



MPLS: Layer 3 VPNs Configuration Guide, Cisco IOS XE 16 (NCS 4200 Series)

First Published: 2017-06-09

Last Modified: 2019-06-10

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.

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)

© 2017–2020 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Feature History 1

CHAPTER 2

MPLS Virtual Private Networks 3

- Prerequisites for MPLS Virtual Private Networks 3
- Restrictions for MPLS Virtual Private Networks 3
- Information About MPLS Virtual Private Networks 5
 - MPLS Virtual Private Network Definition 5
 - How an MPLS Virtual Private Network Works 6
 - How Virtual Routing and Forwarding Tables Work in an MPLS Virtual Private Network 7
 - How VPN Routing Information Is Distributed in an MPLS Virtual Private Network 7
 - MPLS Forwarding 7
- Major Components of an MPLS Virtual Private Network 8
- Benefits of an MPLS Virtual Private Network 8
- How to Configure MPLS Virtual Private Networks 10
 - Configuring the Core Network 10
 - Assessing the Needs of MPLS Virtual Private Network Customers 10
 - Configuring MPLS in the Core 11
 - Connecting the MPLS Virtual Private Network Customers 11
 - Defining VRFs on the PE Devices to Enable Customer Connectivity 11
 - Configuring VRF Interfaces on PE Devices for Each VPN Customer 13
 - Configuring Routing Protocols Between the PE and CE Devices 14
- Verifying the Virtual Private Network Configuration 17
- Verifying Connectivity Between MPLS Virtual Private Network Sites 17
 - Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core 18
 - Verifying That the Local and Remote CE Devices Are in the PE Routing Table 18
- Configuration Examples for MPLS Virtual Private Networks 20

Example: Configuring an MPLS Virtual Private Network Using RIP	20
Example: Configuring an MPLS Virtual Private Network Using Static Routes	22
Additional References	23

CHAPTER 3**Multiprotocol BGP MPLS VPN 25**

Prerequisites for Multiprotocol BGP MPLS VPN	25
Information About Multiprotocol BGP MPLS VPN	25
MPLS Virtual Private Network Definition	25
How an MPLS Virtual Private Network Works	26
How Virtual Routing and Forwarding Tables Work in an MPLS Virtual Private Network	27
How VPN Routing Information Is Distributed in an MPLS Virtual Private Network	27
BGP Distribution of VPN Routing Information	27
Major Components of an MPLS Virtual Private Network	28
How to Configure Multiprotocol BGP MPLS VPN	28
Configuring Multiprotocol BGP Connectivity on the PE Devices and Route Reflectors	28
Troubleshooting Tips	30
Configuring BGP as the Routing Protocol Between the PE and CE Devices	31
Verifying the Virtual Private Network Configuration	32
Verifying Connectivity Between MPLS Virtual Private Network Sites	33
Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core	33
Verifying That the Local and Remote CE Devices Are in the PE Routing Table	33
Configuration Examples for Multiprotocol BGP MPLS VPN	35
Example: Configuring an MPLS Virtual Private Network Using BGP	35
Additional References	36

CHAPTER 4**IPv6 VPN over MPLS 37**

Prerequisites for IPv6 VPN over MPLS	37
Restrictions for IPv6 VPN over MPLS	38
Information About IPv6 VPN over MPLS	38
IPv6 VPN over MPLS Overview	38
Addressing Considerations for IPv6 VPN over MPLS	38
Basic IPv6 VPN over MPLS Functionality	39
IPv6 VPN Architecture Overview	39
IPv6 VPN Next Hop	40

MPLS Forwarding	40
VRF Concepts	40
IPv6 VPN Scalability	41
Advanced IPv6 MPLS VPN Functionality	42
Internet Access	42
Multiautonomous-System Backbones	43
Carrier Supporting Carriers	44
How to Configure IPv6 VPN over MPLS	44
Configuring a Virtual Routing and Forwarding Instance for IPv6	44
Binding a VRF to an Interface	46
Configuring a Static Route for PE-to-CE Routing	47
Configuring eBGP PE-to-CE Routing Sessions	48
Configuring the IPv6 VPN Address Family for iBGP	49
Configuring Route Reflectors for Improved Scalability	51
Configuring Internet Access	57
Configuring the Internet Gateway	57
Configuring the IPv6 VPN PE	61
Configuring a Multiautonomous-System Backbone for IPv6 VPN	64
Configuring the PE VPN for a Multiautonomous-System Backbone	66
Configuring the Route Reflector for a Multiautonomous-System Backbone	69
Configuring the ASBR	77
Configuring CSC for IPv6 VPN	80
Configuration Examples for IPv6 VPN over MPLS	81
Examples: IPv6 VPN over MPLS Routing	81
Example: BGP IPv6 Activity Summary	81
Example: Dumping the BGP IPv6 Tables	82
Example: Dumping the IPv6 Routing Tables	82
Examples: IPv6 VPN over MPLS Forwarding	83
Example: PE-CE Connectivity	83
Examples: PE Imposition Path	84
Examples: PE Disposition Path	85
Examples: Label Switch Path	85
Examples: IPv6 VPN over MPLS VRF	86
Examples: VRF Information	86

Example: IPv6 VPN Configuration Using IPv4 Next Hop	87
Additional References	87
Feature Information for IPv6 VPN over MPLS	88
Glossary	88

CHAPTER 5**IPv6 Switching: Provider Edge Router over MPLS 91**

Prerequisites for IPv6 Switching: Provider Edge Router over MPLS	91
Information About IPv6 Switching: Provider Edge Router over MPLS	91
Benefits of Deploying IPv6 over MPLS Backbones	91
IPv6 on the Provider Edge Devices	92
How to Deploy IPv6 Switching: Provider Edge Router over MPLS	93
Deploying IPv6 on the Provider Edge Devices (6PE)	93
Specifying the Source Address Interface on a 6PE Device	93
Binding and Advertising the 6PE Label to Advertise Prefixes	94
Configuring IBGP Multipath Load Sharing	96
Configuration Examples for IPv6 Switching: Provider Edge Router over MPLS	97
Example: Provider Edge Device	97
Example: Core Device	97
Example: Monitoring 6PE	98
Additional References for IPv6 Switching: Provider Edge Router over MPLS	100

CHAPTER 6**Multi-VRF Support 101**

Prerequisites for Multi-VRF Support	101
Restrictions for Multi-VRF Support	101
Information About Multi-VRF Support	102
How the Multi-VRF Support Feature Works	102
How Packets Are Forwarded in a Network Using the Multi-VRF Support Feature	103
Considerations When Configuring the Multi-VRF Support Feature	104
How to Configure Multi-VRF Support	104
Configuring VRFs	104
Configuring BGP as the Routing Protocol	106
Configuring PE-to-CE MPLS Forwarding and Signaling with BGP	108
Configuring a Routing Protocol Other than BGP	110
Configuring PE-to-CE MPLS Forwarding and Signaling with LDP	111

Configuration Examples for Multi-VRF Support	112
Example: Configuring Multi-VRF Support on the PE Device	112
Example: Configuring Multi-VRF Support on the CE Device	114
Additional References	117

CHAPTER 7**ECMP Load Balancing 119**

Restrictions for ECMP Load Balancing	120
Configuring ECMP Load Balancing	120
Configuration Examples for ECMP Load Balancing	121
Example: Configuring ECMP Load balancing	121
Verifying ECMP Load Balancing	122



CHAPTER 1

Feature History

The following table lists the new and modified features supported in the MPLS: Layer 3 VPNs Configuration Guide in Cisco IOS XE 16 releases, on Cisco NCS 4201 and Cisco NCS 4202 routers.

Feature Name	Cisco IOS XE Release
IPv4 Unicast Generic Routing Encapsulation Tunnel	16.11.1a

The following table lists the new features supported in the MPLS: Layer 3 VPNs Configuration Guide in Cisco IOS XE 16 releases, on Cisco NCS 4206 and Cisco NCS 4216 routers.

Feature Name	Cisco IOS XE Release
Generic Routing Encapsulation (GRE) Feature Updates	16.12.1



CHAPTER 2

MPLS Virtual Private Networks

An MPLS Virtual Private Network (VPN) consists of a set of sites that are interconnected by means of a Multiprotocol Label Switching (MPLS) provider core network. At each customer site, one or more customer edge (CE) devices attach to one or more provider edge (PE) devices. This module explains how to create an MPLS VPN.

- [Prerequisites for MPLS Virtual Private Networks, on page 3](#)
- [Restrictions for MPLS Virtual Private Networks, on page 3](#)
- [Information About MPLS Virtual Private Networks, on page 5](#)
- [How to Configure MPLS Virtual Private Networks, on page 10](#)
- [Configuration Examples for MPLS Virtual Private Networks, on page 20](#)
- [Additional References, on page 23](#)

Prerequisites for MPLS Virtual Private Networks

- Make sure that you have installed Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP), and Cisco Express Forwarding in your network.
- All devices in the core, including the provider edge (PE) devices, must be able to support Cisco Express Forwarding and MPLS forwarding. See the “Assessing the Needs of the MPLS Virtual Private Network Customers” section.
- Cisco Express Forwarding must be enabled on all devices in the core, including the PE devices. For information about how to determine if Cisco Express Forwarding is enabled, see the “Configuring Basic Cisco Express Forwarding” module in the *Cisco Express Forwarding Configuration Guide*.

Restrictions for MPLS Virtual Private Networks

When static routes are configured in a Multiprotocol Label Switching (MPLS) or MPLS virtual private network (VPN) environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in software releases that support the Tag Forwarding Information Base (TFIB). The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in releases that support the MPLS Forwarding Infrastructure (MFI). For details about the supported releases, see the *Multiprotocol Label Switching Command Reference*. Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

The following **ip route** command is supported when you configure static routes in an MPLS environment:

- **ip route** *destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS environment and configure load sharing with static nonrecursive routes and a specific outbound interface:

- **ip route** *destination-prefix mask interface1 next-hop1*
- **ip route** *destination-prefix mask interface2 next-hop2*

Unsupported Static Routes in an MPLS Environment That Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS environment:

- **ip route** *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the next hop can be reached through two paths:

- **ip route** *destination-prefix mask next-hop-address*

The following **ip route** commands are not supported when you configure static routes in an MPLS environment and enable load sharing where the destination can be reached through two next hops:

- **ip route** *destination-prefix mask next-hop1*
- **ip route** *destination-prefix mask next-hop2*

Use the *interface* and *next-hop* arguments when specifying static routes.

Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface1 next-hop1*
- **ip route vrf** *vrf-name destination-prefix mask interface2 next-hop2*

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table in the MPLS cloud in the global routing table. For example, these commands are supported when the next hop is pointing to the Internet gateway.

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address global*
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* (This command is supported when the next hop and interface are in the core.)

The following **ip route** commands are supported when you configure static routes in an MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interface:

- **ip route** *destination-prefix mask interface1 next-hop1*
- **ip route** *destination-prefix mask interface2 next-hop2*

Unsupported Static Routes in an MPLS VPN Environment That Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the next hop can be reached through two paths:

- **ip route vrf** *destination-prefix mask next-hop-address global*

The following **ip route** commands are not supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the destination can be reached through two next hops:

- **ip route vrf** *destination-prefix mask next-hop1 global*
- **ip route vrf** *destination-prefix mask next-hop2 global*

The following **ip route vrf** commands are not supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

- **ip route vrf** *vrf-name destination-prefix mask next-hop1 vrf-name destination-prefix mask next-hop1*
- **ip route vrf** *vrf-name destination-prefix mask next-hop2*

Supported Static Routes in an MPLS VPN Environment Where the Next Hop Resides in the Global Table on the CE Device

The following **ip route vrf** command is supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table on the customer edge (CE) side. For example, the following command is supported when the destination prefix is the CE device's loopback address, as in external Border Gateway Protocol (EBGP) multihop cases.

- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static nonrecursive routes and a specific outbound interface:

- **ip route** *destination-prefix mask interface1 nexthop1*
- **ip route** *destination-prefix mask interface2 nexthop2*

Information About MPLS Virtual Private Networks

MPLS Virtual Private Network Definition

Before defining a Multiprotocol Label Switching virtual private network (MPLS VPN), you must define a VPN in general. 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, because 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 the customer's involvement.

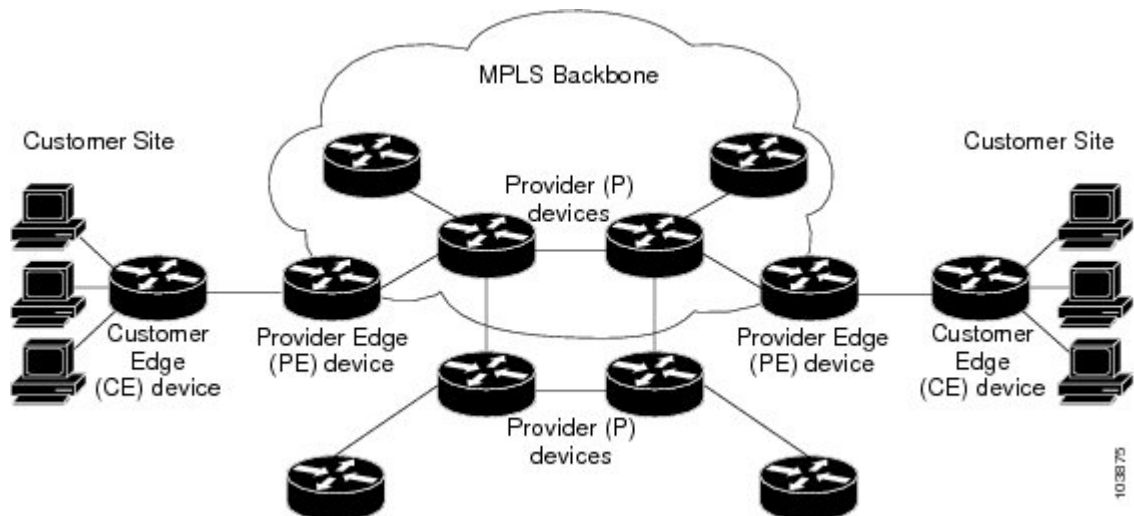
MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the service provider's edge device that provides services to the customer site needs to be updated.

The different parts of the MPLS VPN are described as follows:

- Provider (P) device—Device in the core of the provider network. P devices run MPLS switching, and do not attach VPN labels to routed packets. The MPLS label in each route is assigned by the provider edge (PE) device. VPN labels are used to direct data packets to the correct egress device.
- PE device—Device that attaches the VPN label to incoming packets based on the interface or subinterface on which they are received. A PE device attaches directly to a customer edge (CE) device.
- Customer (C) device—Device in the ISP or enterprise network.
- CE device—Edge device on the network of the ISP that connects to the PE device on the network. A CE device must interface with a PE device.

The figure below shows a basic MPLS VPN.

Figure 1: Basic MPLS VPN Terminology



How an MPLS Virtual Private Network Works

Multiprotocol Label Switching virtual private network (MPLS VPN) functionality is enabled at the edge of an MPLS network. The provider edge (PE) device performs the following:

- Exchanges routing updates with the customer edge (CE) device.
- Translates the CE routing information into VPNv4 routes.
- Exchanges VPNv4 routes with other PE devices through the Multiprotocol Border Gateway Protocol (MP-BGP).

The following sections describe how MPLS VPN works:

How Virtual Routing and Forwarding Tables Work in an MPLS Virtual Private Network

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

- An IP routing table
- A derived Cisco Express Forwarding 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

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 site's 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 Cisco Express Forwarding table for each VRF. A separate set of routing and Cisco Express Forwarding tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN, and they also prevent packets that are outside a VPN from being forwarded to a device within the VPN.

How VPN Routing Information Is Distributed in an MPLS Virtual Private Network

The distribution of virtual private network (VPN) routing information is controlled through the use of VPN route target communities, implemented by Border Gateway Protocol (BGP) extended communities. VPN routing information is distributed as follows:

- When a VPN route that is learned from a customer edge (CE) device is injected into 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 virtual routing and forwarding (VRF) instance 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 in order 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.

MPLS Forwarding

Based on routing information stored in the virtual routing and forwarding (VRF) IP routing table and VRF Cisco Express Forwarding table, packets are forwarded to their destination using Multiprotocol Label Switching (MPLS).

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

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

Major Components of an MPLS Virtual Private Network

An Multiprotocol Label Switching (MPLS)-based virtual private network (VPN) 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 VPN community provider edge (PE) devices—MP-BGP propagates virtual routing and forwarding (VRF) reachability information to all members of a VPN community. MP-BGP peering must be configured on all PE devices 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.

Benefits of an MPLS Virtual Private Network

Multiprotocol Label Switching virtual private networks (MPLS VPNs) allow service providers to deploy scalable VPNs and build the foundation to deliver value-added services, such as the following:

Connectionless Service

A significant technical advantage of MPLS VPNs is that they are connectionless. The Internet owes its success to its basic technology, TCP/IP. TCP/IP is built on a packet-based, connectionless network paradigm. This means that no prior action is necessary to establish communication between hosts, making it easy for two parties to communicate. To establish privacy in a connectionless IP environment, current VPN solutions impose a connection-oriented, point-to-point overlay on the network. Even if it runs over a connectionless network, a VPN cannot take advantage of the ease of connectivity and multiple services available in connectionless networks. When you create a connectionless VPN, you do not need tunnels and encryption for network privacy, thus eliminating significant complexity.

Centralized Service

Building VPNs in Layer 3 allows delivery of targeted services to a group of users represented by a VPN. A VPN must give service providers more than a mechanism for privately connecting users to intranet services. It must also provide a way to flexibly deliver value-added services to targeted customers. Scalability is critical,

because customers want to use services privately in their intranets and extranets. Because MPLS VPNs are seen as private intranets, you may use new IP services such as:

- Multicast
- Quality of service (QoS)
- Telephony support within a VPN
- Centralized services including content and web hosting to a VPN

You can customize several combinations of specialized services for individual customers. For example, a service that combines IP multicast with a low-latency service class enables video conferencing within an intranet.

Scalability

If you create a VPN using connection-oriented, point-to-point overlays, Frame Relay, or ATM virtual connections (VCs), the VPN's key deficiency is scalability. Specifically, connection-oriented VPNs without fully meshed connections between customer sites are not optimal. MPLS-based VPNs, instead, use the peer model and Layer 3 connectionless architecture to leverage a highly scalable VPN solution. The peer model requires a customer site to peer with only one provider edge (PE) device as opposed to all other customer edge (CE) devices that are members of the VPN. The connectionless architecture allows the creation of VPNs in Layer 3, eliminating the need for tunnels or VCs.

Other scalability issues of MPLS VPNs are due to the partitioning of VPN routes between PE devices and the further partitioning of VPN and Interior Gateway Protocol (IGP) routes between PE devices and provider (P) devices in a core network.

- PE devices must maintain VPN routes for those VPNs who are members.
- P devices do not maintain any VPN routes.

This increases the scalability of the provider's core and ensures that no one device is a scalability bottleneck.

Security

MPLS VPNs offer the same level of security as connection-oriented VPNs. Packets from one VPN do not inadvertently go to another VPN.

Security is provided in the following areas:

- At the edge of a provider network, ensuring packets received from a customer are placed on the correct VPN.
- At the backbone, VPN traffic is kept separate. Malicious spoofing (an attempt to gain access to a PE device) is nearly impossible because the packets received from customers are IP packets. These IP packets must be received on a particular interface or subinterface to be uniquely identified with a VPN label.

Ease of Creation

To take full advantage of VPNs, customers must be able to easily create new VPNs and user communities. Because MPLS VPNs are connectionless, no specific point-to-point connection maps or topologies are required. You can add sites to intranets and extranets and form closed user groups. Managing VPNs in this manner enables membership of any given site in multiple VPNs, maximizing flexibility in building intranets and extranets.

Flexible Addressing

To make a VPN service more accessible, customers of a service provider can design their own addressing plan, independent of addressing plans for other service provider customers. Many customers use private address spaces, as defined in RFC 1918, and do not want to invest the time and expense of converting to public IP addresses to enable intranet connectivity. MPLS VPNs allow customers to continue to use their present address spaces without network address translation (NAT) by providing a public and private view of the address. A NAT is required only if two VPNs with overlapping address spaces want to communicate. This enables customers to use their own unregistered private addresses, and communicate freely across a public IP network.

Integrated QoS Support

QoS is an important requirement for many IP VPN customers. It provides the ability to address two fundamental VPN requirements:

- Predictable performance and policy implementation
- Support for multiple levels of service in an MPLS VPN

Network traffic is classified and labeled at the edge of the network before traffic is aggregated according to policies defined by subscribers and implemented by the provider and transported across the provider core. Traffic at the edge and core of the network can then be differentiated into different classes by drop probability or delay.

Straightforward Migration

For service providers to quickly deploy VPN services, use a straightforward migration path. MPLS VPNs are unique because you can build them over multiple network architectures, including IP, ATM, Frame Relay, and hybrid networks.

Migration for the end customer is simplified because there is no requirement to support MPLS on the CE device and no modifications are required to a customer's intranet.

How to Configure MPLS Virtual Private Networks

Configuring the Core Network

Assessing the Needs of MPLS Virtual Private Network Customers

Before you configure a Multiprotocol Label Switching virtual private network (MPLS VPN), you need to identify the core network topology so that it can best serve MPLS VPN customers. Perform this task to identify the core network topology.

SUMMARY STEPS

1. Identify the size of the network.
2. Identify the routing protocols in the core.
3. Determine if you need MPLS VPN High Availability support.
4. Determine if you need Border Gateway Protocol (BGP) load sharing and redundant paths in the MPLS VPN core.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Identify the size of the network.	Identify the following to determine the number of devices and ports that you need: <ul style="list-style-type: none"> • How many customers do you need to support? • How many VPNs are needed per customer? • How many virtual routing and forwarding instances are there for each VPN?
Step 2	Identify the routing protocols in the core.	Determine which routing protocols you need in the core network.
Step 3	Determine if you need MPLS VPN High Availability support.	MPLS VPN Nonstop Forwarding and Graceful Restart are supported on select devices and Cisco software releases. Contact Cisco Support for the exact requirements and hardware support.
Step 4	Determine if you need Border Gateway Protocol (BGP) load sharing and redundant paths in the MPLS VPN core.	For configuration steps, see the “Load Sharing MPLS VPN Traffic” feature module in the <i>MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide</i> .

Configuring MPLS in the Core

To enable Multiprotocol Label Switching (MPLS) on all devices in the core, you must configure either of the following as a label distribution protocol:

- MPLS Label Distribution Protocol (LDP). For configuration information, see the “MPLS Label Distribution Protocol (LDP)” module in the *MPLS Label Distribution Protocol Configuration Guide*.
- MPLS Traffic Engineering Resource Reservation Protocol (RSVP). For configuration information, see the “MPLS Traffic Engineering and Enhancements” module in the *MPLS Traffic Engineering Path Calculation and Setup Configuration Guide*.

Connecting the MPLS Virtual Private Network Customers

Defining VRFs on the PE Devices to Enable Customer Connectivity

Use this procedure to define a virtual routing and forwarding (VRF) configuration for IPv4. To define a VRF for IPv4 and IPv6, see the “Configuring a Virtual Routing and Forwarding Instance for IPv6” section in the “IPv6 VPN over MPLS” module in the *MPLS Layer 3 VPNs Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf vrf-name**
4. **rd route-distinguisher**

5. `route-target {import | export | both} route-target-ext-community`
6. `import map route-map`
7. `exit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>configure terminal</code></p> <p>Example:</p> <pre>Device# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p><code>ip vrf vrf-name</code></p> <p>Example:</p> <pre>Device(config)# ip vrf vpn1</pre>	<p>Defines the virtual private network (VPN) routing instance by assigning a virtual routing and forwarding (VRF) name and enters VRF configuration mode.</p> <ul style="list-style-type: none"> • The <i>vrf-name</i> argument is the name assigned to a VRF.
Step 4	<p><code>rd route-distinguisher</code></p> <p>Example:</p> <pre>Device(config-vrf)# rd 100:1</pre>	<p>Creates routing and forwarding tables.</p> <ul style="list-style-type: none"> • The <i>route-distinguisher</i> argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a route distinguisher (RD) in either of these formats: <ul style="list-style-type: none"> • 16-bit AS number:your 32-bit number, for example, 101:3 • 32-bit IP address:your 16-bit number, for example, 10.0.0.1:1
Step 5	<p><code>route-target {import export both} route-target-ext-community</code></p> <p>Example:</p> <pre>Device(config-vrf)# route-target both 100:1</pre>	<p>Creates a route-target extended community for a VRF.</p> <ul style="list-style-type: none"> • The import keyword imports routing information from the target VPN extended community. • The export keyword exports routing information to the target VPN extended community. • The both keyword imports routing information from and exports routing information to the target VPN extended community. • The <i>route-target-ext-community</i> argument adds the route-target extended community attributes to the VRF's list of import, export, or both route-target extended communities.

	Command or Action	Purpose
Step 6	import map <i>route-map</i> Example: Device(config-vrf)# import map vpn1-route-map	(Optional) Configures an import route map for a VRF. <ul style="list-style-type: none"> The <i>route-map</i> argument specifies the route map to be used as an import route map for the VRF.
Step 7	exit Example: Device(config-vrf)# exit	(Optional) Exits to global configuration mode.

Configuring VRF Interfaces on PE Devices for Each VPN Customer

To associate a virtual routing and forwarding (VRF) instance with an interface or subinterface on the provider edge (PE) devices, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# interface FastEthernet 1/0/0GigabitEthernet 0/0/1	Specifies the interface to configure and enters interface configuration mode. <ul style="list-style-type: none"> The <i>type</i> argument specifies the type of interface to be configured. The <i>number</i> argument specifies the port, connector, or interface card number.
Step 4	ip vrf forwarding <i>vrf-name</i> Example:	Associates a VRF with the specified interface or subinterface.

	Command or Action	Purpose
	Device(config-if)# ip vrf forwarding vpn1	<ul style="list-style-type: none"> The <i>vrf-name</i> argument is the name assigned to a VRF.
Step 5	end Example: Device(config-if)# end	(Optional) Exits to privileged EXEC mode.

Configuring Routing Protocols Between the PE and CE Devices

Configure the provider edge (PE) device with the same routing protocol that the customer edge (CE) device uses. You can configure the Border Gateway Protocol (BGP), Routing Information Protocol version 2 (RIPv2), or static routes between the PE and CE devices.

Configuring RIPv2 as the Routing Protocol Between the PE and CE Devices

SUMMARY STEPS

1. enable
2. configure terminal
3. router rip
4. version {1 | 2}
5. address-family ipv4 [multicast | unicast | vrf vrf-name]
6. network ip-address
7. redistribute protocol [process-id] {level-1 | level-1-2 | level-2} [as-number] [metric metric-value] [metric-type type-value] [match {internal | external 1 | external 2}] [tag tag-value] [route-map map-tag] [subnets]
8. exit-address-family
9. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router rip Example: Device(config)# router rip	Enables the Routing Information Protocol (RIP).

	Command or Action	Purpose
Step 4	version {1 2} Example: Device(config-router)# version 2	Specifies RIP version used globally by the device.
Step 5	address-family ipv4 [multicast unicast vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 vrf vpn1	Specifies the IPv4 address family type and enters address family configuration mode. <ul style="list-style-type: none"> • The multicast keyword specifies IPv4 multicast address prefixes. • The unicast keyword specifies IPv4 unicast address prefixes. • The vrf <i>vrf-name</i> keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 6	network <i>ip-address</i> Example: Device(config-router-af)# network 192.168.7.0	Enables RIP on the PE-to-CE link.
Step 7	redistribute <i>protocol</i> [<i>process-id</i>] {level-1 level-1-2 level-2} [<i>as-number</i>] [<i>metric metric-value</i>] [<i>metric-type type-value</i>] [match {internal external 1 external 2}] [tag <i>tag-value</i>] [route-map <i>map-tag</i>] [subnets] Example: Device(config-router-af)# redistribute bgp 200	Redistributes routes from one routing domain into another routing domain. <ul style="list-style-type: none"> • For the RIPv2 routing protocol, use the redistribute bgp <i>as-number</i> command.
Step 8	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode.
Step 9	end Example: Device(config-router)# end	(Optional) Exits to privileged EXEC mode.

Configuring Static Routes Between the PE and CE Devices

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route vrf *vrf-name***

4. **address-family ipv4** [**multicast** | **unicast** | **vrf vrf-name**]
5. **redistribute protocol** [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric metric-value**] [**metric-type type-value**] [**match** {**internal** | **external 1** | **external 2**}] [**tag tag-value**] [**route-map map-tag**] [**subnets**]
6. **redistribute protocol** [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric metric-value**] [**metric-type type-value**] [**match** {**internal** | **external 1** | **external 2**}] [**tag tag-value**] [**route-map map-tag**] [**subnets**]
7. **exit-address-family**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip route vrf vrf-name Example: Device(config)# ip route vrf vpn1	Defines static route parameters for every provider edge-to-customer edge (PE-to-CE) session and enters router configuration mode.
Step 4	address-family ipv4 [multicast unicast vrf vrf-name] Example: Device(config-router)# address-family ipv4 vrf vpn1	Specifies the IPv4 address family type and enters address family configuration mode. <ul style="list-style-type: none"> • The multicast keyword specifies IPv4 multicast address prefixes. • The unicast keyword specifies IPv4 unicast address prefixes. • The vrf vrf-name keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	redistribute protocol [<i>process-id</i>] { level-1 level-1-2 level-2 } [<i>as-number</i>] [metric metric-value] [metric-type type-value] [match { internal external 1 external 2 }] [tag tag-value] [route-map map-tag] [subnets] Example: Device(config-router-af)# redistribute static	Redistributes routes from one routing domain into another routing domain. <ul style="list-style-type: none"> • To redistribute virtual routing and forwarding (VRF) static routes into the VRF Border Gateway Protocol (BGP) table, use the redistribute static command. See the command reference page for information about other arguments and keywords.

	Command or Action	Purpose
Step 6	<p>redistribute <i>protocol</i> [<i>process-id</i>] {level-1 level-1-2 level-2} [<i>as-number</i>] [metric <i>metric-value</i>] [metric-type <i>type-value</i>] [match {internal external 1 external 2}] [tag <i>tag-value</i>] [route-map <i>map-tag</i>] [subnets]</p> <p>Example:</p> <pre>Device(config-router-af)# redistribute connected</pre>	<p>Redistributes routes from one routing domain into another routing domain.</p> <ul style="list-style-type: none"> To redistribute directly connected networks into the VRF BGP table, use the redistribute connected command.
Step 7	<p>exit-address-family</p> <p>Example:</p> <pre>Device(config-router-af)# exit-address-family</pre>	Exits address family configuration mode.
Step 8	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	(Optional) Exits to privileged EXEC mode.

Verifying the Virtual Private Network Configuration

A route distinguisher must be configured for the virtual routing and forwarding (VRF) instance, and Multiprotocol Label Switching (MPLS) must be configured on the interfaces that carry the VRF. Use the **show ip vrf** command to verify the route distinguisher (RD) and interface that are configured for the VRF.

SUMMARY STEPS

1. **show ip vrf**

DETAILED STEPS

show ip vrf

Displays the set of defined VRF instances and associated interfaces. The output also maps the VRF instances to the configured route distinguisher.

Verifying Connectivity Between MPLS Virtual Private Network Sites

To verify that the local and remote customer edge (CE) devices can communicate across the Multiprotocol Label Switching (MPLS) core, perform the following tasks:

Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

DETAILED STEPS

-
- Step 1** **enable**
Enables privileged EXEC mode.
- Step 2** **ping** [*protocol*] {*host-name* | *system-address*}
- Diagnoses basic network connectivity on AppleTalk, Connectionless-mode Network Service (CLNS), IP, Novell, Apollo, Virtual Integrated Network Service (VINES), DECnet, or Xerox Network Service (XNS) networks. Use the **ping** command to verify the connectivity from one CE device to another.
- Step 3** **trace** [*protocol*] [*destination*]
- Discovers the routes that packets take when traveling to their destination. The **trace** command can help isolate a trouble spot if two devices cannot communicate.
- Step 4** **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]
- Displays the current state of the routing table. Use the *ip-address* argument to verify that CE1 has a route to CE2. Verify the routes learned by CE1. Make sure that the route for CE2 is listed.
-

Verifying That the Local and Remote CE Devices Are in the PE Routing Table

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show ip cef vrf** *vrf-name* [*ip-prefix*]

DETAILED STEPS

-
- Step 1** **enable**
Enables privileged EXEC mode.
- Step 2** **show ip route vrf** *vrf-name* [*prefix*]
- Displays the IP routing table associated with a virtual routing and forwarding (VRF) instance. Check that the loopback addresses of the local and remote customer edge (CE) devices are in the routing table of the provider edge (PE) devices.

Step 3 `show ip cef vrf vrf-name [ip-prefix]`

Displays the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE device is in the Cisco Express Forwarding table.

Configuration Examples for MPLS Virtual Private Networks

Example: Configuring an MPLS Virtual Private Network Using RIP

PE Configuration	CE Configuration
	<pre>ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface FastEthernet 0/0/0GigabitEthernet 0/0/0 ip address 192.0.2.1 255.255.255.0 no cdp enable router rip version 2 timers basic 30 60 60 120 redistribute connected network 10.0.0.0 network 192.0.2.0 no auto-summary</pre>

PE Configuration	CE Configuration
<pre> ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface FastEthernet 0/0/0GigabitEthernet 0/0/0 ip vrf forwarding vpn1 ip address 192.0.2.3 255.255.255.0 no cdp enable interface FastEthernet 1/1/0GigabitEthernet 0/0/1 ip address 192.0.2.2 255.255.255.0 mpls label protocol ldp mpls ip ! router rip version 2 timers basic 30 60 60 120 ! address-family ipv4 vrf vpn1 version 2 redistribute bgp 100 metric transparent network 192.0.2.0 distribute-list 20 in no auto-summary exit-address-family ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute rip no auto-summary no synchronization exit-address-family </pre>	

Example: Configuring an MPLS Virtual Private Network Using Static Routes

PE Configuration	CE Configuration
<pre> ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface FastEthernet 0/0/0GigabitEthernet 0/0/0 ip vrf forwarding vpn1 ip address 192.0.2.3 255.255.255.0 no cdp enable ! interfaceFastEthernet 1/1/0GigabitEthernet 0/0/1 ip address 192.168.0.1 255.255.0.0 mpls label protocol ldp mpls ip ! router ospf 100 network 10.0.0. 0.0.0.0 area 100 network 192.168.0.0 255.255.0.0 area 100 ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute static no auto-summary no synchronization exit-address-family ! ip route vrf vpn1 10.0.0.9 255.255.255.255 192.0.2.2 ip route vrf vpn1 192.0.2.0 255.255.0.0 192.0.2.2 </pre>	<pre> ip cef ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface FastEthernet 0/0/0GigabitEthernet 0/0/0 ip address 192.0.2.2 255.255.0.0 no cdp enable ! ip route 10.0.0.9 255.255.255.255 192.0.2.3 3 ip route 198.51.100.0 255.255.255.0 192.0.2.3 3 </pre>

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Description of commands associated with MPLS and MPLS applications	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring Cisco Express Forwarding	“Configuring Basic Cisco Express Forwarding” module in the <i>Cisco Express Forwarding Configuration Guide</i>
Border Gateway Protocol (BGP) load sharing	“Load Sharing MPLS VPN Traffic” module in the <i>MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide</i>
Configuring LDP	“MPLS Label Distribution Protocol (LDP)” module in the <i>MPLS Label Distribution Protocol Configuration Guide</i>
Configuring MPLS Traffic Engineering Resource Reservation Protocol (RSVP)	“MPLS Traffic Engineering and Enhancements” module in the <i>MPLS Traffic Engineering Path Calculation and Setup Configuration Guide</i>
IPv6 VPN over MPLS	“IPv6 VPN over MPLS” module in the <i>MPLS Layer 3 VPNs Configuration Guide</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 3

Multiprotocol BGP MPLS VPN

A Multiprotocol Label Switching (MPLS) virtual private network (VPN) consists of a set of sites that are interconnected by means of an MPLS provider core network. At each site, there are one or more customer edge (CE) devices, which attach to one or more provider edge (PE) devices. PEs use the Multiprotocol-Border Gateway Protocol (MP-BGP) to dynamically communicate with each other.

- [Prerequisites for Multiprotocol BGP MPLS VPN, on page 25](#)
- [Information About Multiprotocol BGP MPLS VPN, on page 25](#)
- [How to Configure Multiprotocol BGP MPLS VPN, on page 28](#)
- [Configuration Examples for Multiprotocol BGP MPLS VPN, on page 35](#)
- [Additional References, on page 36](#)

Prerequisites for Multiprotocol BGP MPLS VPN

Configure MPLS virtual private networks (VPNs) in the core.

Information About Multiprotocol BGP MPLS VPN

MPLS Virtual Private Network Definition

Before defining a Multiprotocol Label Switching virtual private network (MPLS VPN), you must define a VPN in general. 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, because 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 the customer's involvement.

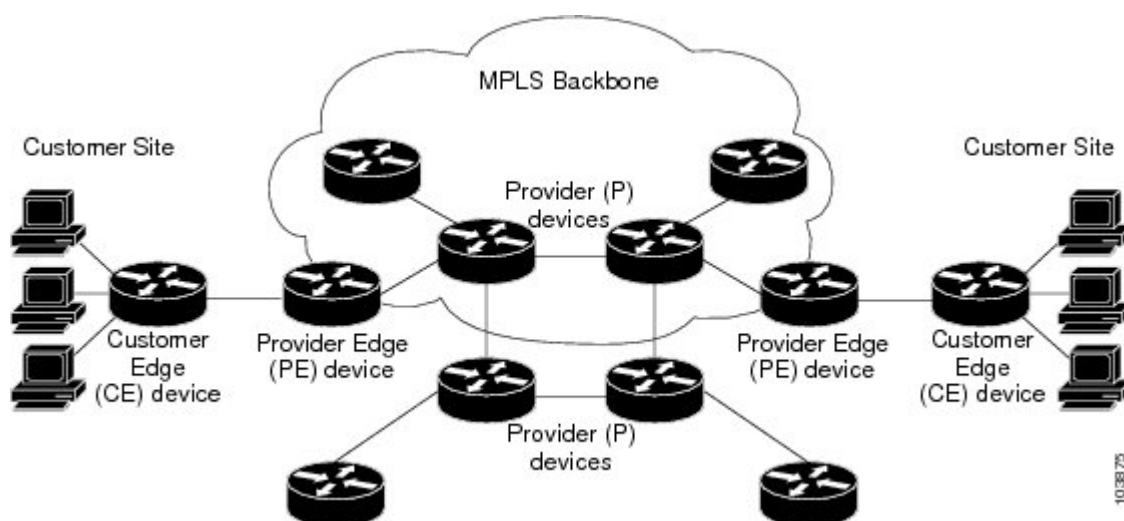
MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the service provider's edge device that provides services to the customer site needs to be updated.

The different parts of the MPLS VPN are described as follows:

- Provider (P) device—Device in the core of the provider network. P devices run MPLS switching, and do not attach VPN labels to routed packets. The MPLS label in each route is assigned by the provider edge (PE) device. VPN labels are used to direct data packets to the correct egress device.
- PE device—Device that attaches the VPN label to incoming packets based on the interface or subinterface on which they are received. A PE device attaches directly to a customer edge (CE) device.
- Customer (C) device—Device in the ISP or enterprise network.
- CE device—Edge device on the network of the ISP that connects to the PE device on the network. A CE device must interface with a PE device.

The figure below shows a basic MPLS VPN.

Figure 2: Basic MPLS VPN Terminology



How an MPLS Virtual Private Network Works

Multiprotocol Label Switching virtual private network (MPLS VPN) functionality is enabled at the edge of an MPLS network. The provider edge (PE) device performs the following:

- Exchanges routing updates with the customer edge (CE) device.
- Translates the CE routing information into VPNv4 routes.
- Exchanges VPNv4 routes with other PE devices through the Multiprotocol Border Gateway Protocol (MP-BGP).

The following sections describe how MPLS VPN works:

How Virtual Routing and Forwarding Tables Work in an MPLS Virtual Private Network

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

- An IP routing table
- A derived Cisco Express Forwarding 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

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 site's 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 Cisco Express Forwarding table for each VRF. A separate set of routing and Cisco Express Forwarding tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN, and they also prevent packets that are outside a VPN from being forwarded to a device within the VPN.

How VPN Routing Information Is Distributed in an MPLS Virtual Private Network

The distribution of virtual private network (VPN) routing information is controlled through the use of VPN route target communities, implemented by Border Gateway Protocol (BGP) extended communities. VPN routing information is distributed as follows:

- When a VPN route that is learned from a customer edge (CE) device is injected into 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 virtual routing and forwarding (VRF) instance 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 in order 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 provider edge (PE) device can learn an IP prefix from the following sources:

- A customer edge (CE) device by static configuration
- A Border Gateway Protocol (BGP) session with the CE device
- A Routing Information Protocol (RIP) exchange with the CE device

The IP prefix is a member of the IPv4 address family. After the PE device learns the IP prefix, the PE converts it into a VPN-IPv4 prefix by combining it with an 8-byte route distinguisher (RD). 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 a configuration command associated with the virtual routing and forwarding (VRF) instance on the PE device.

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

- Within an IP domains, known as an autonomous system (interior BGP [IBGP])
- Between autonomous systems (external BGP [EBGP])

PE-PE or PE-RR (route reflector) sessions are IBGP sessions, and PE-CE sessions are EBGP sessions. In an Enhanced Interior Gateway Routing Protocol (EIGRP) PE-CE environment, when an EIGRP internal route is redistributed into BGP by one PE, and then back into EIGRP by another PE, the originating router ID for the route is set to the router ID of the second PE, replacing the original internal router ID.

BGP propagates reachability information for VPN-IPv4 prefixes among PE devices by means of the BGP multiprotocol extensions (refer to 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.

Major Components of an MPLS Virtual Private Network

An Multiprotocol Label Switching (MPLS)-based virtual private network (VPN) 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 VPN community provider edge (PE) devices—MP-BGP propagates virtual routing and forwarding (VRF) reachability information to all members of a VPN community. MP-BGP peering must be configured on all PE devices 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.

How to Configure Multiprotocol BGP MPLS VPN

Configuring Multiprotocol BGP Connectivity on the PE Devices and Route Reflectors

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*

6. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type*
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **address-family** *vpn4* [**unicast**]
9. **neighbor** {*ip-address* | *peer-group-name*} **send-community** **extended**
10. **neighbor** {*ip-address* | *peer-group-name*} **activate**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 100	Configures a Border Gateway Protocol (BGP) routing process and enters router configuration mode. <ul style="list-style-type: none"> • The <i>as-number</i> argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks are 64512 to 65535.
Step 4	no bgp default ipv4-unicast Example: Device(config-router)# no bgp default ipv4-unicast	(Optional) Disables the IPv4 unicast address family on all neighbors. <ul style="list-style-type: none"> • Use the no bgp default ipv4-unicast command if you are using this neighbor for Multiprotocol Label Switching (MPLS) routes only.
Step 5	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 10.0.0.1 remote-as 100	Adds an entry to the BGP or multiprotocol BGP neighbor table. <ul style="list-style-type: none"> • The <i>ip-address</i> argument specifies the IP address of the neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group. • The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> } update-source <i>interface-type</i>	Allows the BGP sessions to use a specific operational interface for TCP connections.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-router)# neighbor 10.0.0.1 update-source Loopback0</pre>	<p>The loopback interface is the most commonly-used interface type.</p> <p>If you specify a BGP peer group by using the <i>peer-group-name</i> argument, all the members of the peer group inherit the characteristic configured with this command.</p>
Step 7	<p>neighbor {ip-address peer-group-name} activate</p> <p>Example:</p> <pre>Device(config-router)# neighbor 10.0.0.1 activate</pre>	<p>Enables the exchange of information with a neighboring BGP device.</p> <ul style="list-style-type: none"> • The <i>ip-address</i> argument specifies the IP address of the neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group.
Step 8	<p>address-family vpnv4 [unicast]</p> <p>Example:</p> <pre>Device(config-router)# address-family vpnv4</pre>	<p>Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.</p> <ul style="list-style-type: none"> • The optional unicast keyword specifies VPNv4 unicast address prefixes.
Step 9	<p>neighbor {ip-address peer-group-name} send-community extended</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.1 send-community extended</pre>	<p>Specifies that a communities attribute should be sent to a BGP neighbor.</p> <ul style="list-style-type: none"> • The <i>ip-address</i> argument specifies the IP address of the BGP-speaking neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group.
Step 10	<p>neighbor {ip-address peer-group-name} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.1 activate</pre>	<p>Enables the exchange of information with a neighboring BGP device.</p> <ul style="list-style-type: none"> • The <i>ip-address</i> argument specifies the IP address of the neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group.
Step 11	<p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre>	<p>(Optional) Exits to privileged EXEC mode.</p>

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command is not successful, enter a **debug ip bgp ip-address events** command, where *ip-address* is the IP address of the neighbor.

Configuring BGP as the Routing Protocol Between the PE and CE Devices

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **exit-address-family**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 100	Configures a Border Gateway Protocol (BGP) routing process and enters router configuration mode. <ul style="list-style-type: none"> • The <i>as-number</i> argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	address-family ipv4 [multicast unicast vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4 vrf vpn1	Specifies the IPv4 address family type and enters address family configuration mode. <ul style="list-style-type: none"> • The multicast keyword specifies IPv4 multicast address prefixes. • The unicast keyword specifies IPv4 unicast address prefixes. • The vrf <i>vrf-name</i> keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.

	Command or Action	Purpose
Step 5	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i></p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.1 remote-as 200</pre>	<p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p> <ul style="list-style-type: none"> • The <i>ip-address</i> argument specifies the IP address of the neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group. • The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.
Step 6	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 10.0.0.1 activate</pre>	<p>Enables the exchange of information with a neighboring BGP device.</p> <ul style="list-style-type: none"> • The <i>ip-address</i> argument specifies the IP address of the neighbor. • The <i>peer-group-name</i> argument specifies the name of a BGP peer group.
Step 7	<p>exit-address-family</p> <p>Example:</p> <pre>Device(config-router-af)# exit-address-family</pre>	<p>Exits address family configuration mode.</p>
Step 8	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	<p>(Optional) Exits to privileged EXEC mode.</p>

Verifying the Virtual Private Network Configuration

A route distinguisher must be configured for the virtual routing and forwarding (VRF) instance, and Multiprotocol Label Switching (MPLS) must be configured on the interfaces that carry the VRF. Use the **show ip vrf** command to verify the route distinguisher (RD) and interface that are configured for the VRF.

SUMMARY STEPS

1. **show ip vrf**

DETAILED STEPS

show ip vrf

Displays the set of defined VRF instances and associated interfaces. The output also maps the VRF instances to the configured route distinguisher.

Verifying Connectivity Between MPLS Virtual Private Network Sites

To verify that the local and remote customer edge (CE) devices can communicate across the Multiprotocol Label Switching (MPLS) core, perform the following tasks:

Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*] | [**list** [*access-list-name* | *access-list-number*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **ping** [*protocol*] {*host-name* | *system-address*}

Diagnoses basic network connectivity on AppleTalk, Connectionless-mode Network Service (CLNS), IP, Novell, Apollo, Virtual Integrated Network Service (VINES), DECnet, or Xerox Network Service (XNS) networks. Use the **ping** command to verify the connectivity from one CE device to another.

Step 3 **trace** [*protocol*] [*destination*]

Discovers the routes that packets take when traveling to their destination. The **trace** command can help isolate a trouble spot if two devices cannot communicate.

Step 4 **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*] | [**list** [*access-list-name* | *access-list-number*]

Displays the current state of the routing table. Use the *ip-address* argument to verify that CE1 has a route to CE2. Verify the routes learned by CE1. Make sure that the route for CE2 is listed.

Verifying That the Local and Remote CE Devices Are in the PE Routing Table

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show ip cef vrf** *vrf-name* [*ip-prefix*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 `show ip route vrf vrf-name [prefix]`

Displays the IP routing table associated with a virtual routing and forwarding (VRF) instance. Check that the loopback addresses of the local and remote customer edge (CE) devices are in the routing table of the provider edge (PE) devices.

Step 3 `show ip cef vrf vrf-name [ip-prefix]`

Displays the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE device is in the Cisco Express Forwarding table.

Configuration Examples for Multiprotocol BGP MPLS VPN

Example: Configuring an MPLS Virtual Private Network Using BGP

PE Configuration	CE Configuration
<pre> ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface FastEthernet0/0/0GigabitEthernet0/0/0 ip vrf forwarding vpn1 ip address 192.0.2.3 255.255.255.0 no cdp enable ! interface FastEthernet1/1/0GigabitEthernet0/0/1 ip address 192.0.2.2 255.255.255.0 mpls label protocol ldp mpls ip ! router ospf 100 network 10.0.0. 0.0.0.0 area 100 network 192.0.2.1 255.255.255.0 area 100 ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected neighbor 198.51.100.1 remote-as 200 neighbor 198.51.100.1 activate neighbor 198.51.100.1 as-override neighbor 198.51.100.1 advertisement-interval 5 no auto-summary no synchronization exit-address-family </pre>	<pre> ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface FastEthernet0/0GigabitEthernet0/0/0 ip address 198.51.100.1 255.255.255.0 no cdp enable ! router bgp 200 bgp log-neighbor-changes neighbor 198.51.100.2 remote-as 100 ! address-family ipv4 redistribute connected neighbor 198.51.100.2 activate neighbor 198.51.100.2 advertisement-interval 5 no auto-summary no synchronization exit-address-family </pre>

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Description of commands associated with MPLS and MPLS applications	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring MPLS virtual private networks	“MPLS Virtual Private Networks” module in the <i>MPLS Layer 3 VPNs Configuration Guide</i>

Standards and RFCs

RFC	Title
RFC 2283	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2547	<i>BGP/MPLS VPNs</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 4

IPv6 VPN over MPLS

The Border Gateway Protocol over Multiprotocol Label Switching VPN feature is an implementation of the provider edge (PE)-based Virtual Private Network (VPN) model. In principle, there is no difference between IPv4 and IPv6 VPNs. In both IPv4 and IPv6, multiprotocol Border Gateway Protocol (BGP) is the center of the Multiprotocol Label Switching (MPLS) VPN for IPv6 (VPNv6) architecture. It is used to distribute IPv6 routes over the service provider backbone, using the same procedures to work with overlapping addresses, redistribution policies, and scalability issues.

Effective Cisco IOS XE Release 3.18SP, this feature is supported on the Cisco RSP3 module.

- [Prerequisites for IPv6 VPN over MPLS, on page 37](#)
- [Restrictions for IPv6 VPN over MPLS, on page 38](#)
- [Information About IPv6 VPN over MPLS, on page 38](#)
- [How to Configure IPv6 VPN over MPLS, on page 44](#)
- [Configuration Examples for IPv6 VPN over MPLS, on page 81](#)
- [Additional References, on page 87](#)
- [Feature Information for IPv6 VPN over MPLS, on page 88](#)
- [Glossary, on page 88](#)

Prerequisites for IPv6 VPN over MPLS

Your network must be running the following services before you configure IPv6 VPN operation:

- Multiprotocol Label Switching (MPLS) in provider backbone devices
- MPLS with Virtual Private Network (VPN) code in provider devices with VPN provider edge (PE) devices
- Border Gateway Protocol (BGP) in all devices providing a VPN service
- Cisco Express Forwarding switching in every MPLS-enabled device
- Class of Service (CoS) feature

Restrictions for IPv6 VPN over MPLS

IPv6 VPN over MPLS (6VPE) supports a Multiprotocol Label Switching (MPLS) IPv4-signaled core. An MPLS IPv6-signaled core is not supported.

Information About IPv6 VPN over MPLS

IPv6 VPN over MPLS Overview

Multiprotocol Border Gateway Protocol (BGP) is the center of the Multiprotocol Label Switching (MPLS) IPv6 Virtual Private Network (VPN) architecture in both IPv4 and IPv6. It is used to distribute IPv6 routes over the service provider backbone, using the same procedures to work with overlapping addresses, redistribution policies, and scalability issues.

Although IPv6 should not have overlapping address space, IPv6 addresses are prepended with a route distinguisher (RD). A network layer reachability information (NLRI) 3-tuple format (which contains length, IPv6 prefix, and label) is defined to distribute these routes using multiprotocol BGP. The extended community attribute (for example, the route target) is used to control redistribution of routing information by tagging exported routes and filtering imported ones.

For scalability, route reflectors can be used to concentrate routing paths and avoid a full provider edge (PE) mesh. BGP features in IPv6, such as route refresh, automatic route filtering, and outbound route filtering, help reduce the number of routes held in each PE. This document focuses on the following differences between IPv6 and IPv4:

- Creation of a new multiprotocol BGP IPv6 VPN address family and specification of a IPv6 VPN address format
- Specification of a new IPv6 VPN NLRI
- Specification of BGP next-hop encoding when the device has an IPv4-based MPLS core

Some IPv6 VPN features, such as interprovider and Carrier Supporting Carrier (CSC) topologies, are specific to BGP-MPLS IPv6 VPN. Others, such as the link between Autonomous System Boundary Routers (ASBRs), might support IPv4 only, IPv6 only, or both, regardless of the address family being transported.

Addressing Considerations for IPv6 VPN over MPLS

Regardless of the Virtual Private Network (VPN) model deployed, an addressing plan must be defined for the VPN that allows hosts to communicate with other sites using one site within one VPN, as well as with public resources.

VPN IPv4 sites often use private addressing for their addressing plan. These addresses do not need to be registered, and they are not routable on the public network. Whenever a host within a private site needs to access a public domain, it goes through a device that finds a public address on its behalf. With IPv4, this can be a network address translator or an application proxy.

Given the larger address space available with IPv6, the easiest approach to IPv6 addressing is to use IPv6 global addresses for the private addressing plan. Another approach is to use unique local addresses (ULAs). ULAs are easy to filter at site boundaries based on their scope. ULAs are also Internet service provider

(ISP)-independent and can be used for communications inside a site without any permanent or intermittent Internet connectivity.

In IPv6 VPN over MPLS (6VPE), ULAs are treated as regular global addresses. The device configuration filters ULA prefixes to prevent them from appearing in the public domain. Link-local addresses on the peer will not be announced by Border Gateway Protocol (BGP) (IPv6 or IPv6 VPN) speakers.

A host within a private site that needs to access a public domain can do so through an IPv6 application proxy (such as a web proxy for accessing web pages), which accesses the public resource on the host's behalf with a global routable address, or the host can use a public address of its own. In the latter case, if ULAs have been deployed, the IPv6 host also is configured with a routable global address. A source address selection algorithm is used to select one or the other, based on the destination address.

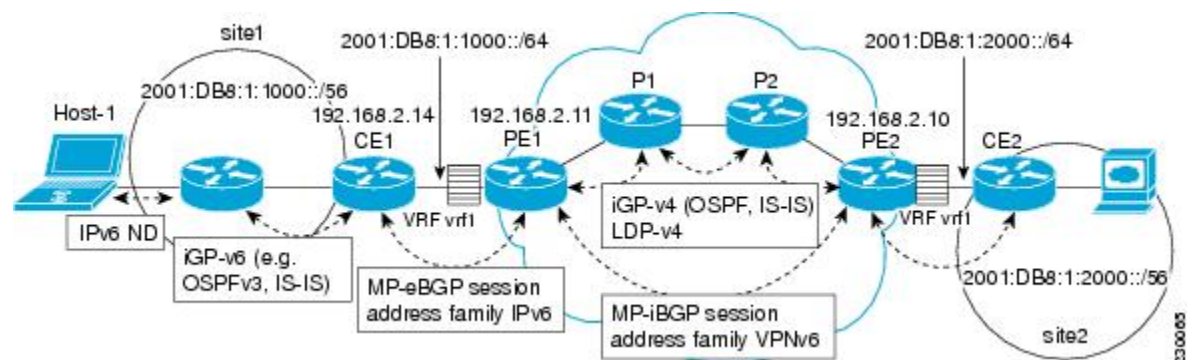
Basic IPv6 VPN over MPLS Functionality

IPv6 VPN over MPLS (6VPE) takes advantage of the coexistence between IPv6 and IPv4 by leveraging an existent Multiprotocol Label Switching (MPLS) IPv4 core network:

IPv6 VPN Architecture Overview

The figure below illustrates the important aspects of the IPv6 Virtual Private Network (VPN) architecture.

Figure 3: Simple IPv6 VPN Architecture



The customer edge (CE) devices are connected to the provider's backbone using provider edge (PE) devices. The PE devices are connected using provider (P1 and P2 in the figure above) devices. The provider (P) devices are unaware of VPN routes, and, in the case of IPv6 over MPLS (6VPE), might support only IPv4. Only PE devices perform VPN-specific tasks. For 6VPE, the PE devices are dual-stack (IPv4 and IPv6) devices.

The routing component of the VPN operation is divided into core routing and edge routing. Core routing, which involves PE devices and P devices, typically is performed by an IPv4 Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). In the figure above, the IGP distributes only routes internal to the provider's autonomous system. The core routing enables connectivity among P and PE devices.

Edge routing takes place in two directions: routing between PE pairs and routing between a PE and a CE. Routing between PE pairs is achieved using multiprotocol internal BGP (iBGP) using the IPv6 VPN address family. This method distributes routes learned from CEs through PE-CE routing, using appropriate route export policies at the ingress PE device and appropriate route import policies at the egress PE device.

Routing between the CE and its PE is achieved using a routing protocol that is VPN routing and forwarding (VRF) aware. Static routes, external BGP (eBGP), and Enhanced Interior Gateway Routing Protocol (EIGRP)

are VRF-instance aware. In the figure above, eBGP is used between the CE (CE1) and the PE (PE1). At the same time, the CE runs an IPv6 IGP within the VPN site (site1 in the figure above). The CE redistributes IGP routes into multiprotocol-eBGP address family IPv6. At the PE, these routes are installed in the VRF named `vrf1`, and forwarded to the remote PEs (PE2 in the figure above), according to export policies defined for this VRF.

IPv6 VPN Next Hop

When the device announces a prefix using the `MP_REACH_NLRI` attribute, the Multiprotocol Border Gateway Protocol (MP-BGP) running on one provider edge (PE) inserts a BGP next hop in the update message sent to a remote PE. This next hop is either propagated from the received update (for instance, if the PE is a route reflector), or it is the address of the PE sending the update message (the egress PE).

For the IPv6 Virtual Private Network (VPN) address family, the next hop must be an IPv6 VPN address, regardless of the nature of the network between the PE speakers. Because the route distinguisher (RD) has no significance (the address is not part of any VPN), it is set to 0. If the provider network is a native IPv6 network, the remaining part of the next hop is the IPv6 address of the egress PE. Otherwise, it is an IPv4 address used as an IPv6-mapped address (for example, `::FFFF:IPv4-address`).

MPLS Forwarding

When it receives IPv6 traffic from one customer site, the ingress provider edge (PE) device uses Multiprotocol Label Switching (MPLS) to tunnel IPv6 Virtual Private Network (VPN) packets over the backbone toward the egress PE device identified as the Border Gateway Protocol (BGP) next hop. The ingress PE device prepends the IPv6 packets with the outer and inner labels before putting the packet on the egress interface.

Under normal operation, a provider (P) device along the forwarding path does not look inside the frame beyond the first label. The provider (P) device either swaps the incoming label with an outgoing one or removes the incoming label if the next device is a PE device. Removing the incoming label is called penultimate hop popping. The remaining label (BGP label) is used to identify the egress PE interface toward the customer site. The label also hides the protocol version (IPv6) from the last P device, which it would otherwise need to forward an IPv6 packet.

A P device is ignorant of the IPv6 VPN routes. The IPv6 header remains hidden under one or more MPLS labels. When the P device receives an MPLS-encapsulated IPv6 packet that cannot be delivered, it has two options. If the P device is IPv6 aware, it exposes the IPv6 header, builds an Internet Control Message Protocol (ICMP) for IPv6 message, and sends the message, which is MPLS encapsulated, to the source of the original packet. If the P device is not IPv6 aware, it drops the packet.

6VPE over GRE Tunnels

In some Cisco software releases, the ingress provider edge (PE) device uses IPv4 generic routing encapsulation (GRE) tunnels combined with IPv6 over MPLS (6VPE) over Multiprotocol Label Switching (MPLS) to tunnel IPv6 Virtual Private Network (VPN) packets over the backbone toward the egress PE device identified as the Border Gateway Protocol (BGP) next hop.

VRF Concepts

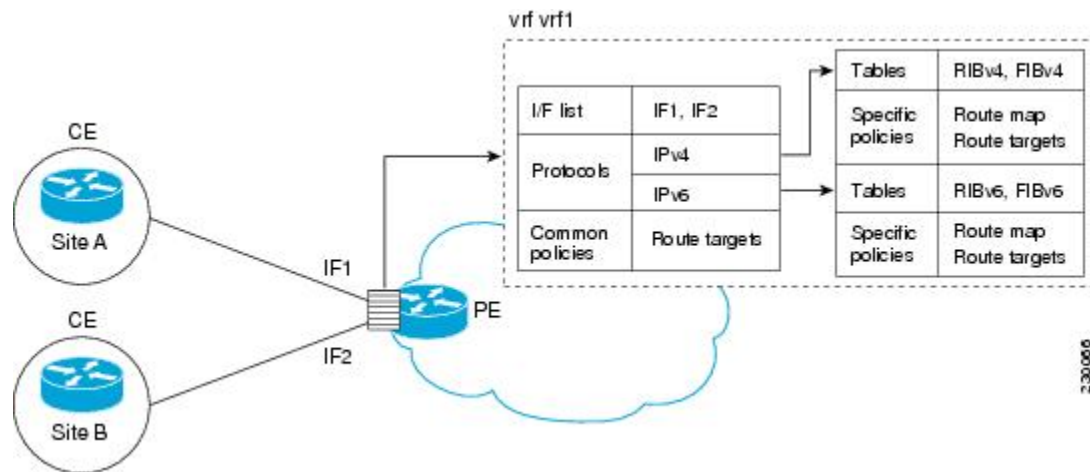
A virtual routing and forwarding (VRF) entity works with a private customer-specific Routing Information Base (RIB) and Forwarding Information Base (FIB). Although IPv4 and IPv6 routing tables are distinct, it is convenient for the two protocols to share the same VRF for a specific customer.

IPv6 VPN customers are likely to be existing VPNv4 customers that are either deploying dual-stack hosts and devices or shadowing some of their IPv4 infrastructure with IPv6 nodes. Several deployment models are

possible. Some customers use separate logical interfaces for IPv4 and IPv6 and define separate VRFs on each. Although this approach provides flexibility to configure separate policies for IPv4 and IPv6, it prevents sharing the same policy. Another approach, the multiprotocol VRF, keeps a single VRF on the provider edge-customer edge (PE-CE) interface, and enables it for IPv4, IPv6, or both. It is then possible to define common or separate policies for each IP version. With this approach, a VRF is better defined as the set of tables, interfaces, and policies found at the PE, and is used by sites of a particular VPN connected to this PE.

The figure below illustrates the multiprotocol VRF, in which the VRF named `vrf1` is enabled for both IPv4 and IPv6 and is associated with two interfaces (IF1, IF2), two sets of tables (IPv4 RIB and FIB and IPv6 RIB and FIB), and a set of common or distinct policies.

Figure 4: Multiprotocol VRF



IPv6 VPN Scalability

Provider edge (PE)-based Virtual Private Networks (VPNs) such as Border Gateway Protocol-Multiprotocol Label Switching (BGP-MPLS) IPv6 VPN scale better than customer edge (CE)-based VPNs. A network designer must consider scaling when designing the network. The following points need to be considered:

- Routing table size, which includes the size of virtual routing and forwarding (VRF) tables and BGP tables
- Number of BGP sessions, which grows as a square number of PEs

Routing table size concerns occur with PEs that handle many customer sites. Not only do these PEs have one Routing Information Base (RIB) and Forwarding Information Base (FIB) per connected customer, but also the PEs' BGP tables, which total all entries from individual VRFs, grow accordingly. Another scalability problem occurs when the number of PEs in the provider network grows beyond a certain level. Assuming that a significant number of sites belonging to the same VPN are spread over many PEs, the number of multiprotocol BGP sessions may rapidly become prohibitive: $(n - 1) \times n / 2$, where n is the number of PEs.

The following features are included in IPv6 VPN over MPLS:

- Route refresh and automatic route filtering—Limits the size of routing tables, because only routes imported into a VRF are kept locally. When the import policy changes, a route refresh can be sent to query a retransmission of routing updates.
- Outbound route filtering (ORF)—Allows the ingress PE to advertise filters to the egress PE so that updates are not sent unnecessarily over the network.

- Route reflectors—Route reflectors (RRs) are internal BGP (iBGP) peers that propagate iBGP routes learned from other iBGP peers. RR's are used to concentrate iBGP sessions.

Advanced IPv6 MPLS VPN Functionality

Advanced Multiprotocol Label Switching (MPLS) features such as accessing the Internet from a Virtual Private Network (VPN) for IPv4, multi-autonomous-system backbones, and Carrier Supporting Carriers (CSCs) are generally the same for IPv6 as for IPv4. However, there are differences in addressing and in the way IPv6 over MPLS (6VPE) operates over an IPv4 backbone.

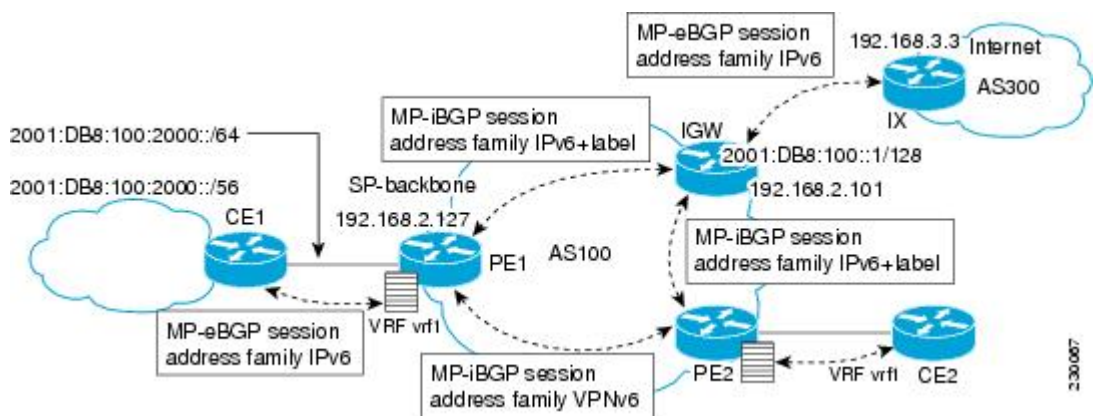
The following sections describe concepts for advanced IPv6 MPLS VPN functionality:

Internet Access

Most Virtual Private Network (VPN) sites require access to the Internet. RFC 4364 describes a set of models for enabling IPv4 and IPv6 VPN access to the Internet. In one model, one interface is used by the customer edge (CE) to connect to the Internet and a different one to connect to the virtual routing and forwarding (VRF) instance. Another model is in which all Internet routes are redistributed into the VRF; however, this approach has the disadvantage of requiring the Internet routes be replicated in each VRF.

In one scenario, a static route is inserted into the VRF table, with a next hop that points to the Internet gateway found in the IPv6 default table. The figure below illustrates this scenario, in which Internet access is provided to the customer in the VRF named `vrf1`.

Figure 5: Internet Access Topology



A customer site that has access public resources over the Internet must be known by a public prefix. Unlike IPv4, IPv6 does not offer a Network Address Translation (NAT) mechanism that translates private addresses into public addresses when leaving the site boundaries. This implies that hosts within the site speak with public addresses and appear in the public domain.

For outbound traffic, the default route configured in the VRF table at ingress provider edge (PE1) directs traffic for destinations outside the VPN to the Internet gateway.

For inbound traffic, a route must exist at the Internet gateway to direct the traffic for a customer site via its PE of attachment (PE1 in the figure above). This route can be distributed by the ingress PE (PE1) using multiprotocol internal Border Gateway Protocol (iBGP) (with the IPv6 address family configuration), so no specific configuration is needed on a per-VPN PE basis at the Internet gateway. Nevertheless, for inbound traffic at PE1, a route must exist in the default table for the customer site global prefix pointing to the VRF of the site.

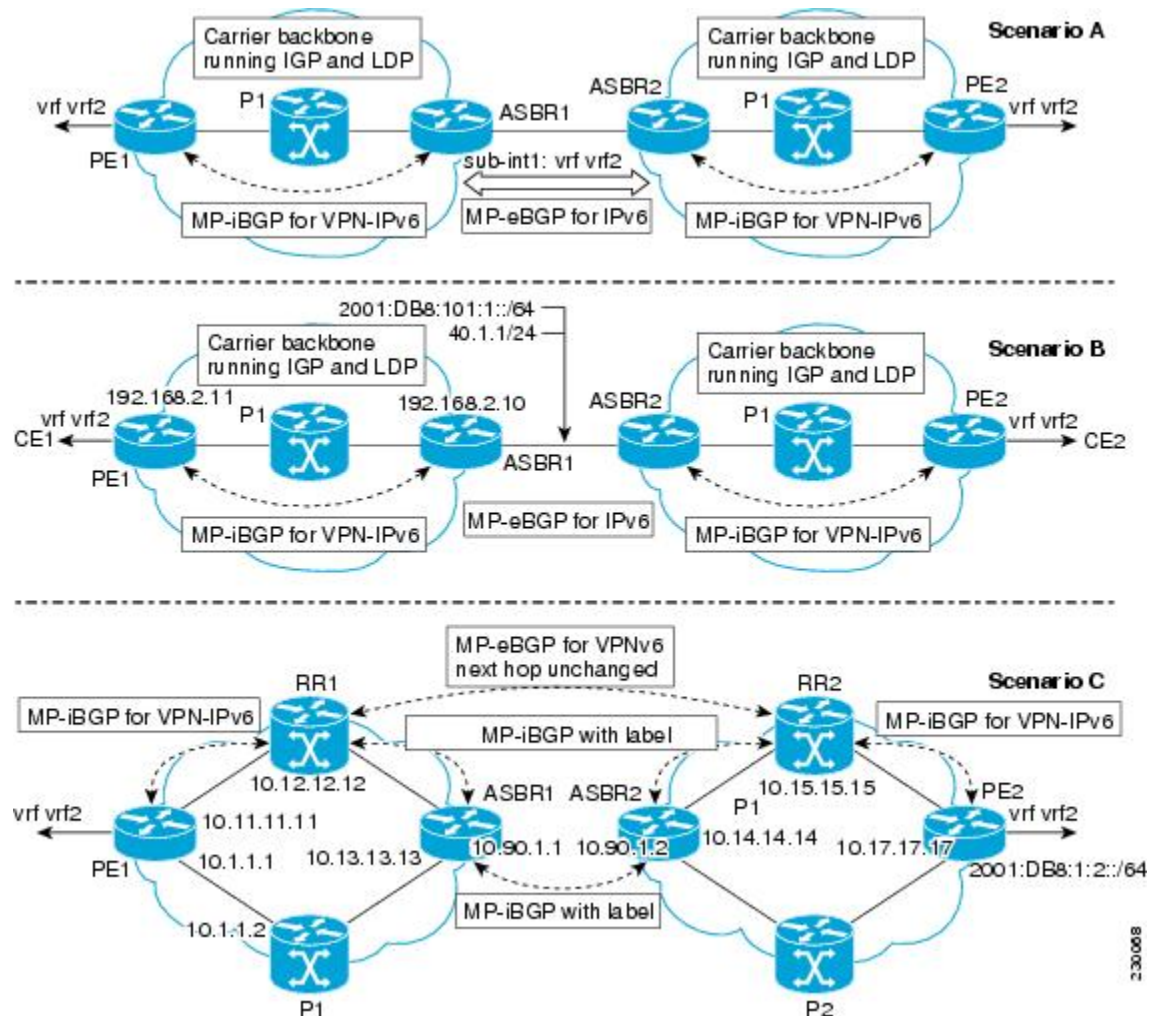
Multiautonomous-System Backbones

The problem of interprovider Virtual Private Networks (VPNs) is similar for IPv6 and IPv4, assuming that IPv6 was deployed everywhere IPv4 was deployed.

In IPv6 deployments that cross autonomous system boundaries, providers may have to obtain a peering model, or work with the peering model put in place for VPNv4.

The figure below illustrates interprovider scenarios in IPv6 VPN.

Figure 6: Interprovider Scenarios



Depending on the network protocol used between Autonomous System Boundary Routers (ASBRs), the three scenarios shown in the figure above can have several implementation options. For instance, scenario B, which suggests a multiprotocol external Border Gateway Protocol (eBGP) IPv6 VPN peering between ASBRs, could use either an IPv6 or an IPv4 link.

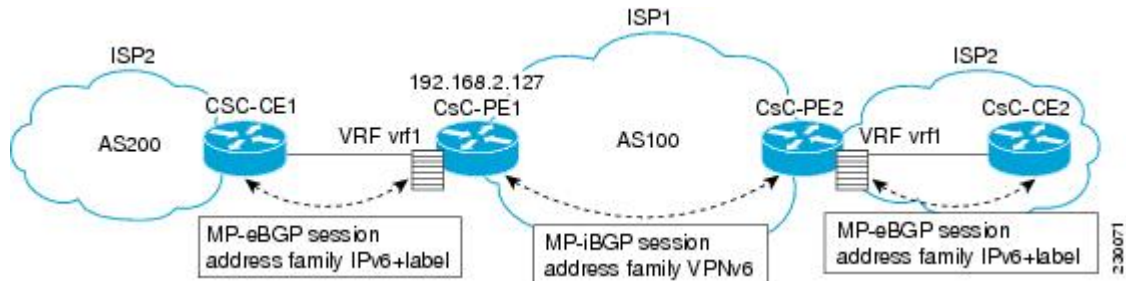
In scenario C, multihop multiprotocol eBGP redistributes IPv6 VPN routes across route reflectors in different autonomous systems. Labeled IPv4 routes to the provider edge (PE) devices (in the IPv6 over MPLS case) need to be advertised across ASBRs so that a complete labeled switch path is set up end to end.

Carrier Supporting Carriers

The Carrier Supporting Carrier (CSC) feature provides Virtual Private Network (VPN) access to a customer service provider, so this service needs to exchange routes and send traffic over the Internet service provider (ISP) Multiprotocol Label Switching (MPLS) backbone. The only difference from a regular provider edge (PE) is that it provides MPLS-to-MPLS forwarding on the CSC-customer edge (CE) to CSC-PE interface, rather than IP-to-MPLS forwarding.

The figure below highlights the two ISPs' interface.

Figure 7: CSC IPv6 over MPLS Configuration Example



How to Configure IPv6 VPN over MPLS

Configuring a Virtual Routing and Forwarding Instance for IPv6

A virtual routing and forwarding (VRF) instance is an address family-independent object that can be enabled and configured for each of the supported address families. Configuring a VRF consists of the following three steps:

- Configuring the address-family-independent part of the VRF
- Enabling and configuring IPv4 for the VRF
- Enabling and configuring IPv6 for the VRF

A VRF is given a name and a route distinguisher (RD). The RD is configured outside the context of the address family, although the RD is used to distinguish overlapping addresses within the context of a particular Border Gateway Protocol (BGP) address family. Having separate RDs for IPv4 VPN addresses and IPv6 VPN addresses does not matter. On Cisco devices, the RDs are the same in order to simplify configuration and VPN management.

Users can configure policies in common between IPv4 and IPv6 when not using an address family context. This feature is shared route targets (import and export), and it is useful in a migration scenario, where IPv4 policies already are configured and IPv6 policies should be the same as the IPv4 policies.

The IPv4 and IPv6 address family can each be enabled and configured separately. Note that the route-target policies entered at this level override global policies that may have been specified during address family-independent configuration.

SUMMARY STEPS

1. `enable`

2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **exit**
7. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
8. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
9. **exit**
10. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
11. **route-target** {**import** | **export** | **both**} *route-target-ext-community*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Device(config)# vrf definition vrf1	Configures a VPN VRF routing table and enters VRF configuration mode.
Step 4	rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 100:1	Specifies the RD for a VRF.
Step 5	route-target { import export both } <i>route-target-ext-community</i> Example: Device(config-vrf)# route target import 100:10	Specifies the route target VPN extended communities for both IPv4 and IPv6.
Step 6	exit Example: Device(config-vrf)# exit	Exits VRF configuration mode.

	Command or Action	Purpose
Step 7	address-family ipv4 [<i>mdt</i> <i>multicast</i> <i>tunnel</i> <i>unicast</i>] [<i>vrf vrf-name</i>] <i>vrf vrf-name</i> Example: Device(config)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 8	route-target { <i>import</i> <i>export</i> <i>both</i> } <i>route-target-ext-community</i> Example: Device(config-vrf-af)# route target import 100:11	Specifies the route target VPN extended communities specific to IPv4.
Step 9	exit Example: Device(config-vrf-af)# exit	Exits address family configuration mode on this VRF.
Step 10	address-family ipv6 [<i>vrf vrf-name</i>] [<i>unicast</i> <i>multicast</i>] Example: Device(config-vrf)# address-family ipv6	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 11	route-target { <i>import</i> <i>export</i> <i>both</i> } <i>route-target-ext-community</i> Example: Device(config-vrf-af)# route target import 100:12	Specifies the route target VPN extended communities specific to IPv6.

Binding a VRF to an Interface

In order to specify which interface belongs to which virtual routing and forwarding (VRF) instance, use the **vrf forwarding** command for both IPv4 and IPv6. An interface cannot belong to more than one VRF. When the interface is bound to a VRF, previously configured addresses (IPv4 and IPv6) are removed, and they must be reconfigured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [*secondary*]
6. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits/prefix-length*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 0/0/0	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	vrf forwarding <i>vrf-name</i> Example: Device(config-if)# vrf forwarding vrf1	Associates a VPN VRF with an interface or subinterface. <ul style="list-style-type: none"> • Note that any address, IPv4 or IPv6, that was configured prior to entering this command will be removed.
Step 5	ip address <i>ip-address mask</i> [secondary] Example: Device(config-if)# ip address 10.10.10.1 255.255.255.0	Configures an IPv4 address on the interface.
Step 6	ipv6 address <i>{ipv6-address / prefix-length prefix-name sub-bits/prefix-length}</i> Example: Device(config-if)# ipv6 address 2001:DB8:100:1::1/64	Configures an IPv6 address on the interface.

Configuring a Static Route for PE-to-CE Routing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix / prefix-length* *{ipv6-address | interface-type interface-number [ipv6-address]}* [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 route [vrf vrf-name] ipv6-prefix / prefix-length {ipv6-address interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1 default]] [administrative-distance] [administrative-multicast-distance unicast multicast] [next-hop-address] [tag tag] Example: Device(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:200::1 nexthop-vrf default	Installs the specified IPv6 static route using the specified next hop.

Configuring eBGP PE-to-CE Routing Sessions

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv6 [vrf vrf-name] [unicast | multicast]
5. neighbor {ip-address | ipv6-address | peer-group-name} remote-as as-number
6. neighbor {ip-address | ipv6-address | peer-group-name} activate

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast] Example: Device(config-router)# address-family ipv6 vrf vrf1	Enters address family configuration mode.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router-af)# neighbor 2001:DB8:100:1::2 remote-as 200	Adds an entry to the multiprotocol BGP neighbor table.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 2001:DB8:100:1::2 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

Configuring the IPv6 VPN Address Family for iBGP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **extended**] **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.11 remote-as 100	Adds an entry to the multiprotocol Border Gateway Protocol (BGP) neighbor table. <ul style="list-style-type: none"> In IPv6 VPN, the peer address typically is an IPv4 address, in order to enable the BGP session to be transported over the IPv4-based core network.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.11 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family vpnv6 [unicast] Example: Device(config-router)# address-family vpnv6	Places the device in address family configuration mode for configuring routing sessions.
Step 7	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.11 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Device(config-router-af)# neighbor 192.168.2.11 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 9	extended] exit Example: Device(config-router-af)# exit	Exits address family configuration mode.

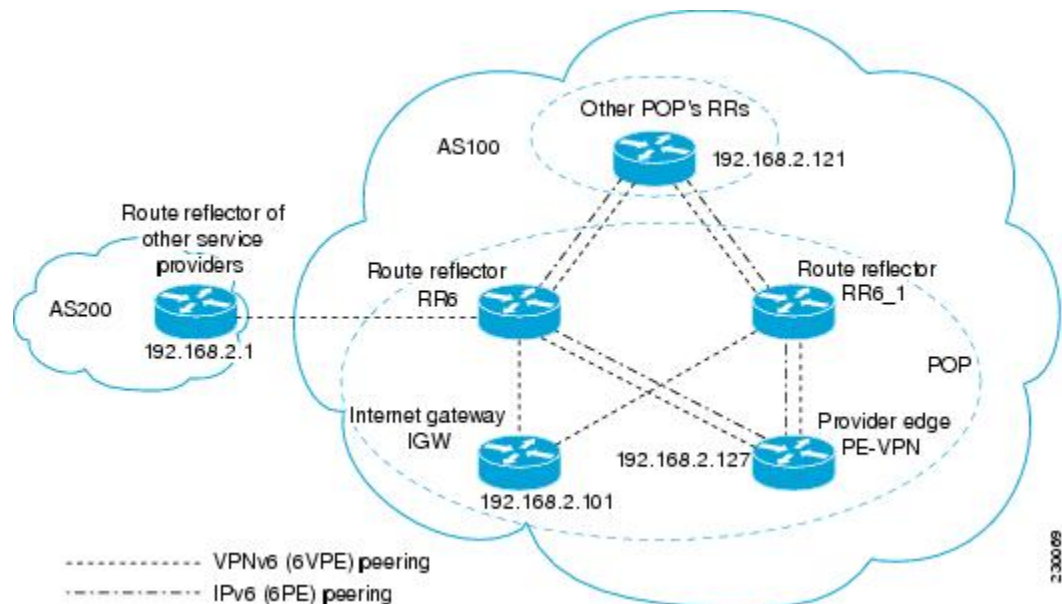
Configuring Route Reflectors for Improved Scalability

In this task, two route reflectors (RRs) are configured for redundancy reasons. Deploying RRs improves scalability by drastically reducing the number of Border Gateway Protocol (BGP) sessions. One RR usually peers with many internal Border Gateway Protocol (iBGP) speakers, preventing a full mesh of BGP sessions.

In a Multiprotocol Label Switching (MPLS)-based core, RRs are not part of the label switch paths and can be located anywhere in the network. For example, in a flat RR design, RRs can be deployed at Level 1 points of presence (POPs) and peer together in a full-mesh topology. In a hierarchical RR design, RRs could be deployed at Level 1 and Level 2 POPs, with Level 1 POPs peering together and with Level 2 RRs.

In a typical case where IPv6 over MPLS (6VPE) is deployed in a preexisting MPLS network (for example, providing VPNv4 services), it is likely that some RR design is already in place, and a similar RR infrastructure for IPv6 Virtual Private Network (VPN) services can be deployed. The figure below illustrates the main peering points between the RR in the ISP POP and the set of its RR clients.

Figure 8: Route Reflector Peering Design



The following list of BGP RR clients must be configured at each IPv6 RR (RR6 and RR6_1 in the figure above) device, at each POP:

- Provider edge (PE) devices (PE-VPN) of the POP providing IPv6 VPN access to the ISP customers. This includes both IPv6 VPN (6VPE) peering for interconnecting customer sites and IPv6 peering (6PE) for providing Internet access to VPN customers (see the “Configuring Internet Access” section).
- Internet gateway (IGW) located in the POP in order to provide PE customers with access to the IPv6 Internet (see the “Configuring Internet Access” section).
- RRs from other service providers. This feature is used to provide interautonomous-system connectivity, and it includes both IPv6 and IPv6 VPN peering. This service is described in the “Configuring a Multiautonomous-System Backbone for IPv6 VPN” section.
- RRs in other POPs. All RRs peer together, with both IPv6 and IPv6 VPN address families enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
10. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
11. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*tll*]
13. **address-family ipv6**
14. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
15. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
16. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
17. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
18. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
19. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
20. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
21. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
22. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
23. **exit**
24. **address-family vpv6** [**unicast**]
25. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
26. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
27. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
28. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
29. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
30. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
31. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
32. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
33. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
34. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.101 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the Internet gateway in order to provide Internet access.
Step 5	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.101 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.121 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the other POP's RR.
Step 7	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.121 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 8	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.127 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table.
Step 9	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} update-source <i>interface-type interface-number</i> Example:	Enables the BGP session to use a source address on the specified interface.

	Command or Action	Purpose
	Device(config-router)# neighbor 192.168.2.127 update-source Loopback 0	
Step 10	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.1 remote-as 200	(Optional) Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the RR of the peer ISP in order to provide inter-VPN service.
Step 11	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.1 update-source Loopback 0	(Optional) Enables the BGP session to use a source address on the specified interface.
Step 12	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } ebgp-multihop [<i>tth</i>] Example: Device(config-router)# neighbor 192.168.2.1 ebgp-multihop	(Optional) Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 13	address-family ipv6 Example: Device(config-router)# address-family ipv6	(Optional) Enters address family configuration mode in order to provide Internet access service.
Step 14	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.101 activate	(Optional) Enables the exchange of information for this address family with the specified neighbor.
Step 15	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Device(config-router-af)# neighbor 192.168.2.101 send-label	(Optional) Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device.
Step 16	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Device(config-router-af)# neighbor 192.168.2.101 route-reflector-client	(Optional) Configures the device as a BGP route reflector and configures the specified neighbor as its client.

	Command or Action	Purpose
Step 17	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: <pre>Device(config-router-af)# neighbor 192.168.2.121 activate</pre>	(Optional) Enables the exchange of information for this address family with the specified BGP neighbor.
Step 18	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: <pre>Device(config-router-af)# neighbor 192.168.2.121 send-label</pre>	(Optional) Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device.
Step 19	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: <pre>Device(config-router-af)# neighbor 192.168.2.121 route-reflector-client</pre>	(Optional) Configures the specified neighbor as a route reflector client.
Step 20	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: <pre>Device(config-router-af)# neighbor 192.168.2.127 activate</pre>	(Optional) Enables the exchange of information for this address family with the specified BGP neighbor.
Step 21	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: <pre>Device(config-router-af)# neighbor 192.168.2.127 send-label</pre>	(Optional) Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device.
Step 22	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: <pre>Device(config-router-af)# neighbor 192.168.2.127 route-reflector-client</pre>	(Optional) Configures the specified neighbor as a route reflector client.
Step 23	exit Example: <pre>Device(config-router-af)# exit</pre>	(Optional) Exits address family configuration mode.
Step 24	address-family vpnv6 [unicast] Example:	Places the device in address family configuration mode for configuring routing sessions.

	Command or Action	Purpose
	Device(config-router)# address-family vpnv6	
Step 25	neighbor {ip-address peer-group-name ipv6-address} activate Example: Device(config-router-af)# neighbor 192.168.2.121 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 26	neighbor {ip-address ipv6-address peer-group-name} send-community [both standard extended] Example: Device(config-router-af)# neighbor 192.168.2.21 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 27	neighbor {ip-address ipv6-address peer-group-name} route-reflector-client Example: Device(config-router-af)# neighbor 192.168.2.121 route-reflector-client	Configures the specified neighbor as a route reflector client.
Step 28	neighbor {ip-address peer-group-name ipv6-address} activate Example: Device(config-router-af)# neighbor 192.168.2.127 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 29	neighbor {ip-address ipv6-address peer-group-name} send-community [both standard extended] Example: Device(config-router-af)# neighbor 192.168.2.127 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 30	neighbor {ip-address ipv6-address peer-group-name} route-reflector-client Example: Device(config-router-af)# neighbor 192.168.2.127 route-reflector-client	Configures the specified neighbor as a route reflector client.
Step 31	neighbor {ip-address peer-group-name ipv6-address} activate Example:	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
	Device(config-router-af)# neighbor 192.168.2.1 activate	
Step 32	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Device(config-router-af)# neighbor 192.168.2.1 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 33	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Device(config-router-af)# neighbor 192.168.2.1 route-reflector-client	Configures the specified neighbor as a route reflector client.
Step 34	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } next-hop-unchanged [allpaths] Example: Device(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths	Enables an EBGP multihop peer to propagate to the next hop unchanged for paths.

Configuring Internet Access

Customers with IPv6 Virtual Private Network (VPN) access need to have access to the Internet through IPv6. The design of this service is similar to a global Internet access service. IPv6 VPN over MPLS (6VPE) devices located in a Level 1 point of presence (POP) (colocated with an IGW device) can access the Internet gateway (IGW) natively, whereas 6VPE devices located in Level 2 and Level 3 POPs with no direct access to the IGW can access the IGW in their closest Level 1 POP over 6PE.

Configuring VPN Internet access in such a 6VPE device involves configuring Border Gateway Protocol (BGP) peering with the IGW (in most cases through the IPv6 RR, as described in the “Configuring Route Reflectors for Improved Scalability” section). Then the user must configure cross-table routing to enable communication between the private domain (the VRF) and the public domain (the Internet).

The figure above illustrates the following configuration tasks:

Configuring the Internet Gateway

Configuring iBGP 6PE Peering to the VPN PE

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*

5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv6**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.127 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table to provide peering with the Virtual Private Network (VPN) provider edge (PE).
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.127 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family ipv6 Example: Device(config-router)# address-family ipv6	Enters address family configuration mode in order to exchange global table reachability.
Step 7	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.127 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: <pre>Device(config-router-af)# neighbor 192.168.2.127 send-label</pre>	Enables a BGP device to send Multiprotocol Label Switching (MPLS) labels with BGP routes to a neighboring BGP device, and allows the PE VPN to reach the Internet gateway over MPLS.

Configuring the Internet Gateway as the Gateway to the Public Domain

Use the 6PE peering configuration established in the “Configuring iBGP 6PE Peering to the VPN PE” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv6**
5. **network** *ipv6-address/prefix-length*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: <pre>Device(config)# router bgp 100</pre>	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	address-family ipv6 Example: <pre>Device(config-router)# address-family ipv6</pre>	Enters address family configuration mode in order to exchange global table reachability.
Step 5	network <i>ipv6-address/prefix-length</i> Example:	Configures the network source of the next hop to be used by the provider edge (PE) Virtual Private Network (VPN).

	Command or Action	Purpose
	Device(config-router-af) # network 2001:DB8:100::1/128	
Step 6	exit Example: Device(config-router-af) # exit	Exits address family configuration mode.

Configuring eBGP Peering to the Internet

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv6**
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
7. **aggregate-address** *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor FE80::300::1 GigabitEthernet0/0/0 remote-as 300	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with PE (PE-VPN). <ul style="list-style-type: none"> • Note that the peering is done over link-local addresses.

	Command or Action	Purpose
Step 5	address-family ipv6 Example: <pre>Device(config-router)# address-family ipv6</pre>	Enters address family configuration mode in order to exchange global table reachability.
Step 6	neighbor {ip-address ipv6-address peer-group-name} activate Example: <pre>Device(config-router-af)# neighbor FE80::300::1 GigabitEthernet0/0/0 activate</pre>	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 7	aggregate-address address mask [as-set] [summary-only] [suppress-map map-name] [advertise-map map-name] [attribute-map map-name] Example: <pre>Device(config-router-af)# aggregate-address 2001:DB8::/32 summary-only</pre>	Creates an aggregate prefix before advertising it to the Internet.

Configuring the IPv6 VPN PE

Configuring a Default Static Route from the VRF to the Internet Gateway

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number [ipv6-address]}** [**next-hop-vrf [vrf-name1 | default]**] [**administrative-distance**] [**administrative-multicast-distance** | **unicast** | **multicast**] [**next-hop-address**] [**tag tag**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.

Configuring a Static Route from the Default Table to the VRF

	Command or Action	Purpose
Step 3	<p>ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1 default]] [administrative-distance] [administrative-multicast-distance unicast multicast] [next-hop-address] [tag tag]</p> <p>Example:</p> <pre>Device(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:100::1 nexthop-vrf default</pre>	Configures a default static route from the VRF to the Internet gateway to allow outbound traffic to leave the VRF.

Configuring a Static Route from the Default Table to the VRF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1 | default]] [administrative-distance] [administrative-multicast-distance | unicast | multicast] [next-hop-address] [tag tag]

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1 default]] [administrative-distance] [administrative-multicast-distance unicast multicast] [next-hop-address] [tag tag]</p> <p>Example:</p> <pre>Device(config)# ipv6 route 2001:DB8:100:2000::/64 nexthop-vrf vrf1</pre>	Configures a static route from the default table to the VRF to allow inbound traffic to reach the VRF.

Configuring iBGP 6PE Peering to the Internet Gateway

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. **network** *ipv6-address/prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.101 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the Internet gateway.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.101 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast] Example:	Enters address family configuration mode to exchange global table reachability.

	Command or Action	Purpose
	Device(config-router)# address-family ipv6	
Step 7	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.101 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Device(config-router-af)# neighbor 192.168.2.101 send-label	Enables label exchange for this address family to this neighbor to enable the Virtual Private Network (VPN) provider edge (PE) to reach the Internet gateway over Multiprotocol Label Switching (MPLS).
Step 9	network <i>ipv6-address/prefix-length</i> Example: Device(config-router-af)# network 2001:DB8:100:2000::/64	Provides the virtual routing and forwarding (VRF) prefix to the Internet gateway.

Configuring a Multiautonomous-System Backbone for IPv6 VPN

Two Virtual Private Network (VPN) sites may be connected to different autonomous systems because the sites are connected to different service providers. The provider edge (PE) devices attached to that VPN is then unable to maintain the internal Border Gateway Protocol (iBGP) connections with each other or with a common route reflector. In this situation, there must be some way to use external BGP (eBGP) to distribute VPN-IPv6 addresses.

The following configuration example illustrates two scenarios, one in which a multiprotocol eBGP-IPv6 VPN peering between autonomous system boundary routers (ASBRs) uses an IPv4 link, and the same scenario using an IPv6 link. If the peering between ASBRs is performed over an IPv4 link, the BGP configuration on ASBR1 is as follows:

```
router bgp 1001
no bgp default ipv4-unicast
no bgp default route-target filter
neighbor 192.1.1.1 remote-as 1002
neighbor 192.168.2.11 remote-as 1001
neighbor 192.168.2.11 update-source Loopback1
!
address-family vpnv6
!Peering to ASBR2 over an IPv4 link
neighbor 192.1.1.1 activate
neighbor 192.1.1.1 send-community extended
!Peering to PE1 over an IPv4 link
neighbor 192.168.2.11 activate
neighbor 192.168.2.11 next-hop-self
neighbor 192.168.2.11 send-community extended
```

If the peering between ASBRs is performed over an IPv6 link, the BGP configuration on ASBR1 is as follows:


```
router bgp 1001
neighbor 2001:DB8:101::72d remote-as 1002
!
address-family vpnv6
!Peering to ASBR2 over an IPv6 link
neighbor 2001:DB8:101::72d activate
neighbor 2001:DB8:101::72d send-community extended
```

The next several tasks describe how to configure the PE VPN for a multiautonomous-system backbone using multihop multiprotocol eBGP to redistribute VPN routes across route reflectors (RRs) in different autonomous systems. Labeled IPv4 routes to the PEs are advertised across ASBRs so that a complete label switch path (LSP) is set up end to end.

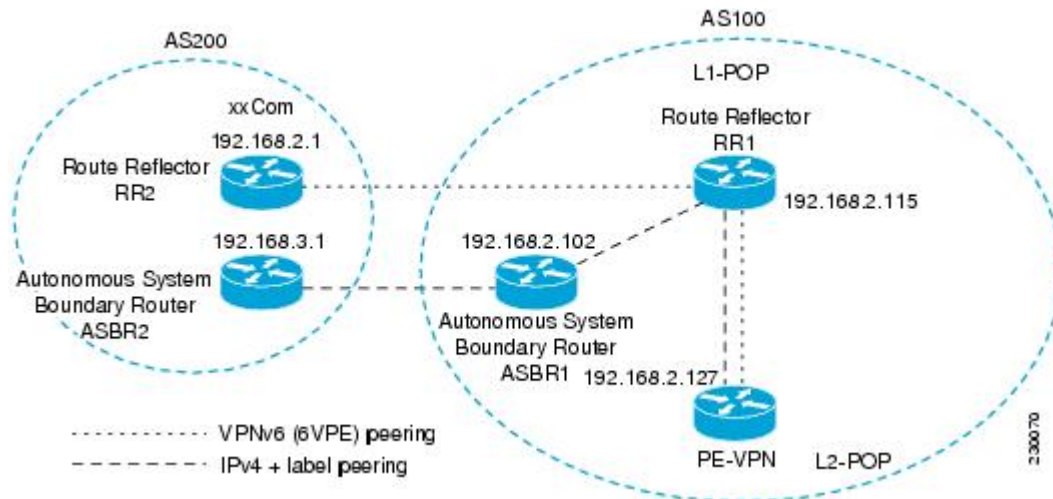
In this scenario, the ASBRs are not VPN aware; only the RRs are VPN aware. The following configuration should be available and understood:

- The ASBRs are providing the PEs' loopback addresses to service providers they peer with. That includes:
 - The VPN PE's IPv4 loopback address (/32) for enabling next-hop resolution at the remote service provider location.
 - The VPN RR's IPv4 loopback address (/32) for enabling interprovider (inter-RR) eBGP peering.
- For the VPN PE's IPv4 loopback address, the address providing is performed over multiprotocol BGP, with the label, up to the remote PEs, so that the label establishes an end-to-end LSP. Therefore, the following MP-BGP peering was set up for VPNv4:
 - VPN PEs are iBGP peering with VPN RRs.
 - ASBRs are iBGP peering with VPN RRs.
 - ASBRs are eBGP peering with the remote service provider ASBR.
- The VPN RRs of each service provider are peering together over eBGP and exchanging VPN routes. The next hop is forwarded unchanged, so that the end-to-end LSP is not via RRs.

To enable IPv6 VPN interautonomous-system access in this scenario, the ISP needs to modify the configurations at the PE VPN and at the RR. The same RRs are set up to provide a similar service for VPNv4. In that context, because the peering between the RR and the ASBR and between ASBRs is solely to exchange labels for IPv4 next hops used by both IPv4 VPN and IPv6 VPN, the ASBRs remain completely IPv6 unaware, and no configuration change is required there.

The figure below shows the BGP peering points required to enable IPv6 interprovider connectivity from the PE-VPN device (providing IPv6 VPN access) to the xxCom network.

Figure 9: BGP Peering Points for Enabling Interautonomous System Scenario C



The following additional BGP peerings are necessary to enable interautonomous-system communication from the IPv6 VPN PE located in the Level 2 point of presence (POP):

- IPv4 with label peering from the PE VPN to the route reflector named RR1 (which is already configured if VPNv4 interautonomous system is deployed on the same nodes, using the same LSP).
- IPv4 with label peering from RR1 to ASBR1.
- IPv4 with label peering between ASBR1 and ASBR2.
- IPv6 VPN peering between RR1 and RR2 (which is the route reflector in the other autonomous systems) to exchange IPv6 VPN routes.
- IPv6 VPN peering with RR1. If the same route reflectors used to scale the IPv6 VPN service are used for interautonomous-system capability, then this function might also be already configured (see the “Configuring Route Reflectors for Improved Scalability” section).

Configuring the multiautonomous-system backbone for IPv6 VPN consists of the following tasks:

Configuring the PE VPN for a Multiautonomous-System Backbone

Configuring iBGP IPv6 VPN Peering to a Route Reflector

Perform this task to configure internal Border Gateway Protocol (iBGP) IPv6 Virtual Private Network (VPN) peering to a route reflector named RR1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**

8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.115 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the route reflector with interautonomous-system functionality.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.115 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family vpv6 [unicast] Example: Device(config-router)# address-family vpv6	(Optional) Places the device in address family configuration mode for configuring routing sessions.
Step 7	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: <pre>Device(config-router-af) # neighbor 192.168.2.115 send-community extended</pre>	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 9	exit Example: <pre>Device(config-router-af) # exit</pre>	Exits address family configuration mode.

Configuring IPv4 and Label iBGP Peering to a Route Reflector

Perform this task to configure IPv4 and label internal Border Gateway Protocol (iBGP) peering to a route reflector named RRI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: <pre>Device(config)# router bgp 100</pre>	Configures the BGP routing process.
Step 4	address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] Example:	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.

	Command or Action	Purpose
	Device(config-router)# address-family ipv4	
Step 5	neighbor {ip-address ipv6-address peer-group-name} activate Example: Device(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 6	neighbor {ip-address ipv6-address peer-group-name} send-label Example: Device(config-router-af)# neighbor 192.168.2.115 send-label	Enables label exchange for this address family to this neighbor in order to receive remote provider edge (PE) peer IPv4 loopback with label via RR1 in order to set up an end-to-end label switch path (LSP).

Configuring the Route Reflector for a Multiautonomous-System Backbone

Configuring Peering to the PE VPN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {ip-address | ipv6-address | peer-group-name} **remote-as** *as-number*
5. **neighbor** {ip-address | ipv6-address | peer-group-name} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {ip-address | ipv6-address | peer-group-name} **activate**
8. **neighbor** {ip-address | ipv6-address | peer-group-name} **send-community** [**both** | **standard** | **extended**]
9. **exit**
10. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
11. **neighbor** {ip-address | ipv6-address | peer-group-name} **activate**
12. **neighbor** {ip-address | ipv6-address | peer-group-name} **send-label**
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.115 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the route reflector for InterAS.
Step 5	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.115 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family vpnv6 [unicast] Example: Device(config-router)# address-family vpnv6	(Optional) Places the device in address family configuration mode.
Step 7	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} activate Example: Device(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} send-community [both standard extended] Example: Device(config-router-af)# neighbor 192.168.2.115 send-community extended	Specifies that a community attribute should be sent to the BGP neighbor.
Step 9	exit Example: Device(config-router-af)# exit	Exits address family configuration mode.

	Command or Action	Purpose
Step 10	address-family ipv4 [mdt multicast tunnel unicast [vrf vrf-name] vrf vrf-name] Example: Device(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 11	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 12	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Device(config-router-af)# neighbor 192.168.2.115 send-label	Enables label exchange for this address family to this neighbor in order to send to the local provider edge (PE) the remote PE IPv4 loopback with a label in order to set up an end-to-end label switch path (LSP).
Step 13	exit Example: Device(config-router-af)# exit	Exits address family configuration mode.

Configuring the Route Reflector

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
10. **exit**
11. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf vrf-name**] | **vrf vrf-name**]
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
13. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.127 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the Virtual Private Network (VPN) provider edge (PE) for Interns.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.127 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family vpnv6 [unicast] Example: Device(config-router)# address-family vpnv6	(Optional) Places the device in address family configuration mode.
Step 7	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.127 activate	Enables the exchange of information for this address family with the specified neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Device(config-router-af)# neighbor 192.168.2.127 send-community extended	Specifies that a community attribute should be sent to the BGP neighbor.

	Command or Action	Purpose
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: <pre>Device(config-router-af)# neighbor 192.168.2.127 route-reflector-client</pre>	Configures the specified neighbor as a route reflector client.
Step 10	exit Example: <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode.
Step 11	address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] Example: <pre>Device(config-router)# address-family ipv4</pre>	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 12	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: <pre>Device(config-router-af)# neighbor 192.168.2.127 activate</pre>	Enables the exchange of information for this address family with the specified neighbor.
Step 13	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: <pre>Device(config-router-af)# neighbor 192.168.2.127 send-label</pre>	Enables label exchange for this address family to this neighbor in order to send to the local PE the remote PE IPv4 loopback with a label in order to set up an end-to-end LSP.
Step 14	exit Example: <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode.

Configuring Peering to the Autonomous System Boundary Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]

7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.102 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the ASBR1.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.102 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family ipv4 [mdt multicast tunnel unicast] [vrf <i>vrf-name</i>] vrf <i>vrf-name</i> Example: Device(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.102 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: <pre>Device(config-router-af)# neighbor 192.168.2.102 send-label</pre>	Enables label exchange for this address family to this neighbor in order to receive the remote PE IPv4 loopback with the label set to an end-to-end label switch path (LSP).

Configuring Peering to Another ISP Route Reflector

Perform this task to configure peering to an Internet service provider (ISP) route reflector named RR2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **address-family vpnv6** [**unicast**]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
10. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: <pre>Device(config)# router bgp 100</pre>	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example:	Adds an entry to the multiprotocol BGP neighbor table for external BGP (eBGP) peering with RR2.

	Command or Action	Purpose
	Device(config-router)# neighbor 192.168.2.1 remote-as 100	
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.1 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } ebgp-multihop [<i>ttl</i>] Example: Device(config-router)# neighbor 192.168.2.1 ebgp-multihop	(Optional) Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 7	address-family vpnv6 [unicast] Example: Device(config-router)# address-family vpnv6	(Optional) Places the device in address family configuration mode for configuring routing sessions.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.1 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Device(config-router-af)# neighbor 192.168.2.1 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 10	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } next-hop-unchanged [allpaths] Example: Device(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths	Enables an eBGP multihop peer to propagate to the next hop unchanged for paths.

Configuring the ASBR

Configuring Peering with Router Reflector RR1

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.2.115 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with RR1.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.2.115 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.

	Command or Action	Purpose
Step 6	address-family ipv4 [<i>mdt</i> <i>multicast</i> <i>tunnel</i> <i>unicast</i> [<i>vrf vrf-name</i>] <i>vrf vrf-name</i>] Example: Device(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Device(config-router-af)# neighbor 192.168.2.115 send-label	Enables label exchange for this address family to this neighbor in order to send to the local PE the remote PE IPv4 loopback with a label in order to set up an end-to-end label switch path (LSP).
Step 9	exit Example: Device(config-router-af)# exit	Exits address family configuration mode.

Configuring Peering with the Other ISP ASBR2

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **address-family ipv4** [*mdt* | *multicast* | *tunnel* | *unicast* [*vrf vrf-name*] | *vrf vrf-name*]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
10. **network** {*network-number* [*mask network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
11. **network** {*network-number* [*mask network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router)# neighbor 192.168.3.1 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the ASBR2.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 192.168.3.1 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } ebgp-multihop [<i>ttl</i>] Example: Device(config-router)# neighbor 192.168.3.1 ebgp-multihop	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 7	address-family ipv4 [mdt multicast tunnel unicast] [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] Example: Device(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor 192.168.3.1 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label	Enables label exchange for this address family to this neighbor in order to receive the remote provider edge (PE)

	Command or Action	Purpose
	Example: Device(config-router-af)# neighbor 192.168.3.1 send-label	IPv4 loopback with a label in order to set up an end-to-end label switch path (LSP).
Step 10	network { <i>network-number</i> [mask <i>network-mask</i>] <i>nsap-prefix</i> } [route-map <i>map-tag</i>] Example: Device(config-router-af)# network 192.168.2.27 mask 255.255.255.255	Flags a network as local to this autonomous system and enters the network to the BGP table. This configuration is for the PE VPN loopback.
Step 11	network { <i>network-number</i> [mask <i>network-mask</i>] <i>nsap-prefix</i> } [route-map <i>map-tag</i>] Example: Device(config-router-af)# network 192.168.2.15 mask 255.255.255.255	Flags a network as local to this autonomous system and enters the network to the BGP table. This configuration is for the RR1 loopback.

Configuring CSC for IPv6 VPN

Perform this task to configure CsC-PE1 peering configuration with CsC-CE1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **router bgp** *autonomous-system-number*
5. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	hostname <i>name</i> Example: Device(config)# hostname CSC-PE1	Specifies or modifies the host name for the network server.
Step 4	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the Border Gateway Protocol (BGP) routing process.
Step 5	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast] Example: Device(config-router)# address-family ipv6 vrf ISP2	Enters address family configuration mode.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Device(config-router-af)# neighbor FE80::866C:99 GigabitEthernet0/0/0 remote-as 200	Adds an entry to the multiprotocol BGP neighbor table.
Step 7	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate Example: Device(config-router-af)# neighbor FE80::866C:99 GigabitEthernet0/0/0 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Device(config-router-af)# neighbor FE80::866C:99 GigabitEthernet0/0/0 send-label	Enables label exchange for this address family to this neighbor.

Configuration Examples for IPv6 VPN over MPLS

Examples: IPv6 VPN over MPLS Routing

Example: BGP IPv6 Activity Summary

```
Device# show bgp ipv6 unicast summary

For address family: IPv6 Unicast
```

Example: Dumping the BGP IPv6 Tables

```

BGP router identifier 192.168.2.126, local AS number 33751
BGP table version is 15, main routing table version 15
12 network entries using 1692 bytes of memory
22 path entries using 1672 bytes of memory
5/4 BGP path/bestpath attribute entries using 580 bytes of memory
14 BGP rrinfo entries using 336 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4328 total bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 13/1 prefixes, 23/1 paths, scan interval 60 secs
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
192.168.2.146  4 33751   991    983     15   0    0 16:26:21      10
192.168.2.147  4 33751   991    983     15   0    0 16:26:22      10
FE80::4F6B:44 GigabitEthernet1/0/0
                  4 20331   982    987     15   0    0 14:55:52        1

```

Example: Dumping the BGP IPv6 Tables

Each table (for example, BGP IPv6, BGP IPv6 VPN) can be reviewed individually, as shown in the following example:

```

Device# show bgp ipv6 unicast
BGP table version is 15, local router ID is 192.168.2.126
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric      LocPrf  Weight  Path
* i2001:DB8:100::/48  ::FFFF:192.168.2.101    0        100      0 10000 ?
*>i                  ::FFFF:192.168.2.101    0        100      0 10000 ?
* i2001:DB8::1/128   ::FFFF:192.168.2.101    0        100      0  i
*>i                  ::FFFF:192.168.2.101    0        100      0  i

```

Example: Dumping the IPv6 Routing Tables

IPv6 routing tables identify each routing protocol contributor to routable entries, as shown in the following example:

```

Device# show ipv6 route
IPv6 Routing Table - default - 13 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
B 2001:DB8:100::/48 [200/0]
   via 192.168.2.101 Default-IP-Routing-Table, indirectly connected
B 2001:DB8::1/128 [200/0]
   via 192.168.2.101 Default-IP-Routing-Table, c
LC 2001:DB8::26/128 [0/0]
   via Loopback0, receive

```

From an IPv6 routing perspective, entries reachable over the MPLS backbone are listed as being indirectly connected, because MPLS is providing a Layer 2 tunnel mechanism.

Examples: IPv6 VPN over MPLS Forwarding

Example: PE-CE Connectivity

The **ipv6 ping** and **traceroute** commands are useful to check connectivity from a provider edge (PE) to a customer edge (CE), whether locally attached or remote over the Multiprotocol Label Switching (MPLS) backbone.

When a device is locally attached, one can use the **ipv6 ping** command with the CE link-local address (used for external BGP peering), as shown in the following example:

```
Device# ping FE80::4F6B:44%
Loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::4F6B:44, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/33/48 ms
```

The **ipv6 ping** command also can be used to test remote PE or CE reachability, but only IPv6 global addresses can be used (link-local addresses are not advertised beyond the link):

```
Device# ping 2001:DB8:1120:1::44
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:1120:1:44::1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/33/48 ms
```

Note that the **ping ipv6** and **traceroute** command functions over MPLS require PEs and CEs to announce one IPv6 global prefix. Each 6PE device announces 2001:DB8::PE#/128, filtered at the autonomous system edge. Each IPv6 CE configures 2001:DB8:prefix:CE#/128 and announces it as part as its less-specific prefix (2001:DB8:prefix::/n).

Reachability of remote PEs and CEs can be tested by using the **traceroute** command. If you have configured all PEs with the **no mpls ip propagate-ttl forwarded** command, when the **traceroute** command is executed from a CE, its output will show only the IPv6 nodes:

```
Device# traceroute 2001:DB8::1
Type escape sequence to abort.
Tracing the route to 2001:DB8::1
 1 2001:DB8::26 [AS 33751] 32 msec 32 msec 20 msec
 2 2001:DB8::1 [AS 33751] [MPLS: Label 73 Exp 0] 20 msec 20 msec 20 msec
 3 2001:DB8::1 [AS 33751] 28 msec 20 msec 20 msec
```

After the P devices have been upgraded with images that support ICMPv6, the **traceroute** command executed on the PE device (Time to Live [TTL] is then propagated) will also show P devices' responses, as shown in the following example:

```
Device# traceroute 2001:DB8::1
Type escape sequence to abort.
Tracing the route to 2001:DB8::1
 1 ::FFFF:172.20.25.1 [MPLS: Labels 38/73 Exp 0] 40 msec 32 msec 32 msec
 2 ::FFFF:172.20.10.1 [MPLS: Labels 30/73 Exp 0] 60 msec 32 msec 32 msec
 3 2001:DB8::1 [MPLS: Label 73 Exp 0] 32 msec 32 msec 16 msec
```

When run from a 6VPE device, both the **ping ipv6** and **traceroute** commands accept a *vrf* argument, exactly as in the case of VPNv4.

Note that the **traceroute** command is useful for evaluating the path across the MPLS backbone, but not for troubleshooting data-plane failures. The P devices are IPv6 unaware (and are also VPNv4 unaware), so the ICMPv6 messages that they generate in response to the **traceroute** command are forwarded to the egress PE using the received label stack. The egress PE can route the ICMPv6 message to the source of the traceroute. When the MPLS path is broken, it is also broken from the ICMP message, which cannot reach the egress PE.

Examples: PE Imposition Path

On Cisco devices, the most useful tool for troubleshooting the imposition path for IPv6 is the **show ipv6 cef** command.

You can use the **show ipv6 cef** command to display the IPv6 forwarding table with label stacks used for each destination prefix, as shown in the following example:

```
Device# show ipv6 cef

2001:DB8:100::/48
  nexthop 172.20.25.1 GigabitEthernet0/0/0 label 38 72
2001:DB8::1/128
  nexthop 172.20.25.1 GigabitEthernet0/0/0 label 38 73
2001:DB8::26/128
  attached to Loopback0, receive
```

You can use the **show ipv6 cef** command to display details for a specific IPv6 entry in the forwarding table and to analyze how the destination was resolved and the label stack computed, as shown in the following example:

```
Device# show ipv6 cef 2001:DB8:100::/48 internal

2001:DB8:100::/48, epoch 0, RIB[B], refcount 4
sources: RIB
..
  recursive via 192.168.2.101[IPv4:Default] label 72, fib 0252B1F8, 1 terminal fib
  path 024F56A8, path list 024F0BA8, share 0/1, type attached nexthop
  ifnums: (none)
  path_list contains at least one resolved destination(s). HW IPv4 notified.
  nexthop 172.20.25.1 GigabitEthernet0/0/0 label 38, adjacency IP adj out of
GigabitEthernet0/0/0 0289BEF0
  output chain: label 72 label 38 TAG adj out of GigabitEthernet0/0/0 0289BD80
```

The detailed output in the previous example shows that each label composing the label stack has a different origin that can be tracked down individually. The Border Gateway Protocol (BGP) table has the bottom label, as shown in the following example:

```
Device# show bgp ipv6 unicast 2001:DB8:100::/48

BGP routing table entry for 2001:DB8:100::/48, version 2
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
    1
  10000
    ::FFFF:192.168.2.101 (metric 30) from 192.168.2.147 (192.168.2.147)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Originator: 192.168.2.101, Cluster list: 192.168.2.147,
      mpls labels in/out nolabel/72
  10000
    ::FFFF:192.168.2.101 (metric 30) from 192.168.2.146 (192.168.2.146)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
```

```
Originator: 192.168.2.101, Cluster list: 192.168.2.146,
mpls labels in/out nlabel/72
```

Label Distribution Protocol (LDP), as shown in this example, displays the other labels:

```
Device# show mpls ldp bindings 192.168.2.101 32
lib entry: 192.168.2.101/32, rev 56
  local binding: label: 40
  remote binding: lsr: 192.168.2.119:0, label: 38
Device# show mpls ldp bindings 172.20.25.0 24
lib entry: 172.20.25.0/24, rev 2
  local binding: label: imp-null
  remote binding: lsr: 192.168.2.119:0, label: imp-null
```

Examples: PE Disposition Path

Use the following examples to troubleshoot the disposition path.

The following example shows the Multiprotocol Label Switching (MPLS) forwarding table information for troubleshooting the disposition path.

```
Device# show mpls forwarding-table
```

Local Label	Outgoing Label or VC	Prefix or Tunnel Id	Bytes Switched	Label	Outgoing interface	Next Hop
16	Pop Label	192.168.2.114/32	0		GE0/0/0	point2point
17	26	192.168.2.146/32	0		GE0/0/0	point2point
..						
72	No Label	2001:DB8:100::/48	63121		GE1/0/0	point2point
73	Aggregate	2001:DB8::1/128	24123			

The following example shows the label used for switching, which has been announced by iBGP (6PE in this example) and can be checked:

```
Device# show bgp ipv6 2001:DB8:100::/48
BGP routing table entry for 2001:DB8:100::/48, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    2
  10000
    FE80::2710:2 (FE80::2710:2) from FE80::2710:2 GigabitEthernet1/0/0 (192.168.2.103)
      Origin incomplete, metric 0, localpref 100, valid, external, best,
```

Examples: Label Switch Path

Because the 6PE and 6VPE label switch path (LSP) endpoints are IPv4 addresses, the IPv4 tools for troubleshooting LSPs are useful for detecting data-plane failures that would lead to IPv6 traffic null route.

The following example displays the LSP IPv4 end to analyze the LSP:

```
Device# show ipv6 route 2001:DB8::1/128
Routing entry for 2001:DB8::1/128
  Known via "bgp 33751", distance 200, metric 0, type internal
  Route count is 1/1, share count 0
  Routing paths:
    192.168.2.101%Default-IP-Routing-Table indirectly connected
```

```
MPLS Required
Last updated 02:42:12 ago
```

The following example shows the traceroute LSP:

```
Device# traceroute mpls ipv4 192.168.2.101/32 verbose

Tracing MPLS Label Switched Path to 192.168.2.101/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
        '.' - timeout, 'U' - unreachable,
        'R' - downstream router but not target,
        'M' - malformed request
Type escape sequence to abort.
 0 172.20.25.2 0.0.0.0 MRU 1500 [Labels: 38 Exp: 0]
R 1 172.20.25.1 0.0.0.0 MRU 1500 [Labels: 30 Exp: 0] 40 ms, ret code 6
R 2 172.20.10.1 0.0.0.0 MRU 1504 [Labels: implicit-null Exp: 0] 60 ms, ret code 6
! 3 172.20.40.1 48 ms
```

Examples: IPv6 VPN over MPLS VRF

Examples: VRF Information

The following entries show VRF information for 6VPE.

The following is sample output from a Cisco Express Forwarding FIB associated with a virtual routing and forwarding (VRF) instance named cisco1:

```
Device# show ipv6 cef vrf cisco1

2001:8::/64
  attached to GigabitEthernet0/0/1
2001:8::3/128
  receive
2002:8::/64
  nexthop 10.1.1.2 GigabitEthernet0/1/0 label 22 19
2010::/64
  nexthop 2001:8::1 GigabitEthernet0/0/1
2012::/64
  attached to Loopback1
2012::1/128
  receive
```

The following is sample output regarding an IPv6 routing table associated with a VRF named cisco1:

```
Device# show ipv6 route vrf cisco1

IPv6 Routing Table cisco1 - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
        U - Per-user Static route
        I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
        O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
C   2001:8::/64 [0/0]
    via ::, GigabitEthernet0/0/1
L   2001:8::3/128 [0/0]
    via ::, GigabitEthernet0/0/1
B   2002:8::/64 [200/0]
    via ::FFFF:192.168.1.4,
B   2010::/64 [20/1]
    via 2001:8::1,
C   2012::/64 [0/0]
```

```

    via ::, Loopback1
L   2012::1/128 [0/0]
    via ::, Loopback1

```

Example: IPv6 VPN Configuration Using IPv4 Next Hop

The following example illustrates a 6VPE next hop:

```

interface Loopback0
 ip address 192.168.2.11 255.255.255.255
!
router bgp 100
 neighbor 192.168.2.10 remote-as 100
 neighbor 192.168.2.10 update-source Loopback0
!
 address-family vpnv6
  neighbor 192.168.2.10 activate
  neighbor 192.168.2.10 send-community extended
 exit-address-family

```

By default, the next hop advertised will be the IPv6 Virtual Private Network (VPN) address:

```
[0:0]::FFFF:192.168.2.10
```

Note that it is a 192-bit address in the format of [RD]::FFFF:IPv4-address.

When the Border Gateway Protocol (BGP) IPv6 VPN peers share a common subnet, the MP_REACH_NLRI attribute contains a link-local address next hop in addition to the global address next hop. This situation typically occurs in an interautonomous-system topology when autonomous system boundary routers (ASBRs) are facing each other. In that case, the link-local next hop is used locally, and the global next hop is readvertised by BGP.

The BGP next hop is the keystone for building the label stack. The inner label is obtained from the BGP network layer reachability information (NLRI), and the outer label is the Label Distribution Protocol (LDP) label to reach the IPv4 address embedded into the BGP next hop.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide Library</i>
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	IPv6 Feature Mapping

Related Topic	Document Title
Configuring MPLS Layer 3 VPNs	"MPLS Virtual Private Networks" module in the <i>MPLS Layer 3 VPNs Configuration Guide</i>

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 VPN over MPLS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for IPv6 VPN over MPLS

Feature Name	Releases	Feature Information
IPv6 VPN over MPLS	Cisco IOS XE Release 3.18SP	This feature was introduced on the Cisco NCS 4200 Series.

Glossary

- **6VPE device**—Provider edge device providing BGP-MPLS IPv6 VPN service over an IPv4-based MPLS core. It is a IPv6 VPN PE, dual-stack device that implements 6PE concepts on the core-facing interfaces.
- **customer edge (CE) device**—A service provider device that connects to VPN customer sites.

- **Forwarding Information Base (FIB)**—Table containing the information necessary to forward IP datagrams. At a minimum, the FIB contains the interface identifier and next-hop information for each reachable destination network prefix.
- **inbound route filtering (IRF)**—A BGP capability used for filtering incoming BGP updates that are not to be imported by the receiving PE device.
- **IPv6 provider edge device (6PE device)**—Device running a BGP-based mechanism to interconnect IPv6 islands over an MPLS-enabled IPv4 cloud.
- **IPv6 VPN address**—A IPv6 VPN address is a 24-byte identifier, beginning with an 8-byte route distinguisher (RD) and ending with a 16-byte IPv6 address. Sometimes it is called an IPv6 VPN address.
- **IPv6 VPN address family**—The address-family identifier (AFI) identifies a particular network-layer protocol and the subsequent AFI (SAFI) provides additional information. The AFI IPv6 SAFI VPN (AFI=2, SAFI=128) is called the IPv6 VPN address family. Sometimes it is called the IPv6 VPN address family. Similarly AFI IPv4 SAFI VPN is the VPNv4 address family.
- **network layer reachability information (NLRI)**—BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and community values.
- **outbound route filtering (ORF)**—A BGP capability used to filtering outgoing BGP routing updates.
- **point of presence (POP)**—Physical location where an interexchange carrier installed equipment to interconnect with a local exchange carrier.
- **provider edge (PE) device**—A service provider device connected to VPN customer sites.
- **route distinguisher (RD)**—A 64-bit value prepended to an IPv6 prefix to create a globally unique IPv6 VPN address.
- **Routing Information Base (RIB)**—Also called the routing table.
- **Virtual routing and forwarding (VRF)**—A VPN routing and forwarding instance in a PE.
- **VRF table**—A routing and a forwarding table associated to a VRF. This is a customer-specific table that enables the PE device to maintain independent routing states for each customer.



CHAPTER 5

IPv6 Switching: Provider Edge Router over MPLS

Multiprotocol Label Switching (MPLS) is deployed by many service providers in their IPv4 networks. Service providers want to introduce IPv6 services to their customers, but changes to their existing IPv4 infrastructure can be expensive and the cost benefit for a small amount of IPv6 traffic does not make economic sense. Several integration scenarios have been developed to leverage an existing IPv4 MPLS infrastructure and add IPv6 services without requiring any changes to the network backbone. This document describes how to implement IPv6 over MPLS.

Effective Cisco IOS XE Release 3.18SP, the 6PE feature is supported on the Cisco ASR 900 RSP3 module.

- [Prerequisites for IPv6 Switching: Provider Edge Router over MPLS, on page 91](#)
- [Information About IPv6 Switching: Provider Edge Router over MPLS, on page 91](#)
- [How to Deploy IPv6 Switching: Provider Edge Router over MPLS, on page 93](#)
- [Configuration Examples for IPv6 Switching: Provider Edge Router over MPLS, on page 97](#)
- [Additional References for IPv6 Switching: Provider Edge Router over MPLS, on page 100](#)

Prerequisites for IPv6 Switching: Provider Edge Router over MPLS

Before the IPv6 Provider Edge Router over MPLS (6PE) feature can be implemented, MPLS must be running over the core IPv4 network. If Cisco devices are used, Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled for both IPv4 and IPv6 protocols. This module assumes that you are familiar with MPLS.

Information About IPv6 Switching: Provider Edge Router over MPLS

Benefits of Deploying IPv6 over MPLS Backbones

IPv6 over MPLS backbones enables isolated IPv6 domains to communicate with each other over an MPLS IPv4 core network. This implementation requires only a few backbone infrastructure upgrades and no reconfiguration of core devices because forwarding is based on labels rather than the IP header itself, providing a very cost-effective strategy for the deployment of IPv6.

Additionally, the inherent Virtual Private Network (VPN) and MPLS traffic engineering (MPLS-TE) services available within an MPLS environment allow IPv6 networks to be combined into IPv4 VPNs or extranets over an infrastructure supporting IPv4 VPNs and MPLS-TE.

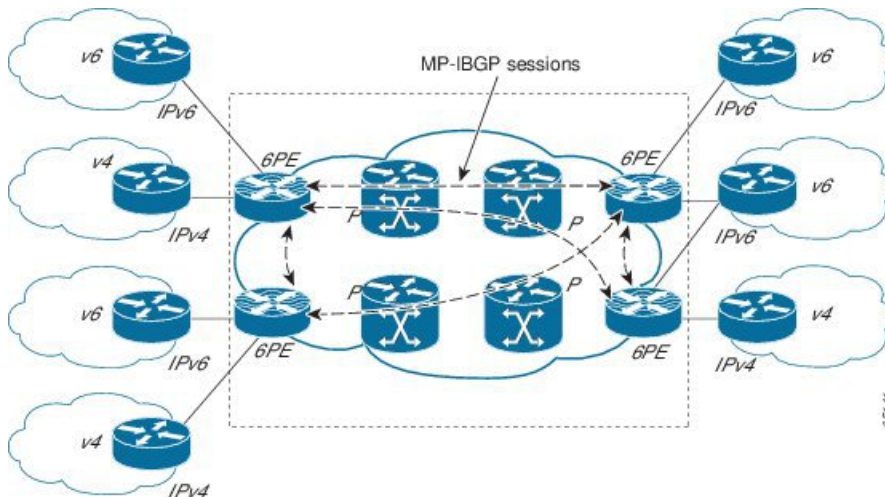
IPv6 on the Provider Edge Devices

The Cisco implementation of IPv6 Provider Edge Router over MPLS is called 6PE, and it enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS label switched paths (LSPs). This feature relies on multiprotocol Border Gateway Protocol (BGP) extensions in the IPv4 network configuration on the provider edge (PE) device to exchange IPv6 reachability information in addition to an MPLS label for each IPv6 address prefix to be advertised. Edge devices are configured to be dual stack running both IPv4 and IPv6, and use the IPv4 mapped IPv6 address for IPv6 prefix reachability exchange.

A hierarchy of labels is imposed on the 6PE ingress device to keep the IPv6 traffic transparent to all the core devices. The top label provides connectivity inside the IPv4 MPLS core network and the label is distributed by Label Distribution Protocol (LDP), Tag Distribution Protocol (TDP), or Resource Reservation Protocol (RSVP). TDP and LDP can both be used for label distribution, but RSVP is used only in the context of MPLS-TE label exchange. The bottom label, automatically assigned to the IPv6 prefix of the destination, is distributed by multiprotocol BGP and used at each 6PE egress device for IPv6 forwarding.

In the figure below the 6PE devices are configured as dual stack devices able to route both IPv4 and IPv6 traffic. Each 6PE device is configured to run LDP, TDP, or RSVP (if traffic engineering is configured) to bind the IPv4 labels. The 6PE devices use multiprotocol BGP to exchange reachability information with the other 6PE devices within the MPLS domain, and to distribute IPv6 labels between them. All 6PE and core devices--P devices in Figure 3--within the MPLS domain share a common IPv4 Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) or Integrated Intermediate System-to-Intermediate System (IS-IS).

Figure 10: 6PE Device Topology



The interfaces on the 6PE devices connecting to the CE device can be configured to forward IPv6 traffic, IPv4 traffic, or both types of traffic depending on the customer requirements. 6PE devices advertise IPv6 reachability information learned from their 6PE peers over the MPLS cloud. Service providers can delegate an IPv6 prefix from their registered IPv6 prefixes over the 6PE infrastructure; otherwise, there is no impact on the CE device.

The P devices in the core of the network are not aware that they are switching IPv6 packets. Core devices are configured to support MPLS and the same IPv4 IGP as the PE devices to establish internal reachability inside

the MPLS cloud. Core devices also use LDP, TDP, or RSVP for binding IPv4 labels. Implementing the Cisco 6PE feature does not have any impact on the MPLS core devices.

Within the MPLS network, IPv6 traffic is forwarded using label switching, making the IPv6 traffic transparent to the core of the MPLS network. No IPv6 over IPv4 tunnels or Layer 2 encapsulation methods are required.

How to Deploy IPv6 Switching: Provider Edge Router over MPLS

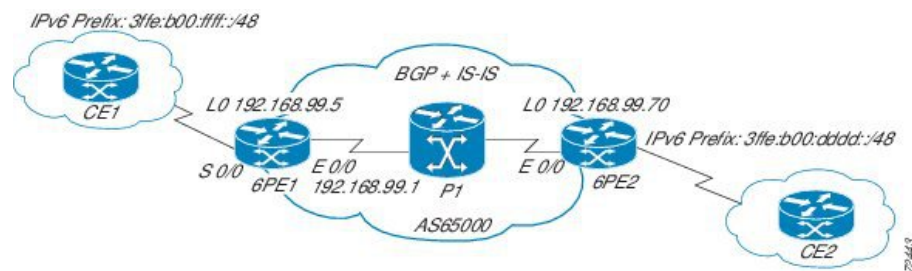
Deploying IPv6 on the Provider Edge Devices (6PE)

Specifying the Source Address Interface on a 6PE Device

Two configuration tasks using the network shown in the figure below are required at the 6PE1 device to enable the 6PE feature.

The customer edge device--CE1 in the figure below--is configured to forward its IPv6 traffic to the 6PE1 device. The P1 device in the core of the network is assumed to be running MPLS, a label distribution protocol, an IPv4 IGP, and Cisco Express Forwarding or distributed Cisco Express Forwarding, and does not require any new configuration to enable the 6PE feature.

Figure 11: 6PE Configuration Example



Before you begin

- The 6PE devices--the 6PE1 and 6PE2 devices in the figure below--must be members of the core IPv4 network. The 6PE device interfaces attached to the core network must be running MPLS, the same label distribution protocol, and the same IPv4 IGP, as in the core network.
- The 6PE devices must also be configured to be dual stack to run both IPv4 and IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **ipv6 cef distributed**
5. **interface** *type number*
6. **ipv6 address** *ipv6-address / prefix-length | prefix-name sub-bits / prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Device(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 4	ipv6 cef distributed Example: Device(config)# ipv6 cef distributed	Enables IPv6 Cisco Express Forwarding.
Step 5	interface <i>type number</i> Example: Device(config)# interface Serial 0/0GigabitEthernet 0/0/0	Specifies an interface type and number and enters interface configuration mode. <ul style="list-style-type: none">• In the context of this feature, the interface to be configured is the interface communicating with the CE device.
Step 6	ipv6 address <i>ipv6-address / prefix-length prefix-name sub-bits / prefix-length</i> Example: Device(config-if)# ipv6 address 2001:DB8:FFFF::2/64	Configures an IPv6 address based on an IPv6 general prefix and enable IPv6 processing on an interface.

Binding and Advertising the 6PE Label to Advertise Prefixes

Perform this task to enable the binding and advertising of labels when advertising IPv6 prefixes to a specified BGP neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address | ipv6-address | peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address | ipv6-address | peer-group-name*} **update-source** *interface-type interface-number*

7. **address-family ipv6 [unicast]**
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified routing process.
Step 4	no bgp default ipv4-unicast Example: <pre>Device(config-router)# no bgp default ipv4-unicast</pre>	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. <p>Note Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the neighbor remote-as command unless you configure the no bgp default ipv4-unicast command before configuring the neighbor remote-as command.</p>
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: <pre>Device(config-router)# neighbor 192.168.99.70 remote-as 65000</pre>	Adds the IP address of the neighbor in the specified autonomous system to the BGP neighbor table of the local device.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: <pre>Device(config-router)# neighbor 192.168.99.70 update-source Loopback 0</pre>	Specifies the interface whose IPv4 address is to be used as the source address for the peering. <ul style="list-style-type: none"> • In the context of this task, the interface must have an IPv4 address with a 32-bit mask configured. Use of a loopback interface is recommended. This address is used to determine the IPv6 next hop by the peer 6PE.
Step 7	address-family ipv6 [unicast] Example:	Specifies the IPv6 address family and enters address family configuration mode.

	Command or Action	Purpose
	Device(config-router)# address-family ipv6	<ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Device(config-router-af) # neighbor 192.168.99.70 activate	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device.
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> } send-label Example: Device(config-router-af) # neighbor 192.168.99.70 send-label	Advertises the capability of the device to send MPLS labels with BGP routes. <ul style="list-style-type: none"> In IPv6 address family configuration mode this command enables binding and advertisement of labels when advertising IPv6 prefixes in BGP.

Configuring IBGP Multipath Load Sharing

Perform this task to configure IBGP multipath load sharing and control the maximum number of parallel IBGP routes that can be installed in a routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast**]
5. **maximum-paths ibgp** *number-of-paths*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [unicast] Example: Device(config-router)# address-family ipv6	Specifies the IPv6 address family and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 5	maximum-paths ibgp <i>number-of-paths</i> Example: Device(config-router)# maximum-paths ibgp 3	Controls the maximum number of parallel IBGP routes that can be installed in a routing table.

Configuration Examples for IPv6 Switching: Provider Edge Router over MPLS

Example: Provider Edge Device

The 6PE device is configured for both IPv4 and IPv6 traffic. Gigabit Ethernet interface 0/0/0 is configured with an IPv4 address and is connected to a device in the core of the network. Integrated IS-IS and TDP configurations on this device are similar to the P1 device.

Device 6PE1 exchanges IPv6 routing information with another 6PE device using internal BGP (IBGP) established over an IPv4 connection so that all the **neighbor** commands use the IPv4 address of the 6PE2 device. All the BGP peers are within autonomous system 65000, so synchronization with IGP is turned off for IPv4. In IPv6 address family configuration mode, synchronization is disabled by default.

IPv6 and Cisco Express Forwarding for IPv6 are enabled, the 6PE2 neighbor is activated, and label binding and advertisement is enabled for IPv6 prefixes using the **neighbor send-label** command. Connected and static IPv6 routes are redistributed using BGP.



Note MPLS is not supported on IPv6.

Example: Core Device

In the following example, the device in the core of the network is running MPLS, IS-IS, and IPv4 only. The Gigabit Ethernet interfaces are configured with IPv4 address and are connected to the 6PE devices. IS-IS is

the IGP for this network and the P1 and 6PE devices are in the same IS-IS area 49.0001. Tag Distribution Protocol (TDP) and tag switching are enabled on both the Gigabit Ethernet interfaces. Cisco Express Forwarding is enabled in global configuration mode.

```
ip cef
!
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 192.168.99.200 255.255.255.255
!
interface GigabitEthernet0/0/0
 description to_6PE1
 ip address 192.168.99.2 255.255.255.252
 ip router isis
 tag-switching ip
!
interface GigabitEthernet0/1/0
 description to_6PE2
 ip address 192.168.99.66 255.255.255.252
 ip router isis
 tag-switching ip
router isis
 passive-interface Loopback0
 net 49.0001.1921.6809.9200.00
```

Example: Monitoring 6PE

In the following example, output information about an IPv6 route is displayed using the **show bgp ipv6** command with an IPv6 prefix:

```
Device# show bgp ipv6 2001:DB8:DDDD::/48

BGP routing table entry for 2001:DB8:DDDD::/48, version 15
Paths: (1 available, best #1, table Global-IPv6-Table)
  Not advertised to any peer
  Local
    ::FFFF:192.168.99.70 (metric 20) from 192.168.99.70 (192.168.99.70)
      Origin IGP, localpref 100, valid, internal, best
```

In the following example, output information about a BGP peer including the IPv6 label capability is displayed using the **show bgp ipv6 neighbors** command with an IP address:

```
Device# show bgp ipv6 neighbors 192.168.99.70

BGP neighbor is 192.168.99.70, remote AS 65000, internal link
BGP version 4, remote router ID 192.168.99.70
BGP state = Established, up for 00:05:17
Last read 00:00:09, hold time is 0, keepalive interval is 60 seconds
Neighbor capabilities:
  Route refresh: advertised and received
  Address family IPv6 Unicast: advertised and received
  ipv6 MPLS Label capability: advertised and received
Received 54 messages, 0 notifications, 0 in queue
Sent 55 messages, 1 notifications, 0 in queue
Default minimum time between advertisement runs is 5 seconds

For address family: IPv6 Unicast
BGP table version 21880, neighbor version 21880
Index 1, Offset 0, Mask 0x2
Route refresh request: received 0, sent 0
```

```

77 accepted prefixes consume 4928 bytes
Prefix advertised 4303, suppressed 0, withdrawn 1328
Number of NLRIs in the update sent: max 1, min 0

```

In the following example, output information linking the MPLS label with prefixes is displayed using the **show mpls forwarding-table** command. If the 6PE feature is configured, the labels are aggregated because there are several prefixes for one local label, and the prefix column contains IPv6 instead of a target prefix.

```
Device# show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Label	Outgoing interface	Next Hop
16	Pop Label	1.1.1.1/32	0		Et0/0	10.0.0.1
18	No Label	nh-id(1)	0		Et2/0	10.0.2.2
19	No Label	nh-id(2)	0		Et1/0	10.0.1.2
20	No Label	nh-id(3)	0		Et1/0	10.0.1.2
22	No Label	nh-id(5)	0		Et1/0	10.0.1.2
24	No Label	nh-id(5)	0		Et2/0	10.0.2.2

In the following example, output information about the top of the stack label with label switching information is displayed using the **show bgp ipv6 labels** command with the **labels** keyword:

```
Device# show bgp ipv6 labels
```

Network	Next Hop	In tag/Out tag
2001:DB8:DDDD::/64	::FFFF:192.168.99.70	notag/20

In the following example, output information about labels from the Cisco Express Forwarding table is displayed using the **show ipv6 cef** command with an IPv6 prefix:

```
Device# show ipv6 cef 2001:DB8:DDDD::/64
```

```

2001:DB8:DDDD::/64
  nexthop ::FFFF:192.168.99.70
  fast tag rewrite with Se0/0, point2point, tags imposed {19 20}

```

In the following example, output information from the IPv6 routing table is displayed using the **show ipv6 route** command. The output shows the IPv6 MPLS virtual interface as the output interface of IPv6 routes forwarded across the MPLS cloud. This example shows output from the 6PE1 router.

The 6PE2 router has advertised the IPv6 prefix of 2001:DB8:dddd::/48 configured for the CE2 router and the next-hop address is the IPv4-compatible IPv6 address ::ffff:192.168.99.70, where 192.168.99.70 is the IPv4 address of the 6PE2 router.

```
Device# show ipv6 route
```

```

IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
B 2001:DB8:DDDD::/64 [200/0]
  via ::FFFF:192.168.99.70, IPv6-mpls
B 2001:DB8:DDDD::/64 [200/0]
  via ::FFFF:192.168.99.70, IPv6-mpls
L 2001:DB8:FFFF::1/128 [0/0]
  via ::, GigabitEthernet0/0/0
C 2001:DB8:FFFF::/64 [0/0]
  via ::, GigabitEthernet0/0/0
S 2001:DB8:FFFF::/48 [1/0]
  via 2001:DB8:B00:FFFF::2, GigabitEthernet0/0/0

```

Additional References for IPv6 Switching: Provider Edge Router over MPLS

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS commands	Master Commands List, All Releases
IPv6 commands	IPv6 Command Reference
Cisco IOS IPv6 features	IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	<i>IPv6 RFCs</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 6

Multi-VRF Support

The Multi-VRF Support feature allows you to configure and maintain more than one instance of a routing and forwarding table within the same customer edge (CE) device.

- [Prerequisites for Multi-VRF Support, on page 101](#)
- [Restrictions for Multi-VRF Support, on page 101](#)
- [Information About Multi-VRF Support, on page 102](#)
- [How to Configure Multi-VRF Support, on page 104](#)
- [Configuration Examples for Multi-VRF Support, on page 112](#)
- [Additional References, on page 117](#)

Prerequisites for Multi-VRF Support

The network's core and provider edge (PE) devices must be configured for Virtual Private Network (VPN) operation.

Restrictions for Multi-VRF Support

- You can configure the Multi-VRF Support feature only on Layer 3 interfaces.
- The Multi-VRF Support feature is not supported by Interior Gateway Routing Protocol (IGRP) nor Intermediate System to Intermediate System (IS-IS).
- Label distribution for a given VPN routing and forwarding (VRF) instance on a given device can be handled by either Border Gateway Protocol (BGP) or Label Distribution Protocol (LDP), but not by both protocols at the same time.
- Multicast cannot operate on a Layer 3 interface that is configured with the Multi-VRF Support feature.

Information About Multi-VRF Support

How the Multi-VRF Support Feature Works

The Multi-VRF Support feature enables a service provider to support two or more Virtual Private Networks (VPNs), where the IP addresses can overlap several VPNs. The Multi-VRF Support feature uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each virtual routing and forwarding (VRF) instance. Interfaces in a VRF can be either physical, such as FastEthernet ports, or logical, such as VLAN switched virtual interfaces (SVIs) bridge domain interfaces (BDIs), but a Layer 3 interface cannot belong to more than one VRF at any one time. The Multi-VRF Support feature allows an operator to support two or more routing domains on a customer edge (CE) device, with each routing domain having its own set of interfaces and its own set of routing and forwarding tables. The Multi-VRF Support feature makes it possible to extend the label switched paths (LSPs) to the CE and into each routing domain that the CE supports.

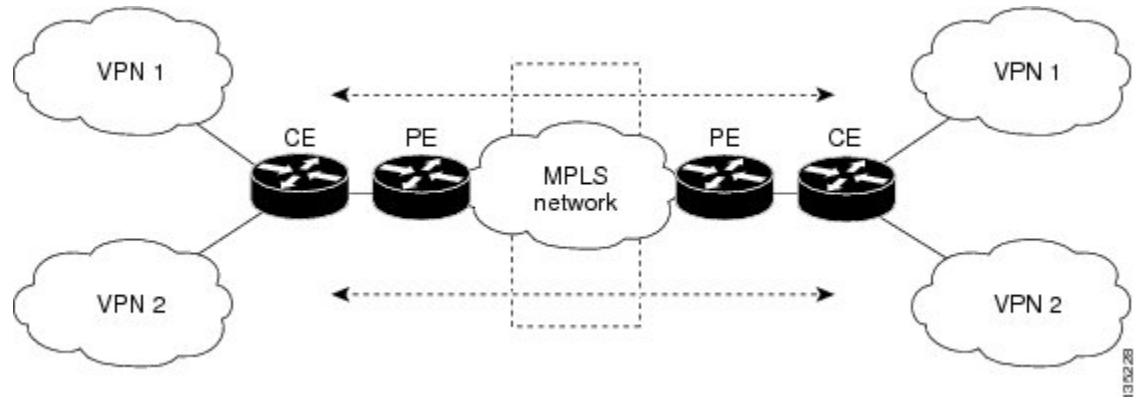
The Multi-VRF Support feature works as follows:

- Each CE device advertises its site's local routes to a provider edge (PE) device and learns the remote VPN routes from that provider edge (PE) device.
- PE devices exchange routing information with CE devices by using static routing or a routing protocol such as the Border Gateway Protocol (BGP), Routing Information Protocol version 1 (RIPv1), or RIPv2.
- PE devices exchange MPLS label information with CE devices through Label Distribution Protocol (LDP) or BGP.
- The PE device needs to maintain VPN routes only for those VPNs to which it is directly attached, eliminating the requirement that the PE maintain all of the service provider's VPN routes. Each PE device maintains a VRF for each of its directly connected sites. Two or more interfaces on a PE device can be associated with a single VRF if all the sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CE devices, the PE device exchanges VPN routing information with other PE devices through internal BGP (iBGP).

With the Multi-VRF Support feature, two or more customers can share one CE device, and only one physical link is used between the CE and the PE devices. The shared CE device maintains separate VRF tables for each customer and routes packets for each customer based on that customer's own routing table. The Multi-VRF Support feature extends limited PE device functionality to a CE device, giving it the ability, through the maintenance of separate VRF tables, to extend the privacy and security of a VPN to the branch office.

The figure below shows a configuration where each CE device acts as if it were two CE devices. Because the Multi-VRF Support feature is a Layer 3 feature, each interface associated with a VRF must be a Layer 3 interface.

Figure 12: Each CE Device Acting as Several Virtual CE Devices



How Packets Are Forwarded in a Network Using the Multi-VRF Support Feature

Following is the packet-forwarding process in an Multi-VRF customer edge (CE)-enabled network, as illustrated in the figure above:

- When the CE receives a packet from a Virtual Private Network (VPN), it looks up the routing table based on the input interface. When a route is found, the CE imposes the Multiprotocol Label Switching (MPLS) label that it received from the provider edge (PE) for that route and forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it swaps the incoming label with the corresponding label stack and sends the packet to the MPLS network.
- When an egress PE receives a packet from the network, it swaps the VPN label with the label that it had earlier received for the route from the CE, and it forwards the packet to the CE.
- When a CE receives a packet from an egress PE, it uses the incoming label on the packet to forward the packet to the correct VPN.

To configure Multi-VRF, you create a VRF table and then specify the Layer 3 interface associated with that VRF. Next, you configure the routing protocols within the VPN, and between the CE and the PE. The Border Gateway Protocol (BGP) is the preferred routing protocol for distributing VPN routing information across the provider's backbone.

The Multi-VRF network has three major components:

- VPN route target communities: These are lists of all other members of a VPN community. You must configure VPN route targets for each VPN community member.
- Multiprotocol BGP peering of VPN community PE devices: This propagates VRF reachability information to all members of a VPN community. You must configure BGP peering in all PE devices within a VPN community.
- VPN forwarding: This transports all traffic between VPN community members across a VPN service-provider network.

Considerations When Configuring the Multi-VRF Support Feature

- A device with the Multi-VRF Support feature is shared by several customers, and each customer has its own routing table.
- Because each customer uses a different virtual routing and forwarding (VRF) table, the same IP addresses can be reused. Overlapping IP addresses are allowed in different Virtual Private Networks (VPNs).
- The Multi-VRF Support feature lets several customers share the same physical link between the provider edge (PE) and the customer edge (CE) devices. Trunk ports with several VLANs separate packets among the customers. Each customer has its own VLAN.
- For the PE device, there is no difference between using the Multi-VRF Support feature or using several CE devices.
- The Multi-VRF Support feature does not affect the packet-switching rate.

How to Configure Multi-VRF Support

Configuring VRFs

To configure virtual routing and forwarding (VRF) instances, complete the following procedure. Be sure to configure VRFs on both the provider edge (PE) and customer edge (CE) devices.

If a VRF has not been configured, the device has the following default configuration:

- No VRFs have been defined.
- No import maps, export maps, or route maps have been defined.
- No VRF maximum routes exist.
- Only the global routing table exists on the interface.

**Note**

Multi-VRF/MVPN GRE configured layer-3 interface cannot participate in more than one VRF at the same time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **ip vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
7. **import map** *route-map*
8. **exit**
9. **interface** *type slot/subslot/port[.subinterface]*

10. **ip vrf forwarding** *vrf-name*
11. **end**
12. **show ip vrf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip routing Example: Device(config)# ip routing	Enables IP routing.
Step 4	ip vrf <i>vrf-name</i> Example: Device(config)# ip vrf v1	Names the VRF, and enters VRF configuration mode.
Step 5	rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 100:1	Creates a VRF table by specifying a route distinguisher. Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).
Step 6	route-target { export import both } <i>route-target-ext-community</i> Example: Device(config-vrf)# route-target export 100:1	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y). Note This command works only if BGP is running.
Step 7	import map <i>route-map</i> Example: Device(config-vrf)# import map importmap1	(Optional) Associates a route map with the VRF.
Step 8	exit Example:	Returns to global configuration mode.

	Command or Action	Purpose
	<code>Device(config-vrf)# exit</code>	
Step 9	interface <i>type slot/subslot/port[,subinterface]</i> Example: <code>Device(config)# interface FastEthernet3/0/0.10GigabitEthernet 0/0/0</code>	Specifies the Layer 3 interface to be associated with the VRF and enters interface configuration mode. The interface can be a routed port or an SVIBDI.
Step 10	ip vrf forwarding <i>vrf-name</i> Example: <code>Device(config-if)# ip vrf forwarding v1</code>	Associates the VRF with the Layer 3 interface.
Step 11	end Example: <code>Device(config-if)# end</code>	Returns to privileged EXEC mode.
Step 12	show ip vrf Example: <code>Device# show ip vrf</code>	Displays the settings of the VRFs.

Configuring BGP as the Routing Protocol

Most routing protocols can be used between the customer edge (CE) and the provider edge (PE) devices. However, external BGP (eBGP) is recommended, because:

- BGP does not require more than one algorithm to communicate with many CE devices.
- BGP is designed to pass routing information between systems run by different administrations.
- BGP makes it easy to pass route attributes to the CE device.

When BGP is used as the routing protocol, it can also be used to handle the Multiprotocol Label Switching (MPLS) label exchange between the PE and CE devices. By contrast, if Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), Routing Information Protocol (RIP), or static routing is used, the Label Distribution Protocol (LDP) must be used to signal labels.

To configure a BGP PE-to-CE routing session, perform the following steps on the CE and on the PE devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **network** *ip-address mask network-mask*
5. **redistribute ospf** *process-id match internal*
6. **network** *ip-address wildcard-mask area area-id*

7. **address-family ipv4 vrf vrf-name**
8. **neighbor {ip-address | peer-group-name} remote-as as-number**
9. **neighbor address activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp autonomous-system-number Example: Device(config)# router bgp 100	Configures the BGP routing process with the autonomous system number passed to other BGP devices, and enters router configuration mode.
Step 4	network ip-address mask network-mask Example: Device(config-router)# network 10.0.0.0 mask 255.255.255.0	Specifies a network and mask to announce using BGP.
Step 5	redistribute ospf process-id match internal Example: Device(config-router)# redistribute ospf 2 match internal	Sets the device to redistribute OSPF internal routes.
Step 6	network ip-address wildcard-mask area area-id Example: Device(config-router)# network 10.0.0.0 255.255.255.0 area 0	Identifies the network address and mask on which OSPF is running, and the area ID of that network address.
Step 7	address-family ipv4 vrf vrf-name Example: Device(config-router)# address-family ipv4 vrf v12	Identifies the name of the virtual routing and forwarding (VRF) instance that will be associated with the next two commands, and enters VRF address-family mode.
Step 8	neighbor {ip-address peer-group-name} remote-as as-number Example:	Informs this device's BGP neighbor table of the neighbor's address (or peer group name) and the neighbor's autonomous system number.

	Command or Action	Purpose
	Device(config-router-af) # neighbor 10.0.0.3 remote-as 100	
Step 9	neighbor address activate Example: Device(config-router-af) # neighbor 10.0.0.3 activate	Activates the advertisement of the IPv4 address-family neighbors.

Configuring PE-to-CE MPLS Forwarding and Signaling with BGP

If the Border Gateway Protocol (BGP) is used for routing between the provider edge (PE) and the customer edge (CE) devices, configure BGP to signal the labels on the virtual routing and forwarding (VRF) interfaces of both the CE and the PE devices. You must enable signalling globally at the router-configuration level and for each interface:

- At the router-configuration level, to enable Multiprotocol Label Switching (MPLS) label signalling via BGP, use the **neighbor send-label** command).
- At the interface level, to enable MPLS forwarding on the interface used for the PE-to-CE external BGP (eBGP) session, use the **mpls bgp forwarding** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **neighbor address send-label**
6. **neighbor address activate**
7. **end**
8. **configure terminal**
9. **interface** *type slot/subslot/port[.subinterface]*
10. **mpls bgp forwarding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# configure terminal	
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 100	Configures the BGP routing process with the autonomous system number passed to other BGP devices and enters router configuration mode.
Step 4	address-family ipv4 vrf <i>vrf-name</i> Example: Device(config-router)# address-family ipv4 vrf v12	Identifies the name of the VRF instance that will be associated with the next two commands and enters address family configuration mode.
Step 5	neighbor <i>address</i> send-label Example: Device(config-router-af)# neighbor 10.0.0.3 send-label	Enables the device to use BGP to distribute MPLS labels along with the IPv4 routes to the peer devices. If a BGP session is running when you issue this command, the command does not take effect until the BGP session is restarted.
Step 6	neighbor <i>address</i> activate Example: Device(config-router-af)# neighbor 10.0.0.3 activate	Activates the advertisement of the IPv4 address-family neighbors.
Step 7	end Example: Device(config-router-af)# end	Returns to privileged EXEC mode.
Step 8	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 9	interface <i>type slot/subslot/port[.subinterface]</i> Example: Device(config)# interface FastEthernet3/0/0.10GigabitEthernet 0/0/0	Enters interface configuration mode for the interface to be used for the BGP session. The interface can be a routed port or an SVIBDI.
Step 10	mpls bgp forwarding Example: Device(config-if)# mpls bgp forwarding	Enables MPLS forwarding on the interface.

Configuring a Routing Protocol Other than BGP

You can use the Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), or static routing. This configuration uses OSPF, but the process is the same for other protocols.

If you use OSPF as the routing protocol between the provider edge (PE) and the customer edge (CE) devices, issue the **capability vrf-lite** command in router configuration mode.



Note If RIP EIGRP, OSPF or static routing is used, the Label Distribution Protocol (LDP) must be used to signal labels.

The Multi-VRF Support feature is not supported by Interior Gateway Routing Protocol (IGRP) or Intermediate System-to-Intermediate System (IS-IS).

Multicast cannot be configured on the same Layer 3 interface as the Multi-VRF Support feature is configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **log-adjacency-changes**
5. **redistribute bgp** *autonomous-system-number* **subnets**
6. **network** *ip-address subnet-mask area* *area-id*
7. **end**
8. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router ospf <i>process-id</i> [vrf <i>vpn-name</i>] Example: Device(config)# router ospf 100 vrf v1	Enables OSPF routing, specifies a virtual routing and forwarding (VRF) table, and enters router configuration mode.
Step 4	log-adjacency-changes Example:	(Optional) Logs changes in the adjacency state. This is the default state.

	Command or Action	Purpose
	<code>Device(config-router)# log-adjacency-changes</code>	
Step 5	<p>redistribute bgp <i>autonomous-system-number</i> subnets</p> <p>Example:</p> <pre>Device(config-router)# redistribute bgp 800 subnets</pre>	Sets the device to redistribute information from the Border Gateway Protocol (BGP) network to the OSPF network.
Step 6	<p>network <i>ip-address subnet-mask area area-id</i></p> <p>Example:</p> <pre>Device(config-router)# network 10.0.0.0 255.255.255.0 area 0</pre>	Indicates the network address and mask on which OSPF runs, and the area ID of that network address.
Step 7	<p>end</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Returns to privileged EXEC mode.
Step 8	<p>show ip ospf</p> <p>Example:</p> <pre>Device# show ip ospf</pre>	Displays information about the OSPF routing processes.

Configuring PE-to-CE MPLS Forwarding and Signaling with LDP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface *type slot /subslot/port[.subinterface]***
4. **mpls ip**

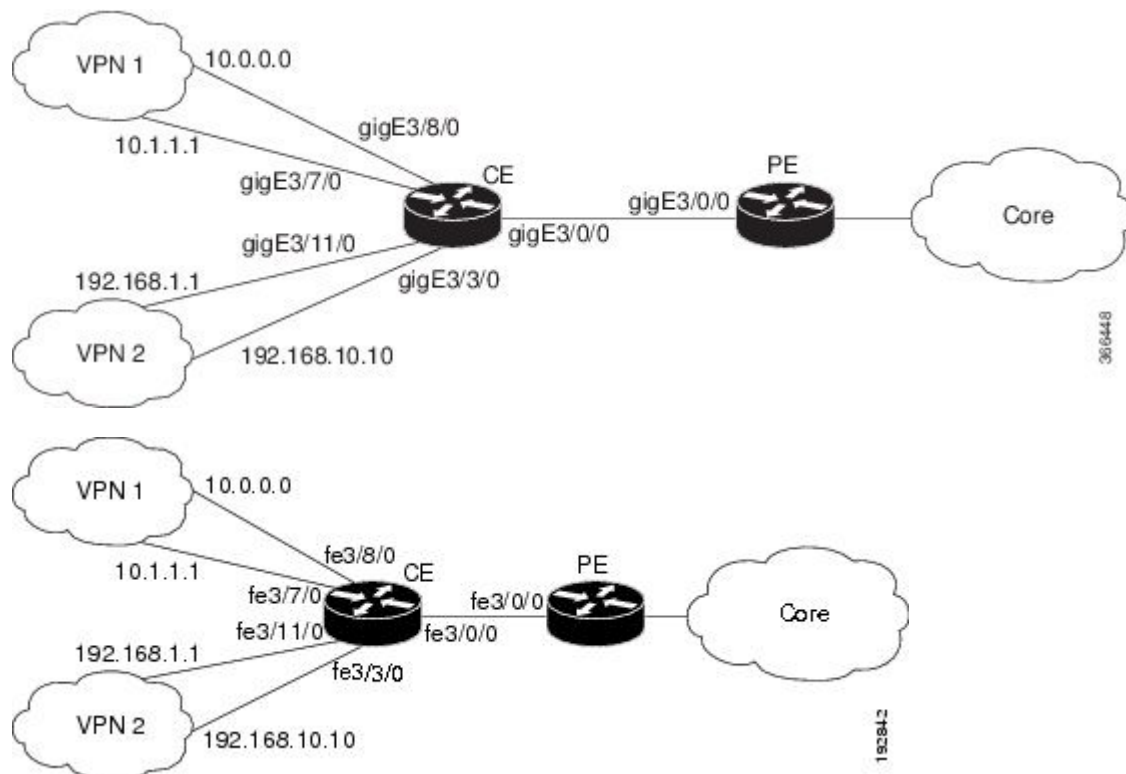
DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type slot /subslot/port[.subinterface]</i> Example: Device(config)# interface FastEthernet3/0/0.10GigabitEthernet 0/0/0	Enters interface configuration mode for the interface associated with the VRF. The interface can be a routed port or an SVIBDI.
Step 4	mpls ip Example: Device(config-if)# mpls ip	Enables MPLS forwarding of IPv4 packets along normally routed paths for this interface.

Configuration Examples for Multi-VRF Support

The figure below is an example of a Multi-VRF topology.



Example: Configuring Multi-VRF Support on the PE Device

The following example shows how to configure a VRF:

```
configure terminal
ip vrf v1
rd 100:1
```



```

route-target export 100:1
route-target import 100:1
exit
ip vrf v2
rd 100:2
route-target export 100:2
route-target import 100:2
exit

```

The following example shows how to configure on PE device, PE-to-CE connections using BGP for both routing and label exchange:

```

router bgp 100
address-family ipv4 vrf v2
neighbor 10.0.0.8 remote-as 800
neighbor 10.0.0.8 activate
neighbor 10.0.0.8 send-label
exit
address-family ipv4 vrf v1
neighbor 10.0.0.8 remote-as 800
neighbor 10.0.0.8 activate
neighbor 10.0.0.8 send-label
end
configure terminal
interface fastethernet3/0/0.10
ip vrf forwarding v1
ip address 10.0.0.3 255.255.255.0
mpls bgp forwarding
exit
interface fastethernet3/0/0.20
ip vrf forwarding v2
ip address 10.0.0.3 255.255.255.0
mpls bgp forwarding
exit

router bgp 100
address-family ipv4 vrf v2
neighbor 10.0.0.8 remote-as 800
neighbor 10.0.0.8 activate
neighbor 10.0.0.8 send-label
exit
address-family ipv4 vrf v1
neighbor 10.0.0.8 remote-as 800
neighbor 10.0.0.8 activate
neighbor 10.0.0.8 send-label
end
configure terminal
interface GigabitEthernet3/0/0
service instance 10 ethernet
encapsulation dot1q 10
rewrite ingress tag pop 1 symmetric
bridge-domain 10
!
service instance 20 ethernet
encapsulation dot1q 20
rewrite ingress tag pop 1 symmetric
bridge-domain 20

interface BDI10
ip vrf forwarding v11
ip address 10.0.0.3 255.255.255.0
mpls bgp forwarding
exit

```

```
interface BDI20
  ip vrf forwarding v12
  ip address 10.0.0.3 255.255.255.0
  mpls bgp forwarding
  exit
```

The following example shows how to configure on PE device, PE-to-CE connections using OSPF for routing and LDP for label exchange:

```
router ospf 100 vrf v1
  network 10.0.0.0 255.255.255.0 area 0
  exit
router ospf 101 vrf v2
  network 10.0.0.0 255.255.255.0 area 0
  exit
interface fastethernet3/0/0.10
  ip vrf forwarding v1
  ip address 10.0.0.3 255.255.255.0
  mpls ip
  exit
interface fastethernet3/0/0.20
  ip vrf forwarding v2
  ip address 10.0.0.3 255.255.255.0
  mpls ip
  exit
```

```
router ospf 100 vrf v1
  network 10.0.0.0 255.255.255.0 area 0
  exit
router ospf 101 vrf v2
  network 10.0.0.0 255.255.255.0 area 0
  exit
interface GigabitEthernet3/0/0
  service instance 10 ethernet
  encapsulation dot1q 10
  rewrite ingress tag pop 1 symmetric
  bridge-domain 10
  !
  service instance 20 ethernet
  encapsulation dot1q 20
  rewrite ingress tag pop 1 symmetric
  bridge-domain 20
```

```
interface BDI10
  ip vrf forwarding v11
  ip address 10.0.0.3 255.255.255.0
  mpls bgp forwarding
  exit
interface BDI20
  ip vrf forwarding v12
  ip address 10.0.0.3 255.255.255.0
  mpls bgp forwarding
  exit
```

Example: Configuring Multi-VRF Support on the CE Device

The following example shows how to configure VRFs:

```
configure terminal
  ip routing
  ip vrf v11
```

```

rd 800:1
route-target export 800:1
route-target import 800:1
exit
ip vrf v12
rd 800:2
route-target export 800:2
route-target import 800:2
exit

```

The following example shows how to configure CE device VPN connections:

```

interface fastethernet3/8/0GigabitEthernet 3/8/0
ip vrf forwarding v11
ip address 10.0.0.8 255.255.255.0
exit
interface fastethernet3/11/0GigabitEthernet 3/11/0
ip vrf forwarding v12
ip address 10.0.0.8 255.255.255.0
exit
router ospf 1 vrf v11
network 10.0.0.0 255.255.255.0 area 0
network 10.0.0.0 255.255.255.0 area 0
exit
router ospf 2 vrf v12
network 10.0.0.0 255.255.255.0 area 0
network 10.0.0.0 255.255.255.0 area 0
exit

```



Note If BGP is used for routing between the PE and CE devices, the BGP-learned routes from the PE device can be redistributed into OSPF using the commands in the following example.

```

router ospf 1 vrf v11
redistribute bgp 800 subnets
exit
router ospf 2 vrf v12
redistribute bgp 800 subnets
exit

```

The following example shows how to configure on CE devices, PE-to-CE connections using BGP for both routing and label exchange:

```

router bgp 800
address-family ipv4 vrf v12
neighbor 10.0.0.3 remote-as 100
neighbor 10.0.0.3 activate
neighbor 10.0.0.3 send-label
redistribute ospf 2 match internal
exit
address-family ipv4 vrf v11
neighbor 10.0.0.3 remote-as 100
neighbor 10.0.0.3 activate
neighbor 10.0.0.3 send-label
redistribute ospf 1 match internal
end
interface fastethernet3/0/0.10
ip vrf forwarding v11
ip address 10.0.0.8 255.255.255.0
mpls bgp forwarding

```

```

exit
interface fastethernet3/0/0.20
 ip vrf forwarding v12
 ip address 10.0.0.8 255.255.255.0
 mpls bgp forwarding
 exit

router bgp 800
 address-family ipv4 vrf v12
 neighbor 10.0.0.3 remote-as 100
 neighbor 10.0.0.3 activate
 neighbor 10.0.0.3 send-label
 redistribute ospf 2 match internal
 exit
 address-family ipv4 vrf v11
 neighbor 10.0.0.3 remote-as 100
 neighbor 10.0.0.3 activate
 neighbor 10.0.0.3 send-label
 redistribute ospf 1 match internal
 end
interface GigabitEthernet3/0/0
 service instance 10 ethernet
 encapsulation dot1q 10
 rewrite ingress tag pop 1 symmetric
 bridge-domain 10
!
 service instance 20 ethernet
 encapsulation dot1q 20
 rewrite ingress tag pop 1 symmetric
 bridge-domain 20

interface BDI10
 ip vrf forwarding v11
 ip address 10.0.0.8 255.255.255.0
 mpls bgp forwarding
 exit
interface BDI20
 ip vrf forwarding v12
 ip address 10.0.0.8 255.255.255.0
 mpls bgp forwarding
 exit

```

The following example shows how to configure on CE devices, PE-to-CE connections using OSPF for both routing and LDP for label exchange:

```

router ospf 1 vrf v11
 network 10.0.0.0 255.255.255.0 area 0
 exit
router ospf 2 vrf v12
 network 10.0.0.0 255.255.255.0 area 0
 exit
interface fastethernet3/0/0.10
 ip vrf forwarding v11
 ip address 10.0.0.3 255.255.255.0
 mpls ip
 exit
interface fastethernet3/0/0.20
 ip vrf forwarding v12
 ip address 10.0.0.3 255.255.255.0
 mpls ip
 exit

```

```

router ospf 1 vrf v11
 network 10.0.0.0 255.255.255.0 area 0
 exit
router ospf 2 vrf v12
 network 10.0.0.0 255.255.255.0 area 0
 exit
interface GigabitEthernet3/0/0
 service instance 10 ethernet
  encapsulation dot1q 10
  rewrite ingress tag pop 1 symmetric
  bridge-domain 10
 !
 service instance 20 ethernet
  encapsulation dot1q 20
  rewrite ingress tag pop 1 symmetric
  bridge-domain 20

interface BDI10
 ip vrf forwarding v11
 ip address 10.0.0.8 255.255.255.0
 mpls bgp forwarding
 exit
interface BDI20
 ip vrf forwarding v12
 ip address 10.0.0.8 255.255.255.0
 mpls bgp forwarding
 exit

```

Additional References

Related Documents

Related Topic	Document Title
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
OSPF with Multi-VRF	“OSPF Support for Multi-VRF in CE Routers” module in the OSPF Configuration Guide .

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 7

ECMP Load Balancing

Equal-cost multi-path routing (ECMP) is a routing strategy where next-hop packet forwarding to a single destination can occur over multiple "best paths" which tie for top place in routing metric calculations. Multipath routing can be used in conjunction with most routing protocols, since it is a per-hop decision that is limited to a single router. It potentially offers substantial increases in bandwidth by load-balancing traffic over multiple paths.

Various routing protocols, including Open Shortest Path First (OSPF), Intermediate System to Intermediate System (ISIS), Enhanced Interior Gateway Routing Protocol (EIGRP), and Border Gateway Protocol (BGP) allow ECMP routing.

Load balancing between ECMP paths is performed on IOS-XE based CEF object called loadbalance.

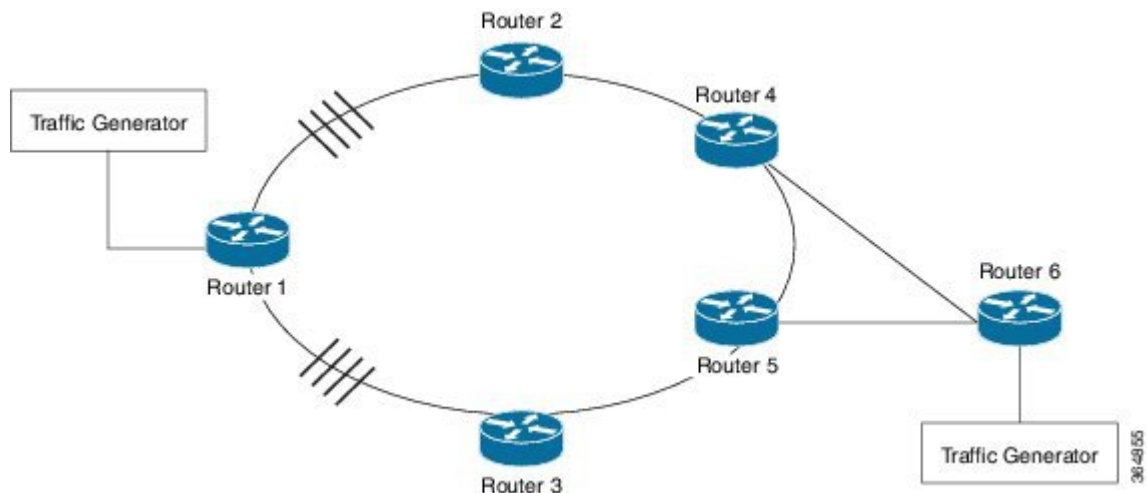
ECMP Per-Flow Load Balancing

Load balancing is a forwarding mechanism that distributes traffic over multiple links based on certain parameters. ECMP Per-Flow Load Balancing distributes packets across multiple links based on Layer 3 routing information. If the router discovers multiple paths to a destination, the routing table is updated with multiple entries for that destination. Per-flow load balancing allows the router to use multiple paths to achieve load sharing across multiple source-destination host pairs. Packets for a given source-destination host pair are guaranteed to take the same path, even if multiple paths are available. Traffic streams destined for different pairs tend to take different paths.

Benefits of Per-Flow Load Balancing

- Incoming data traffic is evenly distributed over multiple equal-cost connections.
- Incoming data traffic is evenly distributed over multiple equal-cost connections member links within a bundle interface.

Figure 13: ECMP Load Balancing with MPLS Enabled



- [Restrictions for ECMP Load Balancing, on page 120](#)
- [Configuring ECMP Load Balancing, on page 120](#)

Restrictions for ECMP Load Balancing

- Both 4 ECMP and 8 ECMP paths are supported.
- Load balancing is supported on global IPv4 and IPv6 traffic. For global IPv4 and IPv6 traffic, the traffic distribution can be equal among the available 8 links.
- Per packet load balancing is not supported.
- Label load balancing is supported.
- BGP multi-path is *not* supported with ECMP.
- BGP multi-path with PIC Edge is *not* supported
- When BGP PIC is configured, the L3VPN prefixes scale reduces by 1/4th of the supported value (Supported scale value/4), for better convergence value at the PIC core. For example, for RSP1A the supported L3VPN scale is 2000, if 4 ECMP path with PIC is enabled, then the maximum supported scale value is reduced to 5000 (20000/4).

Configuring ECMP Load Balancing

Perform the following steps to configure ECMP load balancing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **platform loadbalance max-paths 8**

4. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	platform loadbalance max-paths 8 Example: Device(config)# platform loadbalance max-paths 8	Configures the loadbalance maximum paths. Select Yes to save the configuration and reload the router. Note ISIS by default supports only 4 paths. To increase ISIS max-paths, use the command config-maximum-paths 8 under router ISIS. IGP by default supports only 4 paths. To increase IGP max-paths, use the command config-maximum-paths 8 under respective IGP (OSPF and ISIS) process.
Step 4	exit Example: Device (config) #exit	Returns to privileged EXEC mode.

Configuration Examples for ECMP Load Balancing

This section shows sample configurations for ECMP load balancing.

Example: Configuring ECMP Load balancing

The following is a sample configuration for ECMP load balancing.

```
Router# show ip cef 200.0.0.0 detail
200.0.0.0/24, epoch 2, per-destination sharing
  local label info: global/266
  nexthop 21.1.1.2 GigabitEthernet0/1/3 label 141
  nexthop 21.1.6.1 GigabitEthernet0/0/0 label 269
  nexthop 21.2.1.2 GigabitEthernet0/1/0 label 141
  nexthop 21.2.6.1 GigabitEthernet0/0/1 label 269

Router#

Router# show interface GigabitEthernet 0/1/3 | in output rate
 5 minute output rate 548000 bits/sec, 1009 packets/sec
Router# show interface GigabitEthernet 0/0/0 | in output rate
 5 minute output rate 547000 bits/sec, 1008 packets/sec
Router# show interface GigabitEthernet 0/1/0 | in output rate
```



```
ip address 65.1.109.1 255.255.255.255
!
interface Loopback110
ip address 65.1.110.1 255.255.255.255
!
interface Loopback111
ip address 65.1.111.1 255.255.255.255
!
interface Loopback112
ip address 65.1.112.1 255.255.255.255
!
interface Loopback113
ip address 65.1.113.1 255.255.255.255
!
interface Loopback114
ip address 65.1.114.1 255.255.255.255
!
interface Loopback115
ip address 65.1.115.1 255.255.255.255
!
interface Loopback116
ip address 65.1.116.1 255.255.255.255
!
interface Loopback117
ip address 65.1.117.1 255.255.255.255
!
interface Loopback118
ip address 65.1.118.1 255.255.255.255
!
interface Loopback119
ip address 65.1.119.1 255.255.255.255
!
interface Loopback120
ip address 65.1.120.1 255.255.255.255
!
interface Loopback121
ip address 65.1.121.1 255.255.255.255
!
interface Loopback122
ip address 65.1.122.1 255.255.255.255
!
interface Loopback123
ip address 65.1.123.1 255.255.255.255
!
interface Loopback124
ip address 65.1.124.1 255.255.255.255
!
interface Loopback125
ip address 65.1.125.1 255.255.255.255
!
interface Loopback126
ip address 65.1.126.1 255.255.255.255
!
interface Loopback127
ip address 65.1.127.1 255.255.255.255
!
interface Loopback128
ip address 65.1.128.1 255.255.255.255
!
interface Loopback129
ip address 65.1.129.1 255.255.255.255
!
interface Loopback130
ip address 65.1.130.1 255.255.255.255
```

```
!  
interface Loopback131  
ip address 65.1.131.1 255.255.255.255  
!  
interface Loopback132  
ip address 65.1.132.1 255.255.255.255  
!  
interface Loopback133  
ip address 65.1.133.1 255.255.255.255  
!  
interface Loopback134  
ip address 65.1.134.1 255.255.255.255  
!  
interface Loopback135  
ip address 65.1.135.1 255.255.255.255  
!  
interface Loopback136  
ip address 65.1.136.1 255.255.255.255  
!  
interface Loopback137  
ip address 65.1.137.1 255.255.255.255  
!  
interface Loopback138  
ip address 65.1.138.1 255.255.255.255  
!  
interface Loopback139  
ip address 65.1.139.1 255.255.255.255  
!  
interface Loopback140  
ip address 65.1.140.1 255.255.255.255  
!  
interface Loopback141  
ip address 65.1.141.1 255.255.255.255  
!  
interface Loopback142  
ip address 65.1.142.1 255.255.255.255  
!  
interface Loopback143  
ip address 65.1.143.1 255.255.255.255  
!  
interface Loopback144  
ip address 65.1.144.1 255.255.255.255  
!  
interface Loopback145  
ip address 65.1.145.1 255.255.255.255  
!  
interface Loopback146  
ip address 65.1.146.1 255.255.255.255  
!  
interface Loopback147  
ip address 65.1.147.1 255.255.255.255  
!  
interface Loopback148  
ip address 65.1.148.1 255.255.255.255  
!  
interface Loopback149  
ip address 65.1.149.1 255.255.255.255  
!  
interface Loopback150  
ip address 65.1.150.1 255.255.255.255  
!  
interface Loopback151  
ip address 65.1.151.1 255.255.255.255  
!
```

```
interface Loopback152
ip address 65.1.152.1 255.255.255.255
!
interface Loopback153
ip address 65.1.153.1 255.255.255.255
!
interface Loopback154
ip address 65.1.154.1 255.255.255.255
!
interface Loopback155
ip address 65.1.155.1 255.255.255.255
!
interface Loopback156
ip address 65.1.156.1 255.255.255.255
!
interface Loopback157
ip address 65.1.157.1 255.255.255.255
!
interface Loopback158
ip address 65.1.158.1 255.255.255.255
!
interface Loopback159
ip address 65.1.159.1 255.255.255.255
!
interface Loopback160
ip address 65.1.160.1 255.255.255.255
!
interface GigabitEthernet0/0/0
ip address 21.1.6.2 255.255.255.0
ip router isis core-agg
negotiation auto
mpls ip
!
interface GigabitEthernet0/0/1
ip address 21.2.6.2 255.255.255.0
ip router isis core-agg
negotiation auto
mpls ip
!
interface GigabitEthernet0/0/2
ip address 21.3.6.2 255.255.255.0
ip router isis core-agg
negotiation auto
mpls ip
!
interface GigabitEthernet0/0/3
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet0/0/4
ip address 21.4.1.1 255.255.255.0
ip router isis core-agg
shutdown
negotiation auto
mpls ip
!
interface GigabitEthernet0/0/5
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet0/0/6
ip address 51.1.0.1 255.255.255.0
```

```
negotiation auto
!
interface GigabitEthernet0/0/7
ip address 21.4.6.2 255.255.255.0
ip router isis core-agg
shutdown
negotiation auto
mpls ip
!
interface GigabitEthernet0/1/0
ip address 21.2.1.1 255.255.255.0
ip router isis core-agg
negotiation auto
mpls ip
!
interface GigabitEthernet0/1/1
ip address 21.3.1.1 255.255.255.0
ip router isis core-agg
negotiation auto
mpls ip
!
interface GigabitEthernet0/1/2
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet0/1/3
ip address 21.1.1.1 255.255.255.0
ip router isis core-agg
negotiation auto
mpls ip
!
interface GigabitEthernet0/1/4
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet0/1/5
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet0/1/6
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet0/1/7
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet0
vrf forwarding Mgmt-intf
ip address 7.43.21.101 255.255.0.0
shutdown
negotiation auto
!
router isis core-agg
net 49.0000.0000.1111.00
is-type level-1
metric-style wide
fast-flood 10
ip route priority high tag 10000
```

```
set-overload-bit on-startup 360
max-lsp-lifetime 65535
lsp-refresh-interval 65000
spf-interval 5 50 200
prc-interval 5 50 200
lsp-gen-interval 5 50 200
log-adjacency-changes
passive-interface Loopback0
passive-interface Loopback101
passive-interface Loopback102
passive-interface Loopback103
passive-interface Loopback104
passive-interface Loopback105
passive-interface Loopback106
passive-interface Loopback107
passive-interface Loopback108
passive-interface Loopback109
passive-interface Loopback110
passive-interface Loopback111
passive-interface Loopback112
passive-interface Loopback113
passive-interface Loopback114
passive-interface Loopback115
passive-interface Loopback116
passive-interface Loopback117
passive-interface Loopback118
passive-interface Loopback119
passive-interface Loopback120
passive-interface Loopback121
passive-interface Loopback122
passive-interface Loopback123
passive-interface Loopback124
passive-interface Loopback125
passive-interface Loopback126
passive-interface Loopback127
passive-interface Loopback128
passive-interface Loopback129
passive-interface Loopback130
passive-interface Loopback131
passive-interface Loopback132
passive-interface Loopback133
passive-interface Loopback134
passive-interface Loopback135
passive-interface Loopback136
passive-interface Loopback137
passive-interface Loopback138
passive-interface Loopback139
passive-interface Loopback140
passive-interface Loopback141
passive-interface Loopback142
passive-interface Loopback143
passive-interface Loopback144
passive-interface Loopback145
passive-interface Loopback146
passive-interface Loopback147
passive-interface Loopback148
passive-interface Loopback149
passive-interface Loopback150
passive-interface Loopback151
passive-interface Loopback152
passive-interface Loopback153
passive-interface Loopback154
passive-interface Loopback155
passive-interface Loopback156
```



```
passive-interface Loopback157
passive-interface Loopback158
passive-interface Loopback159
passive-interface Loopback160
maximum-paths 8
mpls ldp sync
!
router bgp 100
  bgp router-id 100.111.14.1
  bgp log-neighbor-changes
  neighbor ABR peer-group
  neighbor ABR remote-as 100
  neighbor ABR update-source Loopback0
  neighbor 100.111.10.1 peer-group ABR
  neighbor 100.111.10.1 shutdown
  neighbor 100.111.10.2 peer-group ABR
  neighbor 100.111.10.2 shutdown
!
address-family ipv4
  bgp additional-paths install
  network 100.111.14.1 mask 255.255.255.255 route-map set-PAN-comm
  neighbor ABR send-community both
  neighbor ABR next-hop-self all
  neighbor ABR route-map deny-PAN-loopbacks in
  neighbor ABR send-label
  neighbor 100.111.10.1 activate
  neighbor 100.111.10.2 activate
exit-address-family
!
address-family vpnv4
  neighbor ABR send-community both
  neighbor 100.111.10.1 activate
  neighbor 100.111.10.2 activate
exit-address-family
!
address-family ipv4 vrf test
  redistribute connected
exit-address-family
!
ip forward-protocol nd
!
ip bgp-community new-format
ip community-list 1 permit 100:100
no ip http server
no ip http secure-server
ip route vrf Mgmt-intf 10.0.0.0 255.0.0.0 7.43.0.1
ip route vrf Mgmt-intf 202.153.144.0 255.255.255.0 7.43.0.1
!
!
route-map set-service-nh permit 10
!
route-map deny-PAN-loopbacks deny 10
match community 1
!
route-map deny-PAN-loopbacks permit 20
!
route-map set-PAN-comm permit 10
set community 100:100
!
mpls ldp router-id Loopback0
!
!
control-plane
!
```

```
alias exec psh reques plat soft sys shell
alias exec shpp6 sh platform hard pp act fea cef da ipv6
alias exec shpp sh platform hard pp act fea cef da ipv4
!
line con 0
exec-timeout 0 0
logging synchronous
stopbits 1
line aux 0
stopbits 1
line vty 0
exec-timeout 0 0
password lab
login
line vty 1 4
login
!
!
!
end
```