 **configure**

#### Example:

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

Enters XR Config mode.

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

#### Example:

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

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

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

#### Example:

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

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

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

#### Example:

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

Specifies the autonomous system number for the confederation ID.

### Step 5 **address-family vpnv4 unicast**

#### Example:

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

Configures VPNv4 address family.

**Step 6** `neighbor ip-address`**Example:**

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

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

**Step 7** `remote-as autonomous-system-number`**Example:**

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

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

**Step 8** `address-family vpnv4 unicast`**Example:**

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

Configures VPNv4 address family.

**Step 9** `route-policy route-policy-name in`**Example:**

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

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

**Step 10** `route-policy route-policy-name out`**Example:**

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

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

**Step 11** `next-hop-self`**Example:**

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

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

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

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

**end** - Prompts user to take one of these actions:

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

---

## Configuring MPLS Forwarding for ASBR Confederations

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



---

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

---

### SUMMARY STEPS

1. **configure**
2. **router bgp** *as-number*
3. **mpls activate**
4. **interface** *type interface-path-id*
5. Use the **commit** or **end** command.

### DETAILED STEPS

---

**Step 1**     **configure**

**Example:**

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

Enters XR Config mode.

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

**Example:**

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

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

**Step 3**     **mpls activate**

**Example:**

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

Enters BGP MPLS activate configuration mode.

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

**Example:**

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

Enables MPLS on the interface.

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

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

**end** - Prompts user to take one of these actions:

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

## Configuring a Static Route to an ASBR Confederation Peer

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

### SUMMARY STEPS

1. **configure**
2. **router static**
3. **address-family ipv4 unicast**
4. **A.B.C.D/length next-hop**
5. Use the **commit** or **end** command.

### DETAILED STEPS

**Step 1** `configure`

**Example:**

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

Enters XR Config mode.

**Step 2** `router static`

**Example:**

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

Enters router static configuration mode.

**Step 3** address-family ipv4 unicast**Example:**

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

Enables an IPv4 address family.

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

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

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

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

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

**end** - Prompts user to take one of these actions:

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

## VRF-lite

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

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

## Configure VRF-lite

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

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

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

## Configuration Example

- Create VRF:

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

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

- Configure VRF under the interface:

```
Router#configure
Router(config)#
Router(config-subif)#vrf vrf1
Router(config-subif)#ipv4 address 192.0.2.2 255.255.255.252
Router(config-subif)#encapsulation dot1q 2001
Router(config-subif)#exit

Router(config)#
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 rip
Router(config-rip)#vrf vrf1
Router(config-rip-vrf)#
Router(config-rip-vrf-if)#exit
Router(config-rip-vrf)#
Router(config-rip-vrf-if)#exit
Router(config-rip-vrf)#default-information originate
Router(config-vrf-import-rt)#commit
```

Similarly configure vrf2 under rip, with

## 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
vrf vrf1
ipv4 address 192.0.2.2 255.255.255.252
encapsulation dot1q 2001
!

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

interface
vrf vrf1
ipv4 address 203.0.113.2 255.255.255.252
encapsulation dot1q 2001
!

interface
vrf vrf2
ipv4 address 203.0.113.5 255.255.255.252
encapsulation dot1q 2000
!

/* Routing Protocol Configuration */
router rip
interface Loopback0
!
interface
!
interface
!
interface
!
interface
!
interface
!
interface
!
```

```

vrf vrf1
  interface
  !
  interface
  !
  default-information originate
  !
vrf vrf2
  interface
  !
  interface
  !
  default-information originate
  !

```

## Verification

Router#show route vrf vrf1

Mon Jul 4 19:12:54.739 UTC

```

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

```

Gateway of last resort is not set

```

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

```

Router#show route vrf vrf2

Mon Jul 4 19:12:59.121 UTC

```

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

```

Gateway of last resort is not set

```

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

```

**Related Topics**

- [VRF-lite, on page 47](#)

**Associated Commands**

- [import route-target](#)
- [export route-target](#)
- [vrf](#)

## MPLS L3VPN Services using Segment Routing

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

## Configure MPLS L3VPN over Segment Routing

**Topology**

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

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

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

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

**Configuration**

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

- Configure protocol support on PE-CE (refer, [Connect MPLS VPN Customers, on page 18](#) )
- Configure protocol support on PE-PE (refer, [Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 14](#))

## 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 mpls
Router-PE1 (config-ospf)#segment-routing mpls sr-prefer
Router-PE1 (config-ospf)#segment-routing prefix-sid-map advertise-local
Router-PE1 (config-ospf)#exit
Router-PE1 (config-ospf)#area 1
Router-PE1 (config-ospf-ar)#interface HundredGigE0/0/0/2
Router-PE1 (config-ospf-ar-if)#exit
Router-PE1 (config-ospf-ar)#interface Loopback0
Router-PE1 (config-ospf-ar-if)#prefix-sid index 1
Router-PE1 (config-ospf-ar-if)#commit

/ * Configure segment routing global block */

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

/* Configure Segment Routing using ISIS */

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

```

## Running Configuration

### PE1:

```

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

configure
  segment-routing
    global-block 180000 200000
  !
!

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

```

### P node:

```

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

```

```

interface HundredGigE0/0/1/0
!
interface HundredGigE0/0/1/1
!
interface Loopback0
  prefix-sid index 1
!
!
!
!
!
configure
  segment-routing
    global-block 180000 200000
!
!
configure
  router isis ring
  net 49.0001.1921.6800.1002.00
  nsr
  distribute link-state
  nsf cisco
  address-family ipv4 unicast
  metric-style wide
  mpls traffic-eng level-1
  mpls traffic-eng router-id Loopback0
  segment-routing mpls
!
interface Loopback0
  address-family ipv4 unicast
  prefix-sid index 30102
!
!
!

```

**PE2:**

```

router ospf dc-sr
  router-id 20.20.20.1
  segment-routing mpls
  segment-routing forwarding mpls
  mpls ldp sync
  mpls ldp auto-config
  segment-routing mpls
  segment-routing mpls sr-prefer
  segment-routing prefix-sid-map receive
  segment-routing prefix-sid-map advertise-local
!
area 0
  interface HundredGigE0/0/0/19
  !
  interface Loopback0
    prefix-sid index 1
  !
!
!
!
configure
  segment-routing
    global-block 180000 200000
!
!
configure
  router isis ring

```

```

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

### Related Topics

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

- [Connect MPLS VPN Customers, on page 18](#)
- [Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 14](#)

### Associated Commands

- [index](#)
- [prefix-sid](#)
- [router isis](#)
- [router ospf](#)
- [segment-routing](#)

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

## Verify MPLS L3VPN Configuration over Segment Routing

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

#### P node:

```

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

Also, refer [Verify MPLS L3VPN Configuration, on page 28](#) for a detailed list of commands and sample outputs.

## Single Pass GRE Encapsulation Allowing Line Rate Encapsulation

Single Pass GRE Encapsulation Allowing Line Rate Encapsulation feature, also known as Prefix-based GRE Tunnel Destination for Load Balancing feature, enables line rate GRE encapsulation traffic and enables flow entropy. Data-plane forwarding performance supports full line rate, which is adjusted to consider added encapsulation. GRE tunnel goes down if the destination is not available in RIB. Routing over GRE Single-pass tunnel is not supported in Release 6.3.2, so the traffic that is eligible for GRE encapsulation is identified using an ACL filter that is based on GRE encapsulation. GRE tunnel destination address is an anycast address. All of the GRE encapsulation must be assigned based upon either an ACL or a policy-map, or both. Destinations may be individual addresses or /28 prefixes.

## Configuration

Perform the following tasks to configure the GRE Single-Pass Entropy feature:

- GRE Single-pass
- GRE Entropy(ECMP/UCMP)

```
/* GRE Single-Pass */

Router# configure
Router(config)# interface tunnel-ip30016
Router(config-if)# ipv4 address 216.1.1.1 255.255.255.0
Router(config-if)# ipv6 address 216:1:1::1/64
Router(config-if)# ipv6 enable
Router(config-if)# tunnel mode gre ipv4 encap
Router(config-if)# tunnel source Loopback22
Router(config-if)# tunnel destination 170.170.170.22
Router(config-if)# commit
```



```

Router(config-if)# exit

/* GRE Entropy(ECMP/UCMP)*/

ECMP (ISIS)

Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# commit
!

/* UCMP (ISIS) */

Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# ucmp
Router(config-isis-af)# metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# redistribute connected
Router(config-isis-af)# commit
Router(config-isis-af)# exit
!

Router# configure
Router(config)# interface Bundle-Ether3
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 20
Router(config-af)# commit
Router(config-af)# exit
!

Router# configure
Router(config)# interface Bundle-Ether111
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 15
Router(config-af)# commit
Router(config-af)# exit
!

/* ECMP (OSPF) */

```

```

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# address-family ipv4 unicast
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface HundredGigE0/7/0/23
Router(config-ospf-af-ar-if)# commit
Router(config-ospf-af-ar-if)# exit

/* UCMP (OSPF) */

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# ucmp
Router(config-ospf)# address-family ipv4 unicast
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3 cost 2
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112 cost 2
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface HundredGigE0/7/0/23
Router(config-ospf-af-ar-if)# commit
Router(config-ospf-af-ar-if)# exit

/* ECMP (BGP) */
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# bgp bestpath as-path multipath-relax
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# network 170.170.170.3/32
Router(config-bgp-af)# network 170.170.170.10/32
Router(config-bgp-af)# network 170.170.170.11/32

```

```

Router(config-bgp-af) # network 170.170.172.3/32
Router(config-bgp-af) # network 180.180.180.9/32
Router(config-bgp-af) # network 180.180.180.20/32
Router(config-bgp-af) # network 180.180.180.21/32
Router(config-bgp-af) # network 180.180.180.24/32
Router(config-bgp-af) # network 180.180.180.25/32
Router(config-bgp-af) # commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# neighbor 4.1.1.2
Router(config-bgp-nbr)# remote-as 300
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 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
!

/* UCMP (BGP) */

Router# configure
Router(config)# router bgp 800
Router(config-bgp)# bgp bestpath as-path multipath-relax
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# maximum-paths ebgp 5
Router(config-bgp-af)# network 180.180.180.9/32
Router(config-bgp-af)# network 180.180.180.20/32
Router(config-bgp-af)# network 180.180.180.21/32
Router(config-bgp-af)# network 180.180.180.24/32
Router(config-bgp-af)# network 180.180.180.25/32
Router(config-bgp-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# neighbor 7.1.5.2
Router(config-bgp-nbr)# remote-as 4000
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy TRANSITO_IN in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# 4.1.111.2
Router(config-bgp-nbr)# remote-as 4000
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy TRANSITO_IN in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# commit
!

/* Configure rouverte policy */

Router# configure
Router(config)# route-policy TRANSITO_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:1250000)
Router(config-rpl-if)# else

```

```

Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
!

Router# configure
Router(config)# route-policy TRANSIT1_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:37500000
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy

```

## Running Configuration

```

/* GRE Single-Pass configuration */

interface tunnel-ip30016
ipv4 address 216.1.1.1 255.255.255.0
ipv6 address 216:1:1::1/64
ipv6 enable
tunnel mode gre ipv4 encap
tunnel source Loopback22
tunnel destination 170.170.170.22
!

/* GRE Entropy(ECMP/UCMP) */

ECMP (ISIS)

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
address-family ipv4 unicast
metric-style wide
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5
!

/* UCMP(ISIS) */

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
address-family ipv4 unicast
metric-style wide
ucmp
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5

```

```

redistribute connected
!
interface Bundle-Ether3
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 20
!

interface Bundle-Ether111
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 15
!

!

/* ECMP (OSPF) */

router ospf 3
nsr
maximum paths 5
address-family ipv4 unicast
area 0
interface Bundle-Ether3
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
!
interface Loopback23
!
interface HundredGigE0/7/0/23
!
!
!
/* UCMP (OSPF) */

router ospf 3
nsr
maximum paths 5
ucmp
address-family ipv4 unicast
area 0
interface Bundle-Ether3
cost 2
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
cost 2
!
interface Loopback23
!
interface HundredGigE0/7/0/23
!
!
!

/* ECMP (BGP) */

```

```

router bgp 800
  bgp bestpath as-path multipath-relax
  address-family ipv4 unicast
  maximum-paths ebgp 5
  network 170.170.170.3/32
  network 170.170.170.10/32
  network 170.170.170.11/32
  network 170.170.172.3/32
  network 180.180.180.9/32
  network 180.180.180.20/32
  network 180.180.180.21/32
  network 180.180.180.24/32
  network 180.180.180.25/32
  !
  neighbor 4.1.1.2
  remote-as 300
  address-family ipv4 unicast
  route-policy PASS-ALL in
  route-policy PASS-ALL out
  next-hop-self
  !
  !

/* UCMP(BGP) */

router bgp 800
  bgp bestpath as-path multipath-relax
  address-family ipv4 unicast
  maximum-paths ebgp 5
  network 180.180.180.9/32
  network 180.180.180.20/32
  network 180.180.180.21/32
  network 180.180.180.24/32
  network 180.180.180.25/32
  !

  neighbor 7.1.5.2
  remote-as 4000
  address-family ipv4 unicast
  route-policy TRANSIT0_IN in
  route-policy PASS-ALL out
  next-hop-self
  !
  !
  neighbor 4.1.111.2
  remote-as 4000
  address-family ipv4 unicast
  route-policy TRANSIT1_IN in
  route-policy PASS-ALL out
  next-hop-self
  !
  !

/* Configure route policy */

route-policy TRANSIT0_IN
  if destination in (170.170.170.24/32) then
  set extcommunity bandwidth (2906:1250000)
  else
  pass
  endif
end-policy
!
route-policy TRANSIT1_IN

```

```

if destination in (170.170.170.24/32) then
set extcommunity bandwidth (2906:37500000)
else
pass
endif
end-policy
!

```

## Verification

Verify if the tunnel mode GRE encapsulation is enabled.

```
Router# show int tunnel-ip2
```

```

interface tunnel-ip2
  ipv4 address 80.80.82.1 255.255.255.0
  ipv6 address 2000:80:80:82::1/64
  load-interval 30
  tunnel mode gre ipv4 encap
  tunnel source Loopback4
  tunnel destination 11.4.2.2
!

```

```

RP/0/RP0/CPU0:PE1_5516#show int tunnel-ip2
tunnel-ip2 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Tunnel
  Internet address is 80.80.82.1/24
  MTU 1500 bytes, BW 100 Kbit (Max: 100 Kbit)
    reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation TUNNEL_IP, loopback not set,
  Last link flapped 1d18h
  Tunnel TOS 0
  Tunnel mode GRE IPV4, encap
  Keepalive is disabled.
  Tunnel source 11.11.12.1 (Loopback4), destination 11.4.2.2/32
  Tunnel TTL 255
  Last input never, output never
  Last clearing of "show interface" counters 14:53:37
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 total input drops
    0 drops for unrecognized upper-level protocol
  Received 0 broadcast packets, 0 multicast packets
    0 packets output, 0 bytes, 0 total output drops
  Output 0 broadcast packets, 0 multicast packets

```

Verify if the tunnel mode GRE encapsulation and decapsulation are enabled.

```
Router# sh interfaces tunnel-ip 5 accounting
```

```
Wed May 16 01:50:57.258 UTC
```

```
tunnel-ip5
```

Protocol	Pkts In	Chars In	Pkts Out	Chars Out
IPV4_UNICAST	489	55746	0	0
IPV6_UNICAST	489	55746	0	0
MPLS	587	69266	0	0

Verify if the recycle of the packets are not done under Recycle VoQ: 48:

```
Router# show tunnel ip ea summary location 0/7/CPU0
```

```

Number of tunnel updates to retry: 0
Number of tunnel updates retried: 0
Number of tunnel retries failed: 0

```

```

Platform:
Recycle VoQ: 48
      ReceivedBytes  ReceivedPackets  ReceivedKbps
      DroppedBytes  DroppedPackets  DroppedKbps

NPU 0:0      0              0              0
             0              0              0
1            0              0              0
             0              0              0
2            0              0              0
             0              0              0
3            0              0              0
             0              0              0
...
NPU 1:0      0              0              0
             0              0              0
1            0              0              0
             0              0              0
2            0              0              0
             0              0              0
3            0              0              0
             0              0              0

NPU 2:0      0              0              0
             0              0              0
1            0              0              0
             0              0              0
2            0              0              0
             0              0              0
3            0              0              0
             0              0              0

```

Verify if the tunnel mode GRE encapsulation is enabled.

```
Router# show interfaces tunnel-ip * brief
```

```

Thu Sep 7 00:04:39.125 PDT
Intf Intf LineP Encap MTU BW
Name   State   State  Type      (byte) (Kbps)
-----
ti30001 down    down   TUNNEL_IP 1500  100
ti30002 up      up     TUNNEL_IP 1500  100

```

Verify the tunnel endpoint route in RIB.

```
Router# show route 10.1.1.1
```

```

Routing entry for 10.0.0.0/8
Known via "static", distance 1, metric 0 (connected)
Installed Oct 2 15:50:56.755 for 00:39:24
Routing Descriptor Blocks
  directly connected, via tunnel-ip109
  Route metric is 0, Wt is 1
  No advertising protos.

```

Verify if the tunnel mode GRE encapsulation is enabled.

```
Router# show tunnel ip ea database tunnel-ip 109 location 0/7/CPU0
```

```

----- node0_0_CPU0 -----
tunnel ifhandle 0x80022cc
tunnel source 161.115.1.2
tunnel destination 162.1.1.1/32
tunnel transport vrf table id 0xe0000000
tunnel mode gre ipv4, encap
tunnel bandwidth 100 kbps

```



```

tunnel platform id 0x0
tunnel flags 0x40003400
IntfStateUp
BcStateUp
Ipv4Caps
Encap
tunnel mtu 1500
tunnel tos 0
tunnel ttl 255
tunnel adjacency flags 0x1
tunnel o/p interface handle 0x0
tunnel key 0x0, entropy length 0 (mask 0xffffffff)
tunnel QT next 0x0
tunnel platform data (nil)
Platform:
Handle: (nil)
Decap ID: 0
Decap RIF: 0
Decap Recycle Encap ID: 0x00000000
Encap RIF: 0
Encap Recycle Encap ID: 0x00000000
Encap IPv4 Encap ID: 0x4001381b
Encap IPv6 Encap ID: 0x00000000
Encap MPLS Encap ID: 0x00000000
DecFEC DecRcyLIF DecStatsId EncRcyLIF

```

Verify if the QoS table is updated properly.

```

Router# show controllers npu stats voq base 48 instance all location
0/0/CPU0
Asic Instance = 0
VOQ Base = 48

```

	ReceivedPkts	ReceivedBytes	DroppedPkts	DroppedBytes
COS0 =	0	0	0	0
COS1 =	0	0	0	0
COS2 =	0	0	0	0
COS3 =	0	0	0	0

```

Asic Instance = 1
VOQ Base = 48

```

	ReceivedPkts	ReceivedBytes	DroppedPkts	DroppedBytes
COS0 =	0	0	0	0
COS1 =	0	0	0	0
COS2 =	0	0	0	0
COS3 =	0	0	0	0

```

Asic Instance = 2
VOQ Base = 48

```

	ReceivedPkts	ReceivedBytes	DroppedPkts	DroppedBytes
COS0 =	0	0	0	0
COS1 =	0	0	0	0
COS2 =	0	0	0	0
COS3 =	0	0	0	0

# Implementing MPLS L3VPNs - References

## MPLS L3VPN Benefits

MPLS L3VPN provides the following benefits:

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

## Major Components of MPLS L3VPN—Details

### Virtual Routing and Forwarding Tables

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

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

These components are collectively called a VRF instance.

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

Packet forwarding information is stored in the IP routing table and the FIB table for each VRF. A separate set of routing and FIB tables is maintained for each VRF. These tables prevent information from being

forwarded outside a VPN and also prevent packets that are outside a VPN from being forwarded to a router within the VPN.

## VPN Routing Information: Distribution

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

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

## BGP Distribution of VPN Routing Information

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

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

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

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

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

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

## MPLS Forwarding

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

A PE router binds a label to each customer prefix learned from a CE router and includes the label in the network reachability information for the prefix that it advertises to other PE routers. When a PE router forwards a packet received from a CE router across the provider network, it labels the packet with the label learned from the destination PE router. When the destination PE router receives the labeled packet, it pops the label

and uses it to direct the packet to the correct CE router. Label forwarding across the provider backbone is based on dynamic label switching. A customer data packet carries two levels of labels when traversing the backbone:

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

## Automatic Route Distinguisher Assignment

To take advantage of iBGP load balancing, every network VRF must be assigned a unique route distinguisher. VRFs 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.

## Layer 3 QinQ

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

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

This feature supports:

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

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

### Types of Subinterfaces

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

### Restrictions

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

## Configure Layer 3 QinQ

### Configuration Example

Perform this task to configure the Layer 3 QinQ feature.

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

### Running Configuration

This section shows the running configuration of Layer 3 QinQ.

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

### Verification

Verify Layer 3 QinQ configuration.

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

### Related Topics

- [Layer 3 QinQ](#) , on page 68

### Associated Commands

- show interfaces



## CHAPTER 3

# Implementing IPv6 VPN Provider Edge Transport over MPLS

IPv6 Provider Edge or IPv6 VPN Provider Edge (6PE/VPE) uses the existing MPLS IPv4 core infrastructure for IPv6 transport. 6PE/VPE 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/VPE.

- [Overview of 6PE/VPE, on page 71](#)
- [Benefits of 6PE/VPE, on page 72](#)
- [Deploying IPv6 over MPLS Backbones, on page 72](#)
- [IPv6 on the Provider Edge and Customer Edge Routers, on page 72](#)
- [OSPFv3 \(CE to PE\), on page 73](#)
- [Restrictions for 6VPE, on page 74](#)
- [Configuring 6PE/VPE, on page 74](#)
- [Configuring OSPFv3 as the Routing Protocol Between the PE and CE Routers, on page 78](#)

## Overview of 6PE/VPE

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/VPE

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

- Minimal operational cost and risk—No impact on existing IPv4 and MPLS services.
- Provider edge routers upgrade only—A 6PE/VPE 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/VPE 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

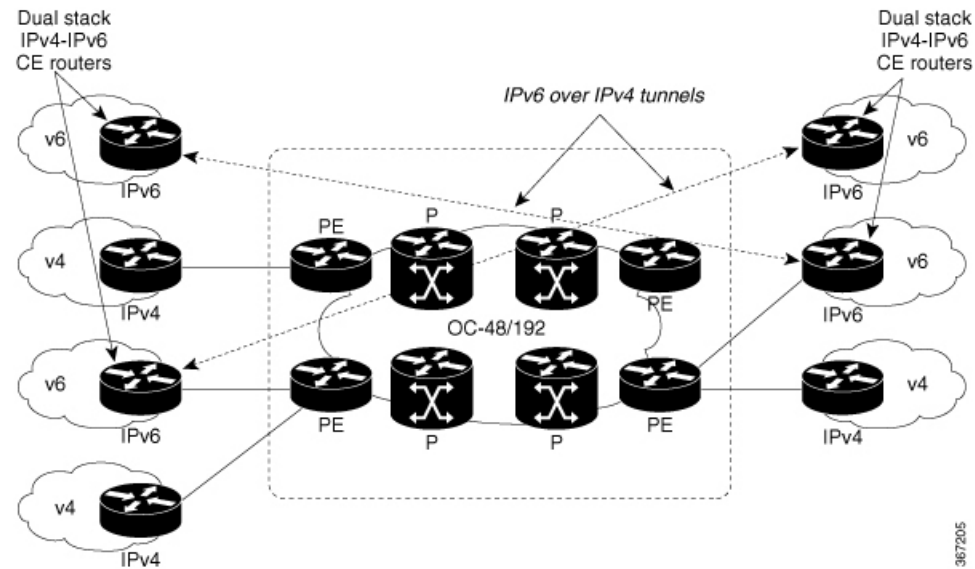
Using tunnels on the CE routers is the simplest way to deploy IPv6 over MPLS networks. It has no impact on the operation or infrastructure of MPLS and requires no changes to the P routers in the core or to the PE



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 10: 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.

## Restrictions for 6VPE

The restrictions applicable for configuring 6VPE are as follows:

- The 6VPE feature does not work with the following configuration:

**hw-module profile sr-policy v6-null-label-autopush**

## Configuring 6PE/VPE

### 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.




---

**Note** 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.

The default allocation mode is per-prefix. While configuring 6PE/VPE on the router, to achieve the required scale value, use per-vrf or per-ce for all routers including peer routers.

---




---

**Note** Route policies must be configured prior to configuring 6PE/VPE.

---

```
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) #label mode per-vrf
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) #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) #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
label mode per-vrf
!
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     FULL/DROTHER    00:00:36   0             TenGigE0/7/0/0/3.7
    Neighbor is up for 1w0d

Total neighbor count: 1

```



## CHAPTER 4

# Implementing DCI Layer 3 Gateway between MPLS-VPN and EVPN Data Center

---

This chapter module provides conceptual and configuration information for Data Center Interconnect (DCI) Layer 3 Gateway between MPLS-VPN and EVPN Data Center.

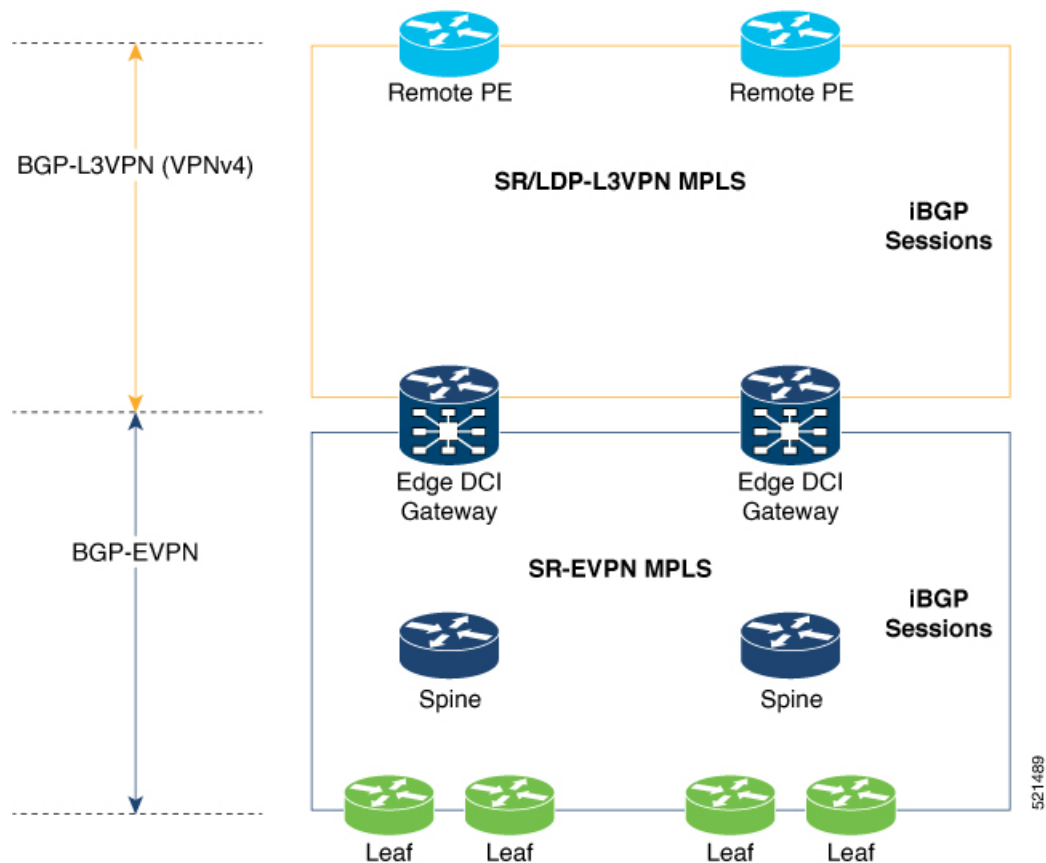
- [Data Center Interconnect between MPLS-VPN and EVPN-MPLS](#) , on page 79

## Data Center Interconnect between MPLS-VPN and EVPN-MPLS

This part provides conceptual and configuration information for Data Center Interconnect (DCI) Layer 3 Gateway with EVPN-MPLS on Cisco NCS 5500 Series Router.

### DCI Layer 3 Gateway with EVPN-MPLS

You can use SR-EVPN for Data Center on routers for a spine-leaf architecture with edge devices such as border leaf. DCI L3 stitching allows Data Centers that run SR-EVPN to communicate with legacy and existing MPLS VPN (VPNv4) sites.



In this topology,

Leaf (ToR) – Router acts as both access switch and distributed PE. Leaf establishes BGP EVPN neighborhood with Spine route-reflector (RR). This router sends and receives prefixes from the DCI Gateway. Leaf ToR provides the following types of services:

- Regular L3 VRF configuration using subinterfaces to attach some CE devices. Traditional PE-CE scenario without EVPN configuration.
- L3 EVPN VRF using L2VPN configuration to attach multiple Data Centers services.

Leaf sends and receives prefixes from or to the DCI gateway:

- Leaf sends prefixes to DCI: Leaf re-originates local learned VRF subnet route as EVPN Route Type 5 with the EVPN RT (stitching-rt or regular RT), then sends to Spine RR. Spine RR sends prefixes to DCI gateway.
- Leaf receives prefixes from DCI: Leaf receives EVPN Route Type 5 from Spine RR that is re-originated at DCI gateway due to stitching between VPNv4 and EVPN. Leaf imports remote VPNv4 prefixes to local VRF matching VPNv4 RT (stitching-rt or regular RT).

Spine RR: Spine RR establishes BGP EVPN neighborhood with Leaf (ToR) and Edge DCI Gateway serving as Route-Reflector for EVPN prefixes between the devices in the Data Center. Leaf and DCI Gateway must be configured as clients of Spine RR.



Edge (DCI gateway): Edge (DCI gateway) acts as an edge router that allows communication between services connected at Leaf and CEs in legacy MPLS network architecture. The edge DCI gateway establishes BGP EVPN neighborship with Spine RR and remote PEs, or RR depending on legacy MPLS network architecture.

The edge DCI gateway sends and receives prefixes from or to the Data Center:

- DCI gateway receives prefixes from legacy MPLS VPNv4 network and sends prefixes to Leaf: DCI gateway receives L3VPN (VPNv4) routes from remote MPLS VPN (VPNv4) PE or RR depending on legacy MPLS network architecture matching the VPNv4 RT (stitching-rt or regular RT). Then re-originate these prefixes as EVPN Route Type 5 with the EVPN RT (stitching-rt or regular RT) advertising to Spine RR due to BGP EVPN neighbor with the Spine.
- DCI gateway receives prefixes from Leaf and sends prefixes to legacy MPLS VPNv4 network: DCI gateway receives EVPN Route Type 5 originated from Leaf (ToR) by Spine RR due to BGP EVPN neighbor with the Spine. Leaf and DCI gateway does not have a direct BGP neighborship. Then import the routes to local VRF matching the EVPN RT (stitching-rt or regular RT) and re-originate this prefix as VPNv4 router with the VPNv4 RT (stitching-rt or regular RT) and advertise to remote MPLS VPN (VPNv4) PE or RR depending on legacy MPLS network architecture.

Remote PE: Remote PE receives traditional MPLS L3VPN prefixes (VPNv4) by DCI Gateway or RR depending on legacy MPLS network architecture. You must have a unique Route-Distinguisher (RD) between remote PEs and DCI gateway to allow stitching re-originate prefixes from VPNv4 to EVPN at DCI Gateway.

Stitching RTs and Regular RTs can be assigned to any side, EVPN or VPNv4, irrespective of the address-family. Consider the following supported scenarios:

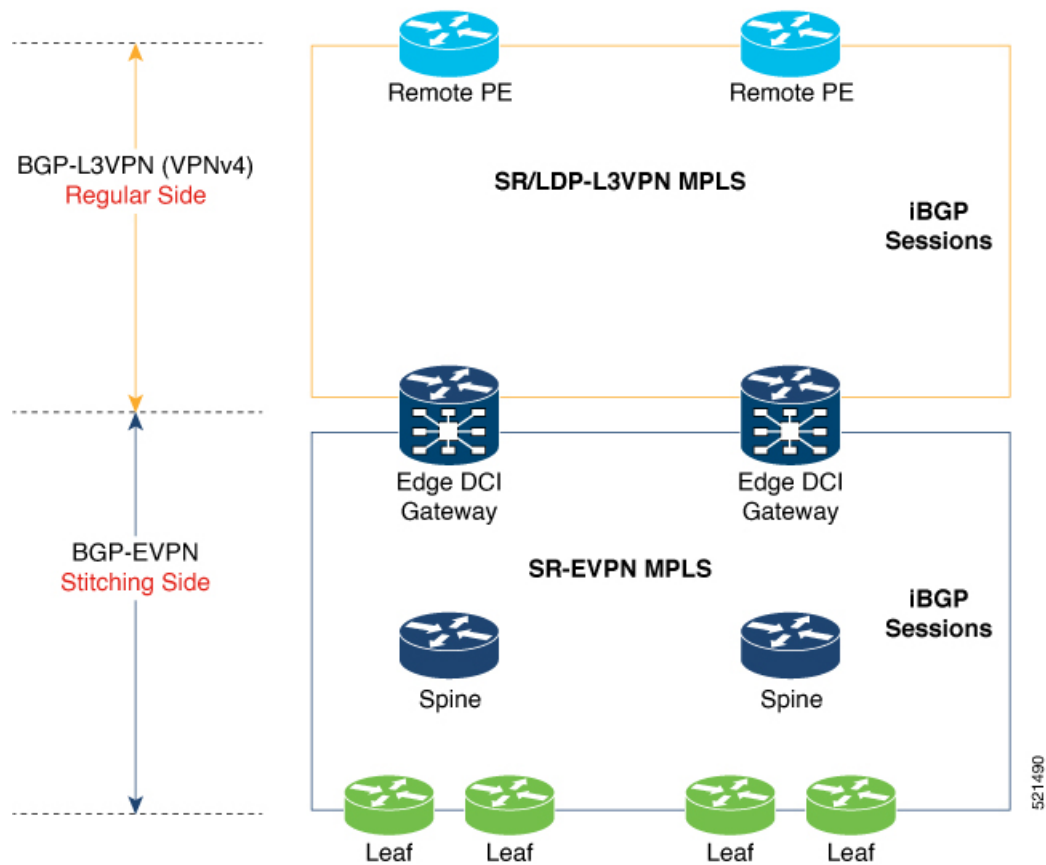
## VPNv4-Regular RT and EVPN-Stitching RT

For each VRF on the DCI gateway, there are two sets of manually configured import and export route-targets for VPNv4 as a regular side and EVPN as a stitching side. Consider the following sets:

- Data Center Route-Targets for EVPN associated with EVPN BGP neighbor (Stitching RT).
- MPLS L3VPN Route-Targets for VPNv4 or VPNv6 associated with L3VPN BGP neighbor (Regular RT).

This separation of RTs enables the two sets of RTs to be independently configured. The RTs associated with the EVPN BGP neighbor require **stitching-rt** keyword under VRF configuration. The route-types associated with the L3VPN BGP neighbor do not require the keyword.

The following topology shows regular/normal and stitching side.



### Route Targets

The RTs associated with the EVPN BGP neighbor are labelled as stitching RTs. The RTs associated with the L3VPN BGP neighbor are normal RTs.

### Route Re-Origination

Consider control plane information propagation by the edge DCI gateway from the L3VPN (regular/normal side) to the Data Center (stitching side). Edge DCI gateway advertises to its BGP EVPN neighbor the routes that are re-originated after importing them from the L3VPN BGP neighbor. For this case of VPNv4 or VPNv6 routes being propagated to the BGP EVPN neighbors (Data Center neighbors), re-originating the routes refers to replacing the normal route-targets with the local route-target values (stitching-rt) associated with the BGP EVPN neighbors.

### Route Address-Family and Encoded Address-Family

When an address-family is configured for a BGP neighbor, it means that the specified address-family routes encoded with the NLRI for that address-family are advertised to the neighbor. This does not hold for Data Center BGP neighbors because they use only EVPN address-family. Here, BGP neighbors advertise VPNv4 or VPNv6 unicast routes using the EVPN NLRI encoding. Thus, the encoded address-family and route address family can be possibly different. You can advertise the VPNv4 or VPNv6 address-family using the **advertise vpnv4 unicast** or **advertise vpnv6 unicast** command. For example, an EVPN address-family BGP neighbor configured with the **advertise vpnv4 unicast** command sends VPNv4 unicast routes in an EVPN encoded NLRI.

### Local VPNv4 or VPNv6 Route Advertisement

On the edge DCI gateway, the locally sourced VPNv4 or VPNv6 routes (any CE directly connected not using L2VPN with BD/EVI/BVI, using only regular L3 VRF) can be advertised to the BGP EVPN neighbors with the normal route targets (RTs) configured for the VRF or the stitching RTs associated with the BGP EVPN neighbors. By default, these routes are advertised with the normal route targets. You can configure this local VPNv4 or VPNv6 route advertisements to be advertised with stitching RTs to the BGP EVPN neighbors by using the **advertise vpnv4 unicast local stitching-rt** or **advertise vpnv6 unicast local stitching-rt** command as required.

VPNv4 neighbors do not require any additional configuration. By default, these routes are advertised with the normal route-targets to BGP L3VPN neighbors.

### Route Distinguishers

The Router Distinguisher (RD) associated per VRF must be unique per PE in the network. There are few available options to keep unique RD per device:

- Manual configuration: You must manually assign a unique value per device in the network. For example, in this scenario:
  - Leaf (ToR) = RD 1
  - Edge DCI Gateway = RD 2
  - Remote PE = RD 3
- Use **rd auto** command under VRF. 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.



---

**Note** In a DCI deployment, for route re-originate with stitching-rt for a particular VRF, using the same Route Distinguisher (RD) between edge DCI gateway and MPLS-VPN PE or same RD between edge DCI gateway and Leaf (ToR) is not supported.

---

### Configure VPNv4-Regular RT and EVPN-Stitching RT

This section describes tasks to configure VPNv4-Regular RT and EVPN-Stitching RT. Perform the following tasks to complete the configuration:

- Configure Leaf (ToR)
- Configure Spine-RR (Route Reflector)
- Configure Edge DCI Gateway
- Configure EVPN BGP neighbor and route advertisements
- Configure L3VPN BGP neighbor relationship and route advertisements

### Configure Leaf (ToR)

Configure VRF in Leaf (ToR) at BGP-EVPN (Stitching Side) with Stitching-RT.

```

vrf data-center1
  address-family ipv4 unicast
  import route-target
    1:2 stitching                               // BGP - EVPN (Stitching Side)
  !
  export route-target
    1:2 stitching                               // BGP - EVPN (Stitching Side)
  !
router bgp 100
  neighbor 10.10.1.1                           // Spine Loopback IP Address
  address-family l2vpn evpn
    advertise vpnv4 unicast
    advertise vpnv6 unicast
  !

```




---

**Note** Advertise vpnv4/vpnv6 unicast enables local learned regular L3 VRF prefixes to be advertised as EVPN prefixes to BGP – EVPN neighbor. This means any local prefixes such as PE-CE without L2VPN with BD/EVI/BVI configuration. If all the services are pure EVPN with L2VPN with BD/EVI/BVI configuration these commands are not required.

---

### Configure Spine-RR

Configure Spine RR with Leaf (ToR) and edge DCI gateway as RR client for AFI L2VPN EVPN. VRF configuration is not required.

```

// VRF Config is not required //

router bgp 100
  neighbor 10.10.2.1                           // Leaf (ToR) Loopback IP Address
  address-family l2vpn evpn
    route-reflector-client
  !
  neighbor 10.10.3.1                           // Edge DCI Gateway Loopback IP Address
  address-family l2vpn evpn
    route-reflector-client
  !

```

### Configure Edge DCI Gateway

You can configure DCI with the same VRF as Leaf (ToR). Use the same RT as remote PE for L3VPN network or the same VRF if that is possible.

### Configure VRF and Route Targets Import and Export rules

Perform the following steps to configure VRF and define route targets to be used for import and export of forwarding information.

```

vrf data-center1
  address-family ipv4 unicast
  import route-target
    1:1                                         // BGP - L3VPN (Regular/normal Side)

```

```

1:2 stitching                // BGP - EVPN (Stitching Side)
!
export route-target
1:1                          // BGP - L3VPN (Regular/normal Side)
1:2 stitching                // BGP - EVPN (Stitching Side)
!

```

### Configure EVPN BGP Neighbor and Route Advertisements

Perform this task on the edge DCI gateway to configure BGP neighbor relationship and route advertisements with the EVPN BGP neighbor.

```

router bgp 100
 address-family l2vpn evpn
!
 neighbor 10.10.1.1          // Spine Loopback IP Address
 address-family l2vpn evpn
  import stitching-rt re-originate //Imp EVPN 1:2, reoriginate VPNv4 RT 1:1
  advertise vpnv4 unicast re-originated stitching-rt //Send routes EVPN 1:2
  advertise vpnv6 unicast re-originated stitching-rt //Send routes EVPN 1:2
!

```

### Configure L3VPN BGP Neighbor Relationship and Route Advertisements

Perform the following steps to configure BGP neighbor relationship and route advertisements with the L3VPN BGP neighbor.

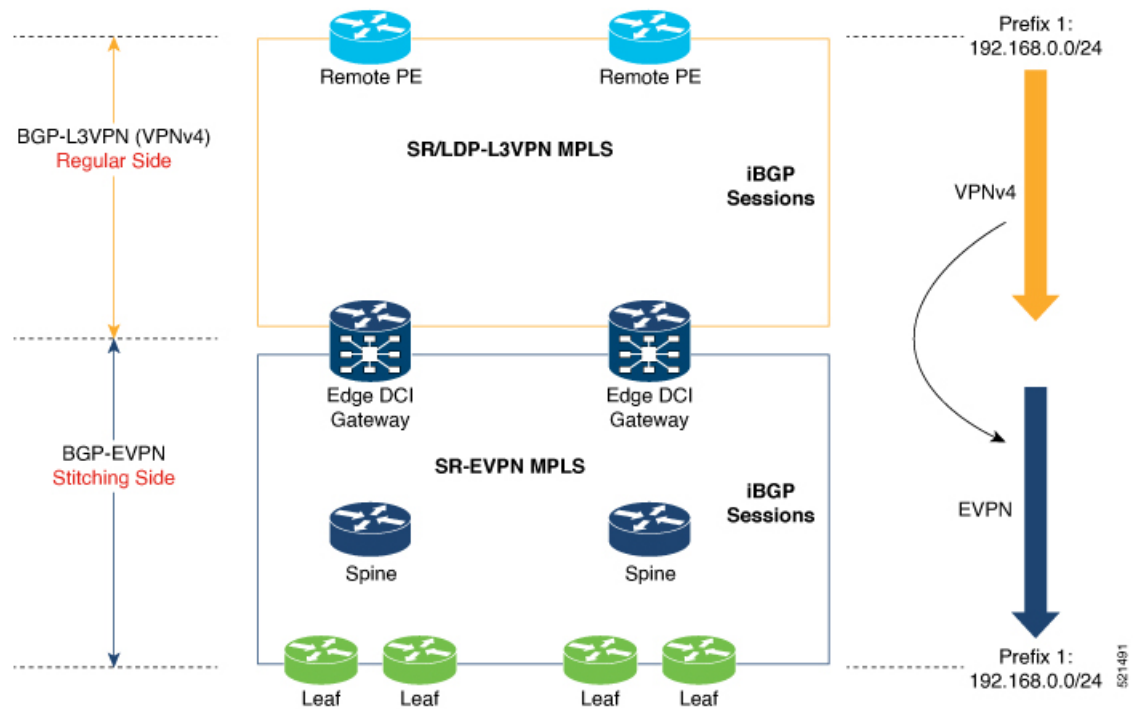
```

router bgp 100
 address-family vpnv4 unicast
!
 neighbor 10.10.1.1          // Spine Loopback IP Address
 address-family vpnv4 unicast // Same config for VPNv6
  import re-originate stitching-rt // Imp VPNv4 1:1, re-originate EVPN 1:2
  advertise vpnv4 unicast re-originated // Send routes VPNv4 RT 1:1
!

```

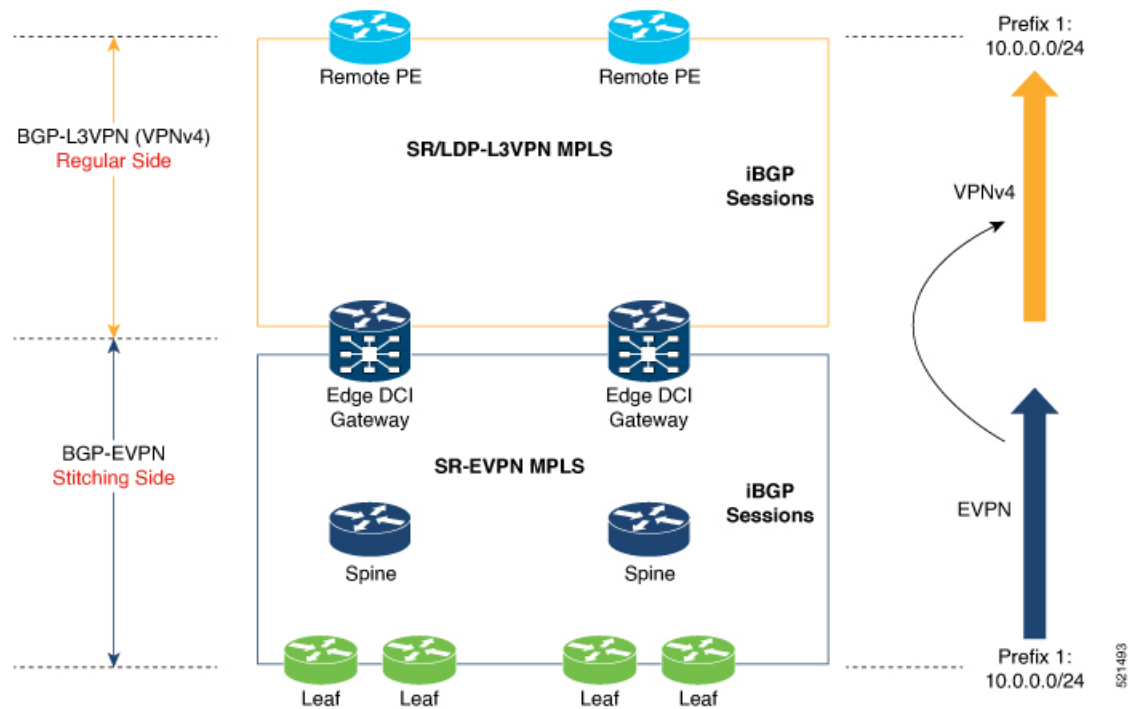
Configuration applies in two directions:

- Stitching from VPNv4 to EVPN routes. Prefixes received from MPLS L3VPN network and re-originated as EVPN prefixes towards Data Center Spine RR and Leaf (ToR).
1. Importing VPNv4 routes with import **re-originate stitching-rt** command under AFI VPNv4 UNICAST. This command imports routes using RT 1:1 and then reoriginate with BGP EVPN 1:2 **stitching-rt**.
  2. Advertising re-originated EVPN routes with VPNv4 RT with advertise **vpnv4 unicast re-originated** command under AFI L2VPN EVPN. This command advertises routes from MPLS L3VPN network (VPNv4) to BGP EVPN neighbors inside Data Center (Spine RR and then Leaf (ToR)), re-originating these routes using BGP EVPN 1:2 **stitching-rt**.



- Stitching from EVPN to VPNv4 routes. Prefixes received from BGP-EVPN Data Center and re-originated as MPLS L3VPN prefixes towards VPNv4 RR or remote PE in L3VPN network.

1. Importing EVPN routes with import **stitching-rt re-originate** command under AFI L2VPN EVPN. This command imports routes using RT 1:2 **stitching-rt** and then re-originate with VPNv4 regular/normal VPNv4 RT 1:1.
2. Advertising re-originated EVPN routes with VPNv4 RT with **advertise vpv4 unicast re-originated** command under AFI VPNv4 UNICAST. This command advertises routes from EVPN Data Center to VPNv4 RR or remote PEs, re-originating these routes using regular/normal VPNv4 RT 1:1.



### Verification of Edge DCI Gateway Configuration

```
Router# show bgp l2vpn evpn
```

```
Fri Aug 21 00:24:10.773 PDT
BGP router identifier 30.30.30.30, local AS number 100
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 16
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 16/0
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: 100:1
*>i [2] [10000] [48] [0226.51bd.c81c] [32] [200::1001]/232
11.0.0.1 100 0 i
*>i [2] [10000] [48] [0226.51bd.c81c] [32] [200:1::1001]/232
11.0.0.1 100 0 i
*>i [2] [10000] [48] [0226.51bd.c81c] [32] [200.1.1.1]/136
11.0.0.1 100 0 i
*>i [2] [10000] [48] [0226.51bd.c81c] [32] [200.1.1.2]/136
11.0.0.1 100 0 i
*>i [5] [4231] [32] [100.1.1.1]/80
11.0.0.1 100 0 i
*>i [5] [4231] [32] [100.1.1.2]/80
11.0.0.1 100 0 i
*>i [5] [4231] [112] [fec0::1001]/176
11.0.0.1 100 0 i
```

```
*>i[5][4232][112][fec0::1:1001]/176
    11.0.0.1                100        0 i
```

Processed 8 prefixes, 8 paths

Router# **show bgp l2vpn evpn rd 100:1 [5][4231][112][fec0::1001]/176 detail**

```
Fri Aug 21 00:34:43.747 PDT
BGP routing table entry for [5][4231][112][fec0::1001]/176, Route Distinguisher: 100:1
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          5         5
  Flags: 0x04040001+0x00000000;
Last Modified: Aug 21 00:16:58.000 for 00:17:46
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Flags: 0x4000600025060005, import: 0x3f
  Not advertised to any peer
  Local
    11.0.0.1 (metric 2) from 20.0.0.1 (11.0.0.1)
      Received Label 16001
      Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
reoriginate, not-in-vrf
      Received Path ID 0, Local Path ID 1, version 5
      Extended community: Flags 0x6: RT:1:1
      Originator: 11.0.0.1, Cluster list: 20.20.20.20
      EVPN ESI: ffff.ffff.ffff.ffff.ff01, Gateway Address : fec0::254
```

Router# **show bgp l2vpn evpn neighbors 20.0.0.1 detail**

```
Fri Aug 21 00:25:37.383 PDT

BGP neighbor is 20.0.0.1
  Remote AS 100, local AS 100, internal link
  Remote router ID 20.20.20.20
  BGP state = Established, up for 00:08:58
  NSR State: NSR Ready
  Last read 00:00:34, Last read before reset 00:00:00
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time: 180, keepalive: 60, min acceptable hold time: 3
  Last write 00:00:36, attempted 19, written 19
  Second last write 00:01:36, attempted 143, written 143
  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 Aug 21 00:25:03.667 last full not set pulse count 33
  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
  Entered Neighbor NSR TCP mode:
    TCP Initial Sync :          Aug 21 00:18:07.291
    TCP Initial Sync Phase Two : Aug 21 00:18:07.319
    TCP Initial Sync Done :      Aug 21 00:18:08.334
  Multi-protocol capability received
  Neighbor capabilities:
    Route refresh:              Yes      Yes
    4-byte AS:                  Yes      Yes
    Address family VPNv4 Unicast: Yes      No
    Address family VPNv6 Unicast: Yes      No
```



```

Address family L2VPN EVPN:      Yes      Yes
Message stats:
  InQ depth: 0, OutQ depth: 0
    Last_Sent      Sent      Last_Rcvd      Rcvd
  Open:           Aug 21 00:16:38.087      1      Aug 21 00:16:40.123      1
  Notification:   ---              0      ---              0
  Update:         Aug 21 00:24:01.421      9      Aug 21 00:24:03.652      13
  Keepalive:      Aug 21 00:25:01.434      8      Aug 21 00:25:03.667      9
  Route_Refresh:  Aug 21 00:24:01.377      3      ---              0
  Total:         ---              21      ---              23
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: VPNv4 Unicast
BGP neighbor version 35
Update group: 0.3 Filter-group: 0.1 No Refresh request being processed
Advertise Reorigination Enabled
Advertise AFI EoR can be sent
Route refresh request: received 0, sent 0
0 accepted prefixes, 0 are bestpaths
Cumulative no. of prefixes denied: 0.
Prefix advertised 4, suppressed 0, withdrawn 0
Maximum prefixes allowed 2097152
Threshold for warning message 75%, restart interval 0 min
AIGP is enabled
An EoR was not received during read-only mode
Last ack version 35, Last synced ack version 35
Outstanding version objects: current 0, max 1
Additional-paths operation: None
Send Multicast Attributes

For Address Family: VPNv6 Unicast
BGP neighbor version 29
Update group: 0.3 Filter-group: 0.1 No Refresh request being processed
Advertise Reorigination Enabled
Advertise AFI EoR can be sent
Route refresh request: received 0, sent 0
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
AIGP is enabled
An EoR was not received during read-only mode
Last ack version 29, Last synced ack version 29
Outstanding version objects: current 0, max 0
Additional-paths operation: None
Send Multicast Attributes
Advertise VPNv4 routes enabled with Reoriginate,Local with stitching-RT option

```

```

For Address Family: L2VPN EVPN
BGP neighbor version 18
Update group: 0.2 Filter-group: 0.1 No Refresh request being processed
Route refresh request: received 0, sent 3
8 accepted prefixes, 8 are bestpaths
Cumulative no. of prefixes denied: 0.
Prefix advertised 4, suppressed 0, withdrawn 6
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 18, Last synced ack version 18
Outstanding version objects: current 0, max 2

```

```

Additional-paths operation: None
Send Multicast Attributes
Advertise VPNv4 routes enabled with Reoriginate, option
Advertise VPNv6 routes is enabled with Reoriginate, option
Import Stitching is enabled for this neighbor address-family
Import Reoriginate is enabled for this neighbor address-family

Connections established 1; dropped 0
Local host: 30.0.0.1, Local port: 59405, IF Handle: 0x00000000
Foreign host: 20.0.0.1, Foreign port: 179
Last reset 00:00:00

```

At the end of each one AFI VPNv4, VPNv6, or L2VPN EVPN, you can see import and advertise information based on the configuration.

```
Router# show bgp sessions
```

```
Fri Aug 21 00:25:57.216 PDT
```

Neighbor	VRF	Spk	AS	InQ	OutQ	NBRState	NSRState
20.0.0.1	default	0	100	0	0	Established	NSR Ready[PP]
32.0.0.2	default	0	200	0	0	Established	NSR Ready

```
Router# show bgp vpnv4 unicast
```

```

Fri Aug 21 00:28:41.253 PDT
BGP router identifier 30.30.30.30, local AS number 100
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 39
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 39/0
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: 1:1					
*> 10.0.0.1/8	32.0.0.2			0 200 300	i
*> 10.0.0.2/8	32.0.0.2			0 200 300	i
Route Distinguisher: 30.30.30.30:0 (default for vrf foo)					
*> 10.0.0.1/8	32.0.0.2			0 200 300	i
*> 10.0.0.2/8	32.0.0.2			0 200 300	i
*>i100.1.1.1/32	172.16.0.1		100	0	i
*>i100.1.1.2/32	172.16.0.1		100	0	i
*>i200.1.1.1/32	172.16.0.1		100	0	i
*>i200.1.1.2/32	172.16.0.1		100	0	i

```
Router# show bgp vpnv4 unicast rd 192.168.0.1 10.0.0.1/8 detail
```

```

Fri Aug 21 00:28:57.824 PDT
BGP routing table entry for 10.0.0.1/8, Route Distinguisher: 192.168.0.1
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          26        26
  Flags: 0x04103001+0x00000000;
Last Modified: Aug 21 00:24:01.000 for 00:04:58
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):

```

```

20.0.0.1
Path #1: Received by speaker 0
Flags: 0x4000c00005060001, import: 0x80
Advertised to peers (in unique update groups):
  20.0.0.1
200 300
  32.0.0.2 from 32.0.0.2 (40.40.40.40)
    Received Label 24001
    Origin IGP, localpref 100, valid, external, best, group-best, import-candidate,
imported, reoriginated with stitching-rt
    Received Path ID 0, Local Path ID 1, version 26
    Extended community: RT: 1:2
    Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 1:1

```

Router# **show bgp vrf foo**

```

Fri Aug 21 00:24:36.523 PDT
BGP VRF foo, state: Active
BGP Route Distinguisher: 192.168.0.1:0
VRF ID: 0x60000002
BGP router identifier 3192.168.0.1, local AS number 100
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000011 RD version: 35
BGP main routing table version 35
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 31/0

```

```

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: 30.30.30.30:0 (default for vrf foo)					
*> 10.0.0.1/8	172.16.0.1			0 200 300	i
*> 10.0.0.2/8	172.16.0.1			0 200 300	i
*>i100.1.1.1/32	172.16.0.1	100		0	i
*>i100.1.1.2/32	172.16.0.1	100		0	i
*>i200.1.1.1/32	172.16.0.1	100		0	i
*>i200.1.1.2/32	172.16.0.1	100		0	i

Processed 6 prefixes, 6 paths

Router# **show bgp vrf foo ipv4 unicast 100.1.1.1/32 detail**

```

Mon Dec 8 23:24:50.243 PST
BGP routing table entry for 100.1.1.1/32, Route Distinguisher:
192.168.0.1:0
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          43        43
  Local Label: 24001 (with rewrite);
  Flags: 0x05081001+0x00000200;
Last Modified: Dec 8 18:04:21.000 for 05:20:30
Paths: (1 available, best #1)
  Advertised to PE peers (in unique update groups):
    32.0.0.2
    Path #1: Received by speaker 0
    Flags: 0x400061000d060005, import: 0x80
    Advertised to PE peers (in unique update groups):
      32.0.0.2
  Local
    172.16.0.1 (metric 2) from 20.0.0.1 (172.16.0.1)
      Received Label 1234

```

```

Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported, reoriginated
Received Path ID 0, Local Path ID 1, version 43
Extended community: RT:1:2
Originator: 172.16.0.1, Cluster list: 20.20.20.20
Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 100:1

```

Router# **show bgp vpnv4 unicast update-group**

Fri Aug 21 00:27:57.910 PDT

Update group for VPNv4 Unicast, index 0.1:

```

Attributes:
  Outbound policy: pass
  First neighbor AS: 200
  Send communities
  Send GSHUT community if originated
  Send extended communities
  4-byte AS capable
  Send Re-originated VPN routes
  Send multicast attributes
  Minimum advertisement interval: 30 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 8, replicated: 8
All neighbors are assigned to sub-group(s)
  Neighbors in sub-group: 0.2, Filter-Groups num:1
  Neighbors in filter-group: 0.2(RT num: 0)
  32.0.0.2

```

Update group for VPNv4 Unicast, index 0.3:

```

Attributes:
  Neighbor sessions are IPv4
  Internal
  Common admin
  First neighbor AS: 100
  Send communities
  Send GSHUT community if originated
  Send extended communities
  4-byte AS capable
  Send AIGP
  Send Re-originated VPN routes
  Send multicast attributes
  Minimum advertisement interval: 0 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 2, replicated: 2
All neighbors are assigned to sub-group(s)
  Neighbors in sub-group: 0.1, Filter-Groups num:1
  Neighbors in filter-group: 0.1(RT num: 0)
  20.0.0.1

```

Router# **show bgp l2vpn evpn update-group**

Fri Aug 21 00:27:42.786 PDT

Update group for L2VPN EVPN, index 0.2:

```

Attributes:
  Neighbor sessions are IPv4
  Internal
  Common admin

```

```
First neighbor AS: 100
Send communities
Send GSHUT community if originated
Send extended communities
4-byte AS capable
Send AIGP
Send multicast attributes
Minimum advertisement interval: 0 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 4, replicated: 4
All neighbors are assigned to sub-group(s)
  Neighbors in sub-group: 0.1, Filter-Groups num:1
    Neighbors in filter-group: 0.1(RT num: 0)
      20.0.0.1
```

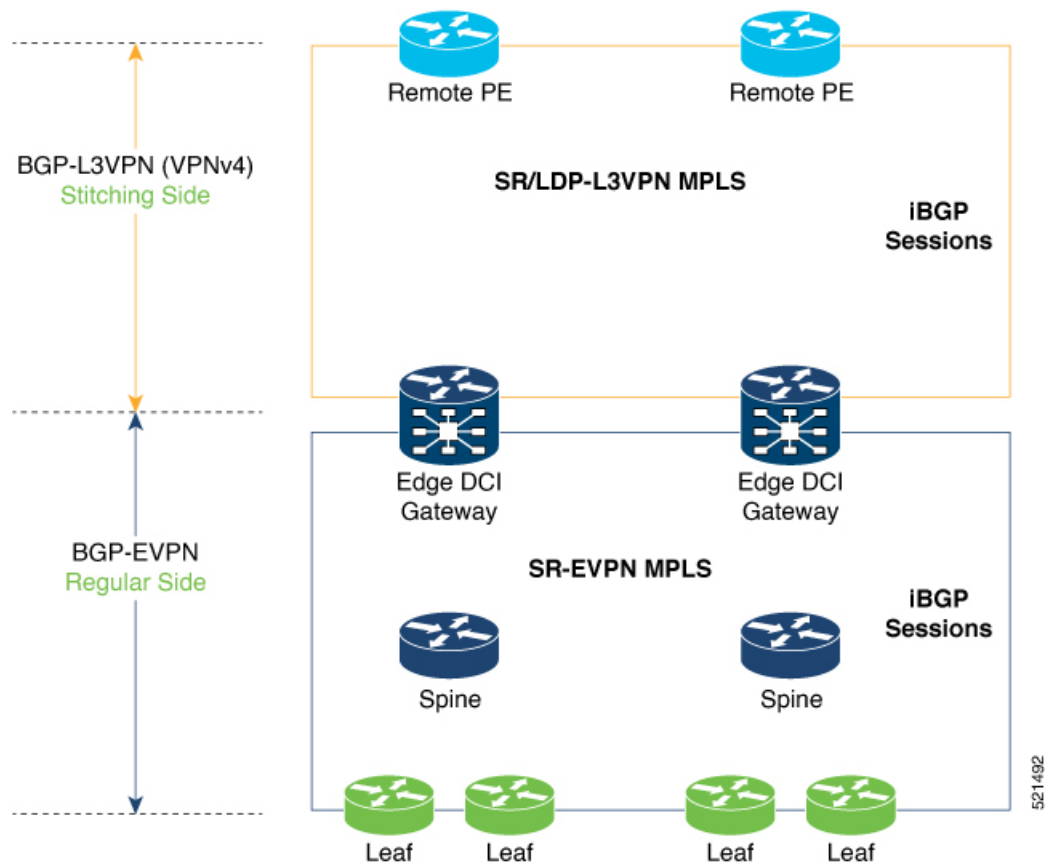
## EVPN-Regular RT and VPNv4-Stitching RT

For each VRF on the DCI gateway, there are two sets of manually configured import and export route-targets for EVPN as regular side and VPNv4 as stitching side. Consider the following sets:

- Data Center Route-Targets for EVPN associated with EVPN BGP neighbor (Regular RT)
- MPLS L3VPN Route-Targets for VPNv4 or VPNv6 associated with L3VPN BGP neighbor (Stitching RT)

This separation of RTs enables the two sets of RTs to be independently configured. The RTs associated with the EVPN BGP neighbor does not require the keyword, it remains a normal configuration. The RTs associated with the L3VPN BGP neighbor require **stitching-rt** keyword under VRF configuration.

The following topology shows regular or normal and stitching side.



### Route Targets

The RTs associated with the L3VPN BGP neighbor are labelled as stitching RTs. The RTs associated with the EVPN BGP neighbor are normal RTs.

### Route Re-Origination

Consider control plane information propagation by the edge DCI gateway from the L3VPN (stitching side) to the Data Center (regular/normal side). Edge DCI gateway advertises to its BGP EVPN neighbor the routes that are re-originated after importing them from the L3VPN BGP neighbor. For this case of VPNv4 or VPNv6 routes being propagated to the BGP EVPN neighbors (Data Center neighbors), re-originating the routes refers to replacing the stitching route-targets with the local route-target values (regular/normal) associated with the BGP EVPN neighbors.

### Local VPNv4 or VPNv6 Route Advertisement

On the edge DCI gateway, the locally sourced VPNv4 or VPNv6 routes (any CE directly connected not using L2VPN with BD/EVI/BVI, using only regular L3 VRF) can be advertised to the BGP EVPN neighbors with the normal route targets (RTs) configured for the VRF or the stitching RTs associated with the BGP EVPN neighbors. By default, these routes are advertised with the normal route targets to the BGP EVPN Neighbors (regular/normal side)

VPNv4 neighbors require an additional configuration on the existing legacy VRF to allow these routes to be advertised to VPNv4 RR or remote PEs. Configure **stitching-rt** keyword on existing VRF under import/export RT.

### Route Distinguishers

The Router Distinguisher (RD) associated per VRF must be unique per PE in the network. There are few available options to keep unique RD per device:

- Manual configuration: You must manually assign a unique value per device in the network. For example, in this scenario:
  - Leaf (ToR) = RD 1
  - Edge DCI Gateway = RD 2
  - Remote PE = RD 3
- Use **rd auto** command under VRF. 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.




---

**Note** In a DCI deployment, for route re-originate with stitching-rt for a particular VRF, using the same Route Distinguisher (RD) between edge DCI gateway and MPLS-VPN PE or same RD between edge DCI gateway and Leaf (ToR) is not supported.

---

### Configure EVPN-Regular RT and VPNv4-Stitching RT

This section describes tasks to configure EVPN-Regular RT and VPNv4-Stitching RT. Perform the following tasks to complete the configuration:

- Configure Leaf (ToR)
- Configure Spine-RR (Route Reflector)
- Configure Edge DCI Gateway
- Configure EVPN BGP neighbor and route advertisements
- Configure L3VPN BGP neighbor relationship and route advertisements

#### Configure Leaf (ToR)

Configure VRF in Leaf (ToR) at BGP-EVPN (regular/normal side). Note that the **stitching-rt** keyword is not required.

```
vrf data-center1
  address-family ipv4 unicast
    import route-target
      1:2                               // BGP - EVPN (Regular/Normal Side)
  !
```

```

export route-target
  1:2 // BGP - EVPN (Regular/Normal Side)
!
router bgp 100
  neighbor 10.10.1.1 // Spine Loopback IP Address
    address-family l2vpn evpn
      advertise vpnv4 unicast
      advertise vpnv6 unicast
!

```



**Note** Advertise vpnv4/vpnv6 unicast enables local learned regular L3 VRF prefixes to be advertised as EVPN prefixes to BGP-EVPN neighbor. This means any local prefixes such as PE-CE without L2VPN with BD/EVI/BVI configuration. If all the services are pure EVPN with L2VPN with BD/EVI/BVI configuration these commands are not required.

### Configure Spine-RR

Configure Spine RR with Leaf (ToR) and edge DCI gateway as RR client for AFI L2VPN EVPN.

```

// VRF Config is not required //
router bgp 100
  neighbor 10.10.2.1 // Leaf (ToR) Loopback IP Address
    address-family l2vpn evpn
      route-reflector-client
  !
  neighbor 10.10.3.1 // Edge DCI Gateway Loopback IP Address
    address-family l2vpn evpn
      route-reflector-client
!

```

### Configure Edge DCI Gateway

You can configure DCI with the same VRF as Leaf (ToR). Use the same RT as remote PE for L3VPN network or the same VRF if that is possible.

### Configure VRF and Route Targets Import and Export rules

Perform the following steps to configure VRF and define route targets to be used for import and export of forwarding information.

```

vrf data-center1
  address-family ipv4 unicast
    import route-target
      1:1 stitching // BGP - L3VPN (Stitching Side)
      1:2 // BGP - EVPN (Regular/normal Side)
  !
  export route-target
    1:1 stitching // BGP - L3VPN (Stitching Side)
    1:2 // BGP - EVPN (Regular/normal Side)
!

```

### Configure EVPN BGP Neighbor and Route Advertisements

Perform this task on the edge DCI gateway to configure BGP neighbor relationship and route advertisements with the EVPN BGP neighbor.



```

router bgp 100
 address-family l2vpn evpn
 !
 neighbor 10.10.1.1          // Spine Loopback IP Address
  address-family l2vpn evpn
   import re-originate stitching-rt //Imp EVPN RT 1:2, re-originate VPNv4 1:1
   advertise vpnv4 unicast re-originated //Send routes VPNv4 RT 1:1
 !

```

### Configure L3VPN BGP Neighbor Relationship and Route Advertisements

Perform the following steps to configure BGP neighbor relationship and route advertisements with the L3VPN BGP neighbor.

```

router bgp 100
 address-family vpnv4 unicast
 !
 neighbor 10.10.1.1          // Spine Loopback IP Address
  address-family vpnv4 unicast // Same config for VPNv6
   import stitching-rt re-originate // Imp VPNv4 1:1, reoriginate EVPN 1:2
   advertise vpnv4 unicast re-originated stitching-rt //Send Routes EVPN 1:2
   advertise vpnv6 unicast re-originated stitching-rt //Send Routes EVPN 1:2
 !

```

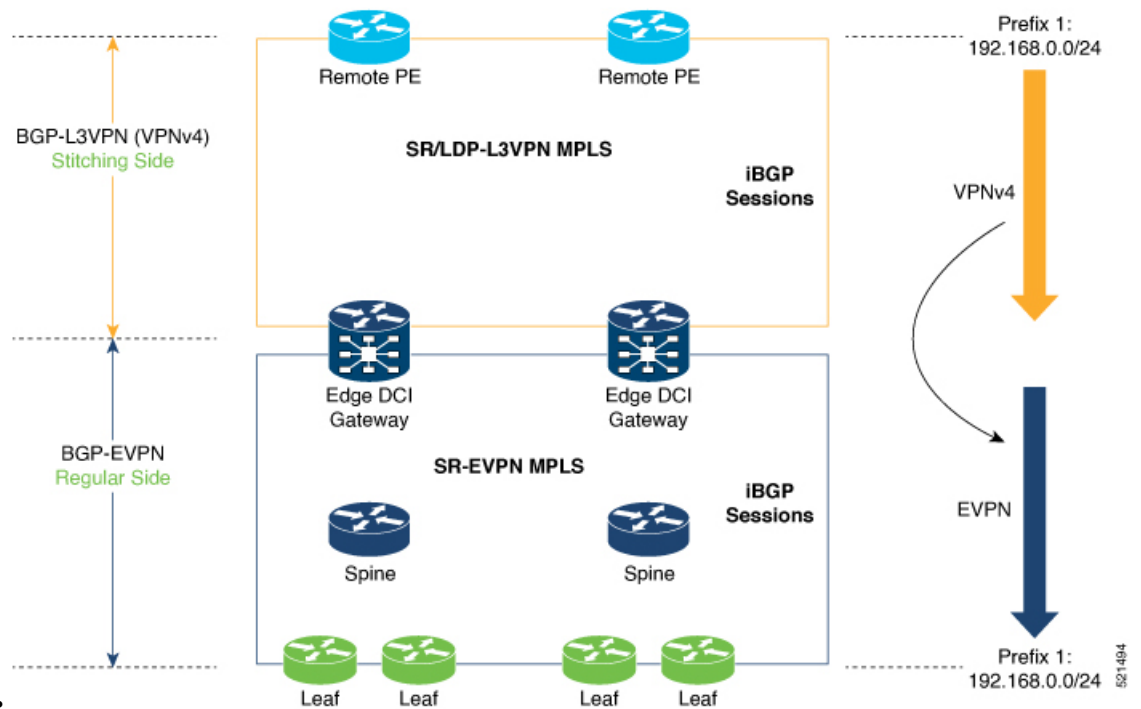


**Note** The **stitching-rt** applies for L3VPN RT and EVPN RT does not require the **stitching-rt** for this use case.

If there are existing regular local L3 VRF without L2VPN with BD/EVI/BVI in these devices, configure import/export Stitching-RT for existing VRFs to advertise to L3VPN RR or remote PEs.

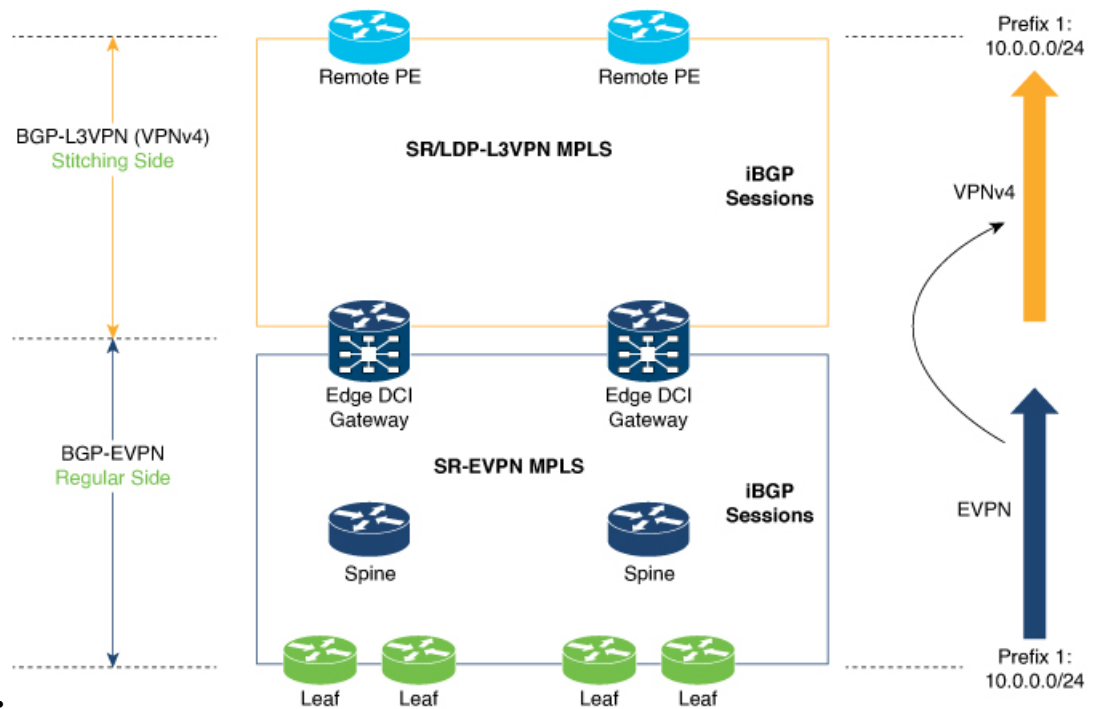
Configuration applies in two directions:

- Stitching from VPNv4 to EVPN routes. Prefixes received from MPLS L3VPN network and re-originated as EVPN prefixes towards Data Center Spine RR and Leaf (ToR)
  1. Importing VPNv4 routes with **import stitching-rt re-originate** command under AFI VPNv4 UNICAST. This command imports routes using RT 1:1 stitching-rt and then re-originate with BGP EVPN 1:2
  2. Advertising re-originated EVPN routes with VPNv4 RT with **advertise vpnv4 unicast re-originated** command under AFI L2VPN EVPN. This command advertises routes from MPLS L3VPN network (VPNv4) to BGP EVPN neighbors inside Data Center (Spine RR and then Leaf (ToR)), re-originating these routes using BGP EVPN 1:2.



- Stitching from EVPN to VPNv4 routes. Prefixes received from BGP-EVPN Data Center and re-originated as MPLS L3VPN prefixes towards VPNv4 RR or remote PE in L3VPN network.

1. Importing EVPN routes with **import re-originate stitching-rt** command under AFI L2VPN EVPN. This command imports routes using RT 1:2 and then re-originate with VPNv4 RT 1:1 **stitching-rt**.
2. Advertising re-originated EVPN routes with VPNv4 RT with **advertise vpnv4 unicast re-originated stitching-rt** command under AFI VPNv4 UNICAST. This command advertises routes from EVPN Data Center to VPNv4 RR or remote PEs, re-originating these routes using VPNv4 RT 1:1 **stitching-rt**



### Verification of Edge DCI Gateway Configuration

Router# **show bgp l2vpn evpn**

```

Fri Aug 21 00:24:10.773 PDT
BGP router identifier 30.30.30.30, local AS number 100
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 16
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 16/0
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: 100:1					
*>i [2] [10000] [48] [0226.51bd.c81c] [32] [200::1001]/232	11.0.0.1	100	0	i	
*>i [2] [10000] [48] [0226.51bd.c81c] [32] [200:1::1001]/232	11.0.0.1	100	0	i	
*>i [2] [10000] [48] [0226.51bd.c81c] [32] [200.1.1.1]/136	11.0.0.1	100	0	i	
*>i [2] [10000] [48] [0226.51bd.c81c] [32] [200.1.1.2]/136	11.0.0.1	100	0	i	
*>i [5] [4231] [32] [100.1.1.1]/80	11.0.0.1	100	0	i	
*>i [5] [4231] [32] [100.1.1.2]/80	11.0.0.1	100	0	i	
*>i [5] [4231] [112] [fec0::1001]/176	11.0.0.1	100	0	i	
*>i [5] [4232] [112] [fec0::1:1001]/176	11.0.0.1	100	0	i	

```

11.0.0.1          100      0 i

Processed 8 prefixes, 8 paths

Router# show bgp l2vpn evpn rd 100:1 [5][4231][112][fec0::1001]/176 detail

Fri Aug 21 00:34:43.747 PDT
BGP routing table entry for [5][4231][112][fec0::1001]/176, Route Distinguisher: 100:1
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          5         5
  Flags: 0x04040001+0x00000000;
Last Modified: Aug 21 00:16:58.000 for 00:17:46
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Flags: 0x4000600025060005, import: 0x3f
  Not advertised to any peer
Local
  11.0.0.1 (metric 2) from 20.0.0.1 (11.0.0.1)
  Received Label 16001
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
reoriginate stitching-rt, not-in-vrf
  Received Path ID 0, Local Path ID 1, version 5
  Extended community: Flags 0x6: RT:1:1
  Originator: 11.0.0.1, Cluster list: 20.20.20.20
  EVPN ESI: ffff.ffff.ffff.ffff.ff01, Gateway Address : fec0::254

```

The main difference with scenario 1 is that the prefixes have a **reoriginate stitching-rt** keyword on the output versus scenario 1 having just reoriginate.

```

Router# show bgp l2vpn evpn neighbors 20.0.0.1 detail

Fri Aug 21 00:25:37.383 PDT

BGP neighbor is 20.0.0.1
Remote AS 100, local AS 100, internal link
Remote router ID 20.20.20.20
BGP state = Established, up for 00:08:58
NSR State: NSR Ready
Last read 00:00:34, Last read before reset 00:00:00
Hold time is 180, keepalive interval is 60 seconds
Configured hold time: 180, keepalive: 60, min acceptable hold time: 3
Last write 00:00:36, attempted 19, written 19
Second last write 00:01:36, attempted 143, written 143
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 Aug 21 00:25:03.667 last full not set pulse count 33
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
Entered Neighbor NSR TCP mode:
  TCP Initial Sync :           Aug 21 00:18:07.291
  TCP Initial Sync Phase Two :  Aug 21 00:18:07.319
  TCP Initial Sync Done :       Aug 21 00:18:08.334
Multi-protocol capability received
Neighbor capabilities:
  Route refresh:                Yes      Yes
  4-byte AS:                    Yes      Yes
  Address family VPNv4 Unicast: Yes      No

```

```

Address family VPNv6 Unicast:  Yes          No
Address family L2VPN EVPN:    Yes          Yes
Message stats:
InQ depth: 0, OutQ depth: 0
      Last_Sent          Sent  Last_Rcvd          Rcvd
Open:      Aug 21 00:16:38.087      1  Aug 21 00:16:40.123      1
Notification:  ---                0  ---                    0
Update:      Aug 21 00:24:01.421      9  Aug 21 00:24:03.652     13
Keepalive:   Aug 21 00:25:01.434      8  Aug 21 00:25:03.667     9
Route_Refresh: Aug 21 00:24:01.377      3  ---                    0
Total:                               21                    23
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: VPNv4 Unicast
BGP neighbor version 35
Update group: 0.3 Filter-group: 0.1 No Refresh request being processed
Advertise Reorigination Enabled
Advertise AFI EoR can be sent
Route refresh request: received 0, sent 0
0 accepted prefixes, 0 are bestpaths
Cumulative no. of prefixes denied: 0.
Prefix advertised 4, suppressed 0, withdrawn 0
Maximum prefixes allowed 2097152
Threshold for warning message 75%, restart interval 0 min
AIGP is enabled
An EoR was not received during read-only mode
Last ack version 35, Last synced ack version 35
Outstanding version objects: current 0, max 1
Additional-paths operation: None
Send Multicast Attributes

For Address Family: VPNv6 Unicast
BGP neighbor version 29
Update group: 0.3 Filter-group: 0.1 No Refresh request being processed
Advertise Reorigination Enabled
Advertise AFI EoR can be sent
Route refresh request: received 0, sent 0
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
AIGP is enabled
An EoR was not received during read-only mode
Last ack version 29, Last synced ack version 29
Outstanding version objects: current 0, max 0
Additional-paths operation: None
Send Multicast Attributes
Advertise VPNv4 routes enabled with Reoriginate,Local with stitching-RT option

```

```

For Address Family: L2VPN EVPN
BGP neighbor version 18
Update group: 0.2 Filter-group: 0.1 No Refresh request being processed
Route refresh request: received 0, sent 3
8 accepted prefixes, 8 are bestpaths
Cumulative no. of prefixes denied: 0.
Prefix advertised 4, suppressed 0, withdrawn 6
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 18, Last synced ack version 18

```

```

Outstanding version objects: current 0, max 2
Additional-paths operation: None
Send Multicast Attributes
Advertise VPNv4 routes enabled with Reoriginate, option
Advertise VPNv6 routes is enabled with Reoriginate, option
Import Reoriginate is enabled for this neighbor address-family

Connections established 1; dropped 0
Local host: 30.0.0.1, Local port: 59405, IF Handle: 0x00000000
Foreign host: 20.0.0.1, Foreign port: 179
Last reset 00:00:00

```

At the end of each one AFI VPNv4, VPNv6, or L2VPN EVPN, you can see import and advertise information based on the configuration.

Based on whether stitching-side or regular side, import stitching applies on VPNv4 AFI. In Scenario 1 you can see import stitching under L2VPN EVPN.

```
Router# show bgp sessions
```

```
Fri Aug 21 00:25:57.216 PDT
```

Neighbor	VRF	Spk	AS	InQ	OutQ	NBRState	NSRState
20.0.0.1	default	0	100	0	0	Established	NSR Ready[PP]
32.0.0.2	default	0	200	0	0	Established	NSR Ready

```
Router# show bgp vpnv4 unicast
```

```

Fri Aug 21 00:28:41.253 PDT
BGP router identifier 30.30.30.30, local AS number 100
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 39
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 39/0
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: 1:1					
*> 1.1.1.0/24	32.0.0.2		0	200	300 i
*> 1.1.2.0/24	32.0.0.2		0	200	300 i
Route Distinguisher: 30.30.30.30:0 (default for vrf foo)					
*> 1.1.1.0/24	32.0.0.2		0	200	300 i
*> 1.1.2.0/24	32.0.0.2		0	200	300 i
*>i100.1.1.1/32	11.0.0.1		100	0	i
*>i100.1.1.2/32	11.0.0.1		100	0	i
*>i200.1.1.1/32	11.0.0.1		100	0	i
*>i200.1.1.2/32	11.0.0.1		100	0	i

In origin IGP line, you can see that the prefix was reoriginated with regular-RT.

```
Router# show bgp vpnv4 unicast rd 30.30.30.30:0 1.1.1.0/24 detail
```

```

Fri Aug 21 00:28:57.824 PDT
BGP routing table entry for 1.1.1.0/24, Route Distinguisher: 30.30.30.30:0
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          26        26
  Flags: 0x04103001+0x00000000;
Last Modified: Aug 21 00:24:01.000 for 00:04:58

```

```

Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    20.0.0.1
  Path #1: Received by speaker 0
  Flags: 0x4000c00005060001, import: 0x80
  Advertised to peers (in unique update groups):
    20.0.0.1
  200 300
    32.0.0.2 from 32.0.0.2 (40.40.40.40)
      Received Label 24001
      Origin IGP, localpref 100, valid, external, best, group-best, import-candidate,
imported, reoriginated
      Received Path ID 0, Local Path ID 1, version 26
      Extended community: RT: 1:2
      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 1:1

```

Router# **show bgp vrf foo**

```

Fri Aug 21 00:24:36.523 PDT
BGP VRF foo, state: Active
BGP Route Distinguisher: 30.30.30.30:0
VRF ID: 0x60000002
BGP router identifier 30.30.30.30, local AS number 100
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000011 RD version: 35
BGP main routing table version 35
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 31/0

```

```

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: 30.30.30.30:0 (default for vrf foo)					
*> 1.1.1.0/24	32.0.0.2			0 200 300	i
*> 1.1.2.0/24	32.0.0.2			0 200 300	i
*>i100.1.1.1/32	11.0.0.1		100	0	i
*>i100.1.1.2/32	11.0.0.1		100	0	i
*>i200.1.1.1/32	11.0.0.1		100	0	i
*>i200.1.1.2/32	11.0.0.1		100	0	i

Processed 6 prefixes, 6 paths

Router# **show bgp vrf foo ipv4 unicast 100.1.1.1/32 detail**

```

Mon Dec 8 23:24:50.243 PST
BGP routing table entry for 100.1.1.1/32, Route Distinguisher:
30.30.30.30:0

```

Versions:

```

Process          bRIB/RIB  SendTblVer
Speaker          43        43

```

Local Label: 24001 (with rewrite);

Flags: 0x05081001+0x00000200;

Last Modified: Dec 8 18:04:21.000 for 05:20:30

Paths: (1 available, best #1)

Advertised to PE peers (in unique update groups):

32.0.0.2

Path #1: Received by speaker 0

Flags: 0x400061000d060005, import: 0x80

Advertised to PE peers (in unique update groups):

32.0.0.2

Local

11.0.0.1 (metric 2) from 20.0.0.1 (11.0.0.1)

Received Label 1234

```

Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported, reoriginated with stitching-rt
Received Path ID 0, Local Path ID 1, version 43
Extended community: RT:1:2
Originator: 11.0.0.1, Cluster list: 20.20.20.20
Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 100:1v

```

```
Router# show bgp vpnv4 unicast update-group
```

```
Fri Aug 21 00:27:57.910 PDT
```

```
Update group for VPNv4 Unicast, index 0.1:
```

```

Attributes:
  Outbound policy: pass
  First neighbor AS: 200
  Send communities
  Send GSHUT community if originated
  Send extended communities
  4-byte AS capable
  Send Re-originated VPN routes
  Send multicast attributes
  Minimum advertisement interval: 30 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 8, replicated: 8
All neighbors are assigned to sub-group(s)
  Neighbors in sub-group: 0.2, Filter-Groups num:1
  Neighbors in filter-group: 0.2(RT num: 0)
  32.0.0.2

```

```
Update group for VPNv4 Unicast, index 0.3:
```

```

Attributes:
  Neighbor sessions are IPv4
  Internal
  Common admin
  First neighbor AS: 100
  Send communities
  Send GSHUT community if originated
  Send extended communities
  4-byte AS capable
  Send AIGP
  Send Re-originated VPN routes
  Send multicast attributes
  Minimum advertisement interval: 0 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 2, replicated: 2
All neighbors are assigned to sub-group(s)
  Neighbors in sub-group: 0.1, Filter-Groups num:1
  Neighbors in filter-group: 0.1(RT num: 0)
  20.0.0.1

```

```
Router# show bgp l2vpn evpn update-group
```

```
Fri Aug 21 00:27:42.786 PDT
```

```
Update group for L2VPN EVPN, index 0.2:
```

```

Attributes:
  Neighbor sessions are IPv4
  Internal
  Common admin
  First neighbor AS: 100
  Send communities

```



```
Send GSHUT community if originated
Send extended communities
4-byte AS capable
Send AIGP
Send multicast attributes
Minimum advertisement interval: 0 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 4, replicated: 4
All neighbors are assigned to sub-group(s)
  Neighbors in sub-group: 0.1, Filter-Groups num:1
    Neighbors in filter-group: 0.1(RT num: 0)
      20.0.0.1
```

