CONTENTS

CHAPTER 1

MPLS Virtual Private Networks  1
  Finding Feature Information  1
  Prerequisites for MPLS Virtual Private Networks  1
  Restrictions for MPLS Virtual Private Networks  2
  Information About MPLS Virtual Private Networks  4
    MPLS Virtual Private Network Definition  4
    How an MPLS Virtual Private Network Works  5
      How Virtual Routing and Forwarding Tables Work in an MPLS Virtual Private Network  5
      How VPN Routing Information Is Distributed in an MPLS Virtual Private Network  6
  MPLS Forwarding  6
  Major Components of an MPLS Virtual Private Network  6
  Benefits of an MPLS Virtual Private Network  7
  How to Configure MPLS Virtual Private Networks  9
    Configuring the Core Network  9
      Assessing the Needs of MPLS Virtual Private Network Customers  9
      Configuring MPLS in the Core  10
    Connecting the MPLS Virtual Private Network Customers  10
      Defining VRFs on the PE Devices to Enable Customer Connectivity  10
      Configuring VRF Interfaces on PE Devices for Each VPN Customer  12
      Configuring Routing Protocols Between the PE and CE Devices  13
        Configuring RIPv2 as the Routing Protocol Between the PE and CE Devices  13
        Configuring Static Routes Between the PE and CE Devices  15
    Verifying the Virtual Private Network Configuration  16
      Verifying Connectivity Between MPLS Virtual Private Network Sites  17
        Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core  17
        Verifying That the Local and Remote CE Devices Are in the PE Routing Table  18
    Configuration Examples for MPLS Virtual Private Networks  19
IPv6 VPN over MPLS Overview 38
Addressing Considerations for IPv6 VPN over MPLS 39
Basic IPv6 VPN over MPLS Functionality 39
IPv6 VPN Architecture Overview 39
IPv6 VPN Next Hop 40
MPLS Forwarding 40
VRF Concepts 41
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 45
Configuring a Virtual Routing and Forwarding Instance for IPv6 45
Binding a VRF to an Interface 47
Configuring a Static Route for PE-to-CE Routing 49
Configuring eBGP PE-to-CE Routing Sessions 50
Configuring the IPv6 VPN Address Family for iBGP 51
Configuring Route Reflectors for Improved Scalability 52
Configuring Internet Access 60
Configuring the Internet Gateway 60
Configuring iBGP 6PE Peering to the VPN PE 60
Configuring the Internet Gateway as the Gateway to the Public Domain 62
Configuring eBGP Peering to the Internet 63
Configuring the IPv6 VPN PE 65
Configuring a Default Static Route from the VRF to the Internet Gateway 65
Configuring a Static Route from the Default Table to the VRF 66
Configuring iBGP 6PE Peering to the Internet Gateway 67
Configuring a Multiautonomous-System Backbone for IPv6 VPN 68
Configuring the PE VPN for a Multiautonomous-System Backbone 70
Configuring iBGP IPv6 VPN Peering to a Route Reflector 70
Configuring IPv4 and Label iBGP Peering to a Route Reflector 72
Configuring the Route Reflector for a Multiautonomous-System Backbone 74
Configuring Peering to the PE VPN 74
Configuring the Route Reflector 76
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Configuring Peering to the Autonomous System Boundary Router</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>Configuring Peering to Another ISP Route Reflector</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>Configuring the ASBR</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Configuring Peering with Router Reflector RR1</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Configuring Peering with the Other ISP ASBR2</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Configuring CSC for IPv6 VPN</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>Configuration Examples for IPv6 VPN over MPLS</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Examples: IPv6 VPN over MPLS Routing</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Example: BGP IPv6 Activity Summary</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Example: Dumping the BGP IPv6 Tables</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Example: Dumping the IPv6 Routing Tables</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Examples: IPv6 VPN over MPLS Forwarding</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Example: PE-CE Connectivity</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Examples: PE Imposition Path</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Examples: PE Disposition Path</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Examples: Label Switch Path</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Examples: IPv6 VPN over MPLS VRF</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Examples: VRF Information</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Example: IPv6 VPN Configuration Using IPv4 Next Hop</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Additional References</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>Feature Information for IPv6 VPN over MPLS</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td><strong>Glossary</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>CHAPTER 4</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>IPv6 Switching: Provider Edge Router over MPLS</strong></td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Finding Feature Information</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Prerequisites for IPv6 Switching: Provider Edge Router over MPLS</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Information About IPv6 Switching: Provider Edge Router over MPLS</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Benefits of Deploying IPv6 over MPLS Backbones</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>IPv6 on the Provider Edge Devices</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>How to Deploy IPv6 Switching: Provider Edge Router over MPLS</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Deploying IPv6 on the Provider Edge Devices (6PE)</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Specifying the Source Address Interface on a 6PE Device</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Binding and Advertising the 6PE Label to Advertise Prefixes</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Configuring IBGP Multipath Load Sharing</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Configuration Examples for IPv6 Switching: Provider Edge Router over MPLS</td>
<td>104</td>
</tr>
</tbody>
</table>
CHAPTER 5
Multi-VRF Support 109
Finding Feature Information 109
Prerequisites for Multi-VRF Support 109
Restrictions for Multi-VRF Support 110
Information About Multi-VRF Support 110
How the Multi-VRF Support Feature Works 110
How Packets Are Forwarded in a Network Using the Multi-VRF Support Feature 111
Considerations When Configuring the Multi-VRF Support Feature 112
How to Configure Multi-VRF Support 112
Configuring VRFs 112
Configuring BGP as the Routing Protocol 115
Configuring PE-to-CE MPLS Forwarding and Signaling with BGP 117
Configuring a Routing Protocol Other than BGP 119
Configuring PE-to-CE MPLS Forwarding and Signaling with LDP 120
Configuration Examples for Multi-VRF Support 121
Example: Configuring Multi-VRF Support on the PE Device 122
Example: Configuring Multi-VRF Support on the CE Device 123
Additional References 125
Feature Information for Multi-VRF Support 125

CHAPTER 6
ECMP Load Balancing 127
Finding Feature Information 128
Restrictions for ECMP Load Balancing 128
Configuring ECMP Load Balancing 129
Configuration Examples for ECMP Load Balancing 129
Example: Configuring ECMP Load balancing 129
Verifying ECMP Load Balancing 130
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.

- Finding Feature Information, page 1
- Prerequisites for MPLS Virtual Private Networks, page 1
- Restrictions for MPLS Virtual Private Networks, page 2
- Information About MPLS Virtual Private Networks, page 4
- How to Configure MPLS Virtual Private Networks, page 9
- Configuration Examples for MPLS Virtual Private Networks, page 19
- Additional References, page 21
- Feature Information for MPLS Virtual Private Networks, page 22

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.

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 an 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**

![Diagram of a basic MPLS VPN showing the MPLS Backbone, Provider (P) devices, Provider Edge (PE) devices, and Customer Site devices.]

### 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Identify the following to determine the number of devices and ports that you need:</td>
</tr>
<tr>
<td>Identify the size of the network.</td>
<td>- How many customers do you need to support?</td>
</tr>
<tr>
<td></td>
<td>- How many VPNs are needed per customer?</td>
</tr>
<tr>
<td></td>
<td>- How many virtual routing and forwarding instances are there for each VPN?</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Identify the routing protocols in the core. Determine which routing protocols you need in the core network.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Determine if you need Border Gateway Protocol (BGP) load sharing and redundant paths in the MPLS VPN core. For configuration steps, see the &quot;Load Sharing MPLS VPN Traffic&quot; feature module in the <em>MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide</em>.</td>
</tr>
</tbody>
</table>

### 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. exit
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enable privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Enters your password if prompted.</td>
</tr>
<tr>
<td><code>Device&gt;</code></td>
<td><code>enable</code></td>
</tr>
</tbody>
</table>

| **Step 2**        | Enters global configuration mode. |
| `configure terminal` | |
| **Example:**      | `Device# configure terminal` |

| **Step 3**        | Defines the virtual private network (VPN) routing instance by assigning a virtual routing and forwarding (VRF) name and enters VRF configuration mode. |
| `ip vrf vrf-name` | The *vrf-name* argument is the name assigned to a VRF. |
| **Example:**      | `Device(config)# ip vrf vpn1` |

| **Step 4**        | Creates routing and forwarding tables. |
| `rd route-distinguisher` | |
| **Example:**      | The *route-distinguisher* 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: |
|                   | - 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 |
| `Device(config-vrf)# rd 100:1` | |

| **Step 5**        | Creates a route-target extended community for a VRF. |
| `route-target {import | export | both} route-target-ext-community` | |
| **Example:**      | 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 *route-target-ext-community* argument adds the route-target extended community attributes to the VRF’s list of import, export, or both route-target extended communities. |
| `Device(config-vrf)# route-target both 100:1` | |

| **Step 6**        | (Optional) Exits to global configuration mode. |
| `exit`            | |
| **Example:**      | `Device(config-vrf)# exit` |
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>interface type number</td>
<td>Specifies the interface to configure and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Device(config)# interface GigabitEthernet 0/0/1 | • The type argument specifies the type of interface to be configured.  
|                                           | • The number argument specifies the port, connector, or interface card number. |
| **Step 4**              |                                |
| ip vrf forwarding vrf-name | Associates a VRF with the specified interface or subinterface. |
| **Example:**            |                                |
| Device(config-if)# ip vrf forwarding vpn1 | • The vrf-name argument is the name assigned to a VRF. |
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Device&gt; enable</strong></td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Device# configure terminal</strong></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>3</td>
<td><code>router rip</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Device(config)# router rip</code></td>
</tr>
<tr>
<td>4</td>
<td>`version {1</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router)# version 2</code></td>
</tr>
<tr>
<td>5</td>
<td>`address-family ipv4 [multicast</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router)# address-family ipv4 vrf vpn1</code></td>
</tr>
<tr>
<td>6</td>
<td><code>network ip-address</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router-af)# network 192.168.7.0</code></td>
</tr>
<tr>
<td>7</td>
<td>`redistribute protocol [process-id] [level-1</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router-af)# redistribute bgp 200</code></td>
</tr>
<tr>
<td>8</td>
<td><code>exit-address-family</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router-af)# exit-address-family</code></td>
</tr>
<tr>
<td>9</td>
<td><code>end</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router)# end</code></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip route vrf vrf-name</td>
<td>Defines static route parameters for every provider edge-to-customer edge (PE-to-CE) session and enters router configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# ip route vrf vpn1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 [multicast</td>
<td>unicast</td>
</tr>
<tr>
<td>Example: Device(config-router)# address-family ipv4 vrf vpn1</td>
<td>• The multicast keyword specifies IPv4 multicast address prefixes.</td>
</tr>
<tr>
<td></td>
<td>• The unicast keyword specifies IPv4 unicast address prefixes.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The <code>vrf vrf-name</code> keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> redistribute protocol [process-id] {level-1</td>
<td>level-1-2</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# redistribute static</td>
<td>To redistribute virtual routing and forwarding (VRF) static routes into the VRF Border Gateway Protocol (BGP) table, use the <code>redistribute static</code> command.</td>
</tr>
<tr>
<td><strong>Step 6</strong> redistribute protocol [process-id] {level-1</td>
<td>level-1-2</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# redistribute connected</td>
<td>To redistribute directly connected networks into the VRF BGP table, use the <code>redistribute connected</code> command.</td>
</tr>
<tr>
<td><strong>Step 7</strong> exit-address-family</td>
<td>Exits address family configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# exit-address-family</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>(Optional) Exits to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device(config-router)# end</td>
<td></td>
</tr>
</tbody>
</table>

### 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
   Enables privileged EXEC mode.
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.
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.
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.

DETAILED STEPS
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

<table>
<thead>
<tr>
<th>PE Configuration</th>
<th>CE Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip vrf vpn1</td>
<td>ip cef</td>
</tr>
<tr>
<td>rd 100:1</td>
<td>mpls ldp router-id Loopback0 force</td>
</tr>
<tr>
<td>route-target export 100:1</td>
<td>mpls label protocol ldp</td>
</tr>
<tr>
<td>route-target import 100:1</td>
<td>interface Loopback0</td>
</tr>
<tr>
<td>!</td>
<td>ip address 10.0.0.1 255.255.255.255</td>
</tr>
<tr>
<td>ip cef</td>
<td>interface GigabitEthernet 0/0/0</td>
</tr>
<tr>
<td>mpls ldp router-id Loopback0 force</td>
<td>ip address 192.0.2.1 255.255.255.0</td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td>no cdp enable</td>
</tr>
<tr>
<td>!</td>
<td>router rip</td>
</tr>
<tr>
<td>interface Loopback0</td>
<td>version 2</td>
</tr>
<tr>
<td>ip address 10.0.0.1 255.255.255.255</td>
<td>timers basic 30 60 60 120</td>
</tr>
<tr>
<td>!</td>
<td>address-family ipv4 vrf vpn1</td>
</tr>
<tr>
<td>interface GigabitEthernet 0/0/0</td>
<td>version 2</td>
</tr>
<tr>
<td>ip vrf forwarding vpn1</td>
<td>redistribute bgp 100 metric transparent</td>
</tr>
<tr>
<td>ip address 192.0.2.3 255.255.255.0</td>
<td>network 192.0.2.0</td>
</tr>
<tr>
<td>no cdp enable</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>interface GigabitEthernet 0/0/1</td>
<td>exit-address-family</td>
</tr>
<tr>
<td>ip address 192.0.2.2 255.255.255.0</td>
<td>router bgp 100</td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td>no synchronization</td>
</tr>
<tr>
<td>mpls ip</td>
<td>bgp log-neighbor changes</td>
</tr>
<tr>
<td>!</td>
<td>neighbor 10.0.0.3 remote-as 100</td>
</tr>
<tr>
<td>router rip</td>
<td>neighbor 10.0.0.3 update-source Loopback0</td>
</tr>
<tr>
<td>version 2</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>redistribute bgp 100 metric transparent</td>
<td>address-family vpnv4</td>
</tr>
<tr>
<td>network 192.0.2.0</td>
<td>neighbor 10.0.0.3 activate</td>
</tr>
<tr>
<td>distribute-list 20 in</td>
<td>neighbor 10.0.0.3 send-community extended</td>
</tr>
<tr>
<td>no auto-summary</td>
<td>bgp scan-time import 5</td>
</tr>
<tr>
<td>exit-address-family</td>
<td>exit-address-family</td>
</tr>
<tr>
<td>!</td>
<td>address-family ipv4 vrf vpn1</td>
</tr>
<tr>
<td>address-family ipv4 vrf vpn1</td>
<td>redistribute connected</td>
</tr>
<tr>
<td>redistribute rip</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>no auto-summary</td>
<td>no synchronization</td>
</tr>
<tr>
<td>exit-address-family</td>
<td>address-family ipv4 vrf vpn1</td>
</tr>
<tr>
<td>network 192.0.2.0</td>
<td>redistribute rip</td>
</tr>
<tr>
<td>no auto-summary</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>exit-address-family</td>
<td>no auto-summary</td>
</tr>
</tbody>
</table>

MPLS: Layer 3 VPNs Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 920 Series)
Example: Configuring an MPLS Virtual Private Network Using Static Routes

<table>
<thead>
<tr>
<th>PE Configuration</th>
<th>CE Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip vrf vpn1</td>
<td></td>
</tr>
<tr>
<td>rd 100:1</td>
<td></td>
</tr>
<tr>
<td>route-target export 100:1</td>
<td></td>
</tr>
<tr>
<td>route-target import 100:1</td>
<td></td>
</tr>
<tr>
<td>ip cef</td>
<td></td>
</tr>
<tr>
<td>mpls ldp router-id Loopback0 force</td>
<td></td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>interface Loopback0</td>
<td></td>
</tr>
<tr>
<td>ip address 10.0.0.1 255.255.255.255</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>interface GigabitEthernet 0/0/0</td>
<td></td>
</tr>
<tr>
<td>ip vrf forwarding vpn1</td>
<td></td>
</tr>
<tr>
<td>ip address 192.0.2.3 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>no cdp enable</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>interface GigabitEthernet 0/0/1</td>
<td></td>
</tr>
<tr>
<td>ip address 192.168.0.1 255.255.0.0</td>
<td></td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td></td>
</tr>
<tr>
<td>mpls ip</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>router ospf 100</td>
<td></td>
</tr>
<tr>
<td>network 10.0.0.0 0.0.0.0 area 100</td>
<td></td>
</tr>
<tr>
<td>network 192.168.0.0 255.255.0.0 area 100</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>router bgp 100</td>
<td></td>
</tr>
<tr>
<td>no synchronization</td>
<td></td>
</tr>
<tr>
<td>bgp log-neighbor changes</td>
<td></td>
</tr>
<tr>
<td>neighbor 10.0.0.3 remote-as 100</td>
<td></td>
</tr>
<tr>
<td>neighbor 10.0.0.3 update-source Loopback0</td>
<td></td>
</tr>
<tr>
<td>no auto-summary</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>address-family vpnv4</td>
<td></td>
</tr>
<tr>
<td>neighbor 10.0.0.3 activate</td>
<td></td>
</tr>
<tr>
<td>neighbor 10.0.0.3 send-community extended</td>
<td></td>
</tr>
<tr>
<td>bgp scan-time import 5</td>
<td></td>
</tr>
<tr>
<td>exit-address-family</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>address-family ipv4 vrf vpn1</td>
<td></td>
</tr>
<tr>
<td>redistribute connected</td>
<td></td>
</tr>
<tr>
<td>redistribute static</td>
<td></td>
</tr>
<tr>
<td>no auto-summary</td>
<td></td>
</tr>
<tr>
<td>no synchronization</td>
<td></td>
</tr>
<tr>
<td>exit-address-family</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>ip route vrf vpn1 10.0.0.9 255.255.255.255</td>
<td></td>
</tr>
<tr>
<td>192.0.2.2</td>
<td></td>
</tr>
<tr>
<td>ip route vrf vpn1 192.0.2.0 255.255.0.0</td>
<td></td>
</tr>
<tr>
<td>192.0.2.2</td>
<td></td>
</tr>
<tr>
<td>ip cef</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>interface Loopback0</td>
<td></td>
</tr>
<tr>
<td>ip address 10.0.0.9 255.255.255.255</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>interface GigabitEthernet 0/0/0</td>
<td></td>
</tr>
<tr>
<td>ip address 192.0.2.2 255.255.0.0</td>
<td></td>
</tr>
<tr>
<td>no cdp enable</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
<tr>
<td>ip route 10.0.0.9 255.255.255.255 192.0.2.3 3</td>
<td></td>
</tr>
<tr>
<td>ip route 198.51.100.0 255.255.255.0 192.0.2.3 3</td>
<td></td>
</tr>
</tbody>
</table>
## Additional References

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>Description of commands associated with MPLS and MPLS applications</td>
<td>Cisco IOS Multiprotocol Label Switching Command Reference</td>
</tr>
<tr>
<td>Configuring Cisco Express Forwarding</td>
<td>“Configuring Basic Cisco Express Forwarding” module in the <strong>Cisco Express Forwarding Configuration Guide</strong></td>
</tr>
<tr>
<td>Border Gateway Protocol (BGP) load sharing</td>
<td>“Load Sharing MPLS VPN Traffic” module in the <strong>MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide</strong></td>
</tr>
<tr>
<td>IPv6 VPN over MPLS</td>
<td>“IPv6 VPN over MPLS” module in the <strong>MPLS Layer 3 VPNs Configuration Guide</strong></td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
<tr>
<td>download documentation, software, and tools. Use these resources to install</td>
<td></td>
</tr>
<tr>
<td>and configure the software and to troubleshoot and resolve technical issues</td>
<td></td>
</tr>
<tr>
<td>with Cisco products and technologies. Access to most tools on the Cisco</td>
<td></td>
</tr>
<tr>
<td>Support and Documentation website requires a Cisco.com user ID and</td>
<td></td>
</tr>
<tr>
<td>password.</td>
<td></td>
</tr>
</tbody>
</table>
Feature Information for MPLS Virtual Private Networks

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 MPLS Virtual Private Networks

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS Virtual Private Networks</td>
<td>Cisco IOS XE Release 3.13.0S</td>
<td>This feature was introduced on the Cisco ASR 920 Routers (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D).</td>
</tr>
</tbody>
</table>
Chapter 2

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.

- Finding Feature Information, page 23
- Prerequisites for Multiprotocol BGP MPLS VPN, page 23
- Information About Multiprotocol BGP MPLS VPN, page 24
- How to Configure Multiprotocol BGP MPLS VPN, page 27
- Configuration Examples for Multiprotocol BGP MPLS VPN, page 34
- Additional References, page 35
- Feature Information for Multiprotocol BGP MPLS VPN, page 35

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.

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 vpnv4 [unicast]`
9. `neighbor {ip-address | peer-group-name} send-community extended`
10. `neighbor {ip-address | peer-group-name} activate`
11. `end`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp <strong>as-number</strong></td>
<td>Configures a Border Gateway Protocol (BGP) routing process and enters router configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# router bgp 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no bgp default ipv4-unicast</td>
<td>(Optional) Disables the IPv4 unicast address family on all neighbors.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# no bgp default ipv4-unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor **{ip-address</td>
<td>peer-group-name}** remote-as <strong>as-number</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# neighbor 10.0.0.1 remote-as 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor **{ip-address</td>
<td>peer-group-name}** update-source <strong>interface-type</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# neighbor 10.0.0.1 update-source Loopback0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> neighbor **{ip-address</td>
<td>peer-group-name}** activate</td>
</tr>
<tr>
<td><strong>Step 4</strong> no bgp default ipv4-unicast</td>
<td>(Optional) Disables the IPv4 unicast address family on all neighbors.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# no bgp default ipv4-unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor **{ip-address</td>
<td>peer-group-name}** remote-as <strong>as-number</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# neighbor 10.0.0.1 remote-as 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor **{ip-address</td>
<td>peer-group-name}** update-source <strong>interface-type</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# neighbor 10.0.0.1 update-source Loopback0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> neighbor **{ip-address</td>
<td>peer-group-name}** activate</td>
</tr>
<tr>
<td><strong>Step 4</strong> no bgp default ipv4-unicast</td>
<td>(Optional) Disables the IPv4 unicast address family on all neighbors.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# no bgp default ipv4-unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor **{ip-address</td>
<td>peer-group-name}** remote-as <strong>as-number</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# neighbor 10.0.0.1 remote-as 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor **{ip-address</td>
<td>peer-group-name}** update-source <strong>interface-type</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# neighbor 10.0.0.1 update-source Loopback0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> neighbor **{ip-address</td>
<td>peer-group-name}** activate</td>
</tr>
</tbody>
</table>

### Notes
- The **as-number** 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.
- Use the **no bgp default ipv4-unicast** command if you are using this neighbor for Multiprotocol Label Switching (MPLS) routes only.
- The **ip-address** argument specifies the IP address of the neighbor.
- The **peer-group-name** argument specifies the name of a BGP peer group.
- The **as-number** argument specifies the autonomous system to which the neighbor belongs.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>• The <code>peer-group-name</code> argument specifies the name of a BGP peer group.</td>
</tr>
<tr>
<td>Device(config-router)# neighbor 10.0.0.1 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• The optional <strong>unicast</strong> keyword specifies VPNv4 unicast address prefixes.</td>
</tr>
<tr>
<td>Device(config-router)# address-family vpnv4 [unicast]</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Specifies that a communities attribute should be sent to a BGP neighbor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• The <strong>ip-address</strong> argument specifies the IP address of the BGP-speaking neighbor.</td>
</tr>
<tr>
<td>Device(config-router-af)# neighbor {ip-address</td>
<td>peer-group-name} send-community extended</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Enables the exchange of information with a neighboring BGP device.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• The <strong>ip-address</strong> argument specifies the IP address of the neighbor.</td>
</tr>
<tr>
<td>Device(config-router-af)# neighbor {ip-address</td>
<td>peer-group-name} activate</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>(Optional) Exits to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router-af)# end</td>
<td></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp as-number</td>
<td>Configures a Border Gateway Protocol (BGP) routing process and enters router configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# router bgp 100</td>
<td>• The <em>as-number</em> 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.</td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 [multicast</td>
<td>unicast</td>
</tr>
<tr>
<td>Example: Device(config-router)# address-family ipv4 vrf vpn1</td>
<td>• The <em>multicast</em> keyword specifies IPv4 multicast address prefixes.</td>
</tr>
<tr>
<td></td>
<td>• The <em>unicast</em> keyword specifies IPv4 unicast address prefixes.</td>
</tr>
<tr>
<td></td>
<td>• The <em>vrf vrf-name</em> keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.</td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
**Step 5** | neighbor {ip-address | peer-group-name} remote-as as-number | Adds an entry to the BGP or multiprotocol BGP neighbor table.  
- The *ip-address* argument specifies the IP address of the neighbor.  
- The *peer-group-name* argument specifies the name of a BGP peer group.  
- The *as-number* argument specifies the autonomous system to which the neighbor belongs.

**Example:**  
Device(config-router-af)# neighbor 10.0.0.1 remote-as 200

**Step 6** | neighbor {ip-address | peer-group-name} activate | Enables the exchange of information with a neighboring BGP device.  
- The *ip-address* argument specifies the IP address of the neighbor.  
- The *peer-group-name* argument specifies the name of a BGP peer group.

**Example:**  
Device(config-router-af)# neighbor 10.0.0.1 activate

**Step 7** | exit-address-family | Exits address family configuration mode.

**Example:**  
Device(config-router-af)# exit-address-family

**Step 8** | end | (Optional) Exits to privileged EXEC mode.

**Example:**  
Device(config-router)# end

### 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]*
Discover 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**

<table>
<thead>
<tr>
<th>PE Configuration</th>
<th>CE Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip vrf vpn1</td>
<td>ip cef</td>
</tr>
<tr>
<td>rd 100:1</td>
<td>mpls ldp router-id Loopback0 force</td>
</tr>
<tr>
<td>route-target export 100:1</td>
<td>mpls label protocol ldp</td>
</tr>
<tr>
<td>route-target import 100:1</td>
<td>!</td>
</tr>
<tr>
<td>ip cef</td>
<td>interface Loopback0</td>
</tr>
<tr>
<td>mpls ldp router-id Loopback0 force</td>
<td>ip address 10.0.0.9 255.255.255.255</td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td>!</td>
</tr>
<tr>
<td>! interface Loopback0</td>
<td>interface GigabitEthernet0/0/0</td>
</tr>
<tr>
<td>ip address 192.0.0.1 255.255.255.255</td>
<td>ip vrf forwarding vpn1</td>
</tr>
<tr>
<td>! interface GigabitEthernet0/0/1</td>
<td>ip address 192.0.2.3 255.255.255.0</td>
</tr>
<tr>
<td>ip address 192.0.2.2 255.255.255.0</td>
<td>no cdp enable</td>
</tr>
<tr>
<td>mpls ip</td>
<td>!</td>
</tr>
<tr>
<td>! router ospf 100</td>
<td>! router ospf 200</td>
</tr>
<tr>
<td>network 10.0.0.0 0.0.0.0 area 100</td>
<td>bgp log-neighbor-changes</td>
</tr>
<tr>
<td>network 192.0.2.1 255.255.255.0 area 100</td>
<td>neighbor 198.51.100.2 remote-as 100</td>
</tr>
<tr>
<td>! router bgp 100</td>
<td>! address-family ipv4</td>
</tr>
<tr>
<td>no synchronization</td>
<td>redistribute connected</td>
</tr>
<tr>
<td>bgp log-neighbor changes</td>
<td>neighbor 198.51.100.2 activate</td>
</tr>
<tr>
<td>neighbor 10.0.0.3 remote-as 100</td>
<td>neighbor 198.51.100.2 advertise-interval 5</td>
</tr>
<tr>
<td>neighbor 10.0.0.3 update-source Loopback0</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>no auto-summary</td>
<td>no synchronization</td>
</tr>
<tr>
<td>! address-family vprn4</td>
<td>exit-address-family</td>
</tr>
<tr>
<td>neighbor 10.0.0.3 activate</td>
<td>! address-family ipv4 vrf vpn1</td>
</tr>
<tr>
<td>neighbor 10.0.0.3 send-community extended</td>
<td>redistribute connected</td>
</tr>
<tr>
<td>bgp scan-time import 5</td>
<td>neighbor 198.51.100.1 remote-as 200</td>
</tr>
<tr>
<td>exit-address-family</td>
<td>neighbor 198.51.100.1 activate</td>
</tr>
<tr>
<td>! address-family ipv4 vrf vpn1</td>
<td>neighbor 198.51.100.1 as-overide</td>
</tr>
<tr>
<td>redistribute connected</td>
<td>neighbor 198.51.100.1 advertisement-interval 5</td>
</tr>
<tr>
<td>neighbor 198.51.100.1 remote-as 200</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>neighbor 198.51.100.1 as-overide</td>
<td>no synchronization</td>
</tr>
<tr>
<td>neighbor 198.51.100.1 advertisement-interval 5</td>
<td>exit-address-family</td>
</tr>
</tbody>
</table>
Additional References

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>Description of commands associated with MPLS and MPLS applications</td>
<td>Cisco IOS Multiprotocol Label Switching Command Reference</td>
</tr>
<tr>
<td>Configuring MPLS virtual private networks</td>
<td>&quot;MPLS Virtual Private Networks&quot; module in the MPLS Layer 3 VPNs Configuration Guide</td>
</tr>
</tbody>
</table>

Standards and RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2283</td>
<td>Multiprotocol Extensions for BGP-4</td>
</tr>
<tr>
<td>RFC 2547</td>
<td>BGP/MPLS VPNs</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

Feature Information for Multiprotocol BGP MPLS VPN

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 2: Feature Information for Multiprotocol BGP MPLS VPN

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiprotocol BGP MPLS VPN</td>
<td>Cisco IOS XE Release 3.13.0S</td>
<td>This feature was introduced on the Cisco ASR 920 Routers (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D).</td>
</tr>
</tbody>
</table>
CHAPTER 3

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.

- Finding Feature Information, page 37
- Prerequisites for IPv6 VPN over MPLS, page 38
- Restrictions for IPv6 VPN over MPLS, page 38
- Information About IPv6 VPN over MPLS, page 38
- How to Configure IPv6 VPN over MPLS, page 45
- Configuration Examples for IPv6 VPN over MPLS, page 88
- Additional References, page 93
- Feature Information for IPv6 VPN over MPLS, page 94
- Glossary, page 94

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.
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.

**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. RRs 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. `address-family ipv4 [mdt | multicast | tunnel | unicast [vrf vrf-name] | vrf vrf-name]`
7. `route-target {import | export | both} route-target-ext-community`
8. `exit`
9. `address-family ipv6 [vrf vrf-name] [unicast | multicast]`
10. `route-target {import | export | both} route-target-ext-community`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures a VPN VRF routing table and enters VRF configuration mode.</td>
</tr>
<tr>
<td><code>vrf definition vrf-name</code></td>
<td>Configures a VPN VRF routing table and enters VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# vrf definition vrf1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the RD for a VRF.</td>
</tr>
<tr>
<td><code>rd route-distinguisher</code></td>
<td>Specifies the RD for a VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-vrf)# rd 100:1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Specifies the route target VPN extended communities for both IPv4 and IPv6.</td>
</tr>
<tr>
<td>`route-target {import</td>
<td>export</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-vrf)# route target import 100:10</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>address-family ipv4 [mdt</td>
<td>multicast</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# address-family ipv4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Specifies the route target VPN extended communities specific to IPv4.</td>
</tr>
<tr>
<td>route-target {import</td>
<td>export</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-vrf-af)# route target import 100:11</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Exits address family configuration mode on this VRF.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-vrf-af)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.</td>
</tr>
<tr>
<td>address-family ipv6 [vrf vrf-name] [unicast</td>
<td>multicast]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-vrf)# address-family ipv6</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Specifies the route target VPN extended communities specific to IPv6.</td>
</tr>
<tr>
<td>route-target {import</td>
<td>export</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-vrf-af)# route target import 100:12</td>
<td></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type number</td>
<td>Specifies an interface type and number, and places the device in interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Device(config)# interface GigabitEthernet 0/0/0</td>
</tr>
<tr>
<td>Step 4</td>
<td>vrf forwarding vrf-name</td>
<td>Associates a VPN VRF with an interface or subinterface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Device(config-if)# vrf forwarding vrf1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Note that any address, IPv4 or IPv6, that was configured prior to entering this command will be removed.</td>
</tr>
<tr>
<td>Step 5</td>
<td>ip address ip-address mask [secondary]</td>
<td>Configures an IPv4 address on the interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Device(config-if)# ip address 10.10.10.1 255.255.255.0</td>
</tr>
<tr>
<td>Step 6</td>
<td>ipv6 address {ipv6-address / prefix-length</td>
<td>prefix-name sub-bits/prefix-length}</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Device(config-if)# ipv6 address 2001:BB8:100::1/64</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 ipv6 route [vrf vrf-name] ipv6-prefix / prefix-length {ipv6-address</td>
<td>interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1</td>
</tr>
<tr>
<td>Example: Device(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:200::1 nexthop-vrf default</td>
<td></td>
</tr>
</tbody>
</table>
# 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td><code>router bgp autonomous-system-number</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config)# router bgp 100</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enters address family configuration mode.</td>
</tr>
<tr>
<td>`address-family ipv6 [vrf vrf-name] [unicast</td>
<td>multicast]`</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config-router)# address-family ipv6 vrf vrf1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Adds an entry to the multiprotocol BGP neighbor table.</td>
</tr>
<tr>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config-router-af)# neighbor 2001:DB8:100:1::2 remote-as 200</code></td>
<td></td>
</tr>
</tbody>
</table>
### 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**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>router bgp autonomous-system-number</code></td>
<td>Configures the BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# router bgp 100</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Adds an entry to the multiprotocol Border Gateway Protocol (BGP) neighbor table.</td>
<td></td>
</tr>
<tr>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} remote-as as-number</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# neighbor 192.168.2.11 remote-as 100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| **Step 5** | Enables the BGP session to use a source address on the specified interface. |
| neighbor {ip-address | ipv6-address | peer-group-name} update-source interface-type interface-number | |
| Example: | |
| Device(config-router)# neighbor 192.168.2.11 update-source Loopback 0 | |

| **Step 6** | Places the device in address family configuration mode for configuring routing sessions. |
| address-family vpnv6 [unicast] | |
| Example: | |
| Device(config-router)# address-family vpnv6 | |

| **Step 7** | Enables the exchange of information for this address family with the specified BGP neighbor. |
| neighbor {ip-address | ipv6-address | peer-group-name} activate | |
| Example: | |
| Device(config-router-af)# neighbor 192.168.2.11 activate | |

| **Step 8** | Specifies that a communities attribute should be sent to the BGP neighbor. |
| neighbor {ip-address | ipv6-address | peer-group-name} send-community [both | standard | extended] | |
| Example: | |
| Device(config-router-af)# neighbor 192.168.2.11 send-community extended | |

| **Step 9** | Exits address family configuration mode. |
| extended | exit | |
| Example: | |
| Device(config-router-af)# exit | |

**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 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 [ttl]
13. address-family ipv6
14. neighbor {ip-address | ipv6-address} activate
15. neighbor {ip-address | ipv6-address} send-label
16. neighbor {ip-address | ipv6-address | peer-group-name} route-reflector-client
17. neighbor {ip-address | ipv6-address | peer-group-name} 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 | 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 vpnv6 [unicast
25. neighbor {ip-address | ipv6-address | peer-group-name} 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

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>enable</em></td>
<td>Enables privileged EXEC mode.</td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>route bgp autonomous-system-number</td>
<td>Configures the BGP routing process.</td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td>Device(config)# router bgp 100</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} remote-as as-number</td>
</tr>
<tr>
<td><em>Example:</em></td>
<td>Device(config-router)# neighbor 192.168.2.101 remote-as 100</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} update-source interface-type interface-number</td>
</tr>
<tr>
<td><em>Example:</em></td>
<td>Device(config-router)# neighbor 192.168.2.101 update-source Loopback 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} remote-as as-number</td>
</tr>
<tr>
<td><em>Example:</em></td>
<td>Device(config-router)# neighbor 192.168.2.121 remote-as 100</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} update-source interface-type interface-number</td>
</tr>
<tr>
<td><em>Example:</em></td>
<td>Device(config-router)# neighbor 192.168.2.121 update-source Loopback 0</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>8</td>
<td><strong>neighbor</strong> `ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# neighbor 192.168.2.127 remote-as 100</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><strong>neighbor</strong> `ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# neighbor 192.168.2.127 update-source Loopback 0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>neighbor</strong> `ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# neighbor 192.168.2.1 remote-as 200</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>neighbor</strong> `ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# neighbor 192.168.2.1 update-source Loopback 0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><strong>neighbor</strong> `ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# neighbor 192.168.2.1 ebgp-multihop</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><code>address-family ipv6</code></td>
<td>(Optional) Enters address family configuration mode in order to provide Internet access service.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# address-family ipv6</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><strong>neighbor</strong> `ip-address</td>
<td>peer-group-name</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.101 activate</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong> neighbor { ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} send-label</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# neighbor 192.168.2.101 send-label</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong> neighbor {ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} route-reflector-client</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# neighbor 192.168.2.101 route-reflector-client</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong> neighbor {ip-address</td>
<td>peer-group-name</td>
<td>ipv6-address} activate</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# neighbor 192.168.2.121 activate</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong> neighbor { ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} send-label</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# neighbor 192.168.2.121 send-label</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 19</strong> neighbor {ip-address</td>
<td>ipv6-address</td>
<td>peer-group-name} route-reflector-client</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# neighbor 192.168.2.121 route-reflector-client</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 20</strong> neighbor {ip-address</td>
<td>peer-group-name</td>
<td>ipv6-address} activate</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# neighbor 192.168.2.127 activate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>21</td>
<td>`neighbor { ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.127 send-label</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.127 route-reflector-client</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td><code>exit</code></td>
<td>(Optional) Exits address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# exit</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td><code>address-family vpnv6 [unicast]</code></td>
<td>Places the device in address family configuration mode for configuring routing sessions.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# address-family vpnv6</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>`neighbor {ip-address</td>
<td>peer-group-name</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.121 activate</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.21 send-community extended</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.121 route-reflector-client</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>28</td>
<td>neighbor {ip-address</td>
<td>peer-group-name</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.127 activate</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.127 send-community extended</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.127 route-reflector-client</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>neighbor {ip-address</td>
<td>peer-group-name</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.1 activate</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.1 send-community extended</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.1 route-reflector-client</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>router bgp autonomous-system-number</code></td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# router bgp 100</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# neighbor 192.168.2.127 remote-as 100</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# neighbor 192.168.2.127 update-source Loopback 0</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>address-family ipv6</code></td>
<td>Enters address family configuration mode in order to exchange global table reachability.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# address-family ipv6</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router-af)# neighbor 192.168.2.127 activate</td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>router bgp autonomous-system-number</td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# router bgp 100</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>address-family ipv6</td>
<td>Enters address family configuration mode in order to exchange global table reachability.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-router)# address-family ipv6</td>
<td></td>
</tr>
</tbody>
</table>
### 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**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>3</td>
<td><strong>router bgp</strong> autonomous-system-number</td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# router bgp 100</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>neighbor {ip-address | ipv6-address | peer-group-name} remote-as as-number</td>
<td>Adds an entry to the multiprotocol BGP neighbor table, and provides peering with PE (PE-VPN).</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-router)# neighbor FE80::300::1 GigabitEthernet0/0/0 remote-as 300</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>address-family ipv6</td>
<td>Enters address family configuration mode in order to exchange global table reachability.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-router)# address-family ipv6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>neighbor {ip-address | ipv6-address | peer-group-name} activate</td>
<td>Enables the exchange of information for this address family with the specified BGP neighbor.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-router-af)# neighbor FE80::300::1 GigabitEthernet0/0/0 activate</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>aggregate-address address mask [as-set] [summary-only] [suppress-map map-name] [advertise-map map-name] [attribute-map map-name]</td>
<td>Creates an aggregate prefix before advertising it to the Internet.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-router-af)# aggregate-address 2001:DB8::/32 summary-only</td>
<td></td>
</tr>
</tbody>
</table>
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]] [nexthop-vrf [vrf-name] | default] [administrative-distance] [administrative-multicast-distance | unicast | multicast] [next-hop-address] [tag tag]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device&gt; enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>`ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length [ipv6-address</td>
<td>interface-type interface-number [ipv6-address]] [nexthop-vrf [vrf-name]</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:100::1 nexthop-vrf default</code></td>
<td></td>
</tr>
</tbody>
</table>
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-name | default]] [administrative-distance] [administrative-multicast-distance | unicast | multicast] [next-hop-address] [tag tag]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>* Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length [ipv6-address</td>
<td>interface-type interface-number [ipv6-address]] [nexthop-vrf [vrf-name</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# ipv6 route 2001:DB8::1000::/64 nexthop-vrf vrf1</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp <em>autonomous-system-number</em></td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# router bgp 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-router)# neighbor 192.168.2.101 remote-as 100</td>
<td></td>
</tr>
</tbody>
</table>
### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>router bgp autonomous-system-number</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# router bgp 100</td>
</tr>
<tr>
<td></td>
<td>Configures the BGP routing process.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# neighbor 192.168.2.115 remote-as 100</td>
</tr>
<tr>
<td></td>
<td>Adds an entry to the multiprotocol BGP neighbor table for peering with the route reflector with interautonomous-system functionality.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# neighbor 192.168.2.115 update-source Loopback 0</td>
</tr>
<tr>
<td></td>
<td>Enables the BGP session to use a source address on the specified interface.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>address-family vpnv6 [unicast]</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router)# address-family vpnv6</td>
</tr>
<tr>
<td></td>
<td>(Optional) Places the device in address family configuration mode for configuring routing sessions.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>**neighbor {ip-address</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router-af)# neighbor 192.168.2.115 activate</td>
</tr>
<tr>
<td></td>
<td>Enables the exchange of information for this address family with the specified BGP neighbor.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>**neighbor {ip-address</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router-af)# neighbor 192.168.2.115 send-community extended</td>
</tr>
<tr>
<td></td>
<td>Specifies that a communities attribute should be sent to the BGP neighbor.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router-af)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits address family configuration mode.</td>
</tr>
</tbody>
</table>

### 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 RR1.

### 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`
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 1    | enable           | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| 2    | configure terminal | Enters global configuration mode. |
| 3    | router bgp autonomous-system-number | Configures the BGP routing process. |
| 4    | address-family ipv4 [mdt | multicast | tunnel | unicast  
  [vrf vrf-name] | vrf vrf-name] | Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes. |
| 5    | neighbor {ip-address | ipv6-address | peer-group-name} activate | Enables the exchange of information for this address family with the specified BGP neighbor. |
| 6    | neighbor {ip-address | ipv6-address | peer-group-name} 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>router bgp autonomous-system-number</td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# router bgp 100</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>`neighbor {ip-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# neighbor 192.168.2.115 remote-as 100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# neighbor 192.168.2.115 update-source Loopback 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>address-family vpnv6 [unicast]</code></td>
<td>(Optional) Places the device in address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# address-family vpnv6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router-af)# neighbor 192.168.2.115 activate</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router-af)# neighbor 192.168.2.115 send-community extended</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>exit</code></td>
<td>Exits address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router-af)# exit</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>`address-family ipv4 [ndt</td>
<td>multicast</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# address-family ipv4</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong></td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router-af)# neighbor 192.168.2.115 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router-af)# neighbor 192.168.2.115 send-label</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>exit</td>
<td>Exits address family configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router-af)# exit</td>
<td></td>
</tr>
</tbody>
</table>

## 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device&gt; enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>router bgp autonomous-system-number</code></td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device(config)# router bgp 100</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device(config-router)# neighbor 192.168.2.127 remote-as 100</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device(config-router)# neighbor 192.168.2.127 update-source Loopback 0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>address-family vpnv6 [unicast]</code></td>
<td>(Optional) Places the device in address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device(config-router)# address-family vpnv6</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device(config-router-af)# neighbor 192.168.2.127 activate</code></td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 8    | `neighbor [ip-address | ipv6-address | peer-group-name]`  
|      | `send-community [both | standard | extended]` | Specifies that a community attribute should be sent to the BGP neighbor. |
|      | Example:  
|      | `Device(config-router-af)# neighbor 192.168.2.127`  
|      | `send-community extended` | |
| 9    | `neighbor [ip-address | ipv6-address | peer-group-name]`  
|      | `route-reflector-client` | Configures the specified neighbor as a route reflector client. |
|      | Example:  
|      | `Device(config-router-af)# neighbor 192.168.2.127`  
|      | `route-reflector-client` | |
| 10   | `exit` | Exits address family configuration mode. |
|      | Example:  
|      | `Device(config-router-af)# exit` | |
| 11   | `address-family ipv4 [mdt | multicast | tunnel | unicast [vrf`  
|      | `vrf-name]` | Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes. |
|      | `vrf vrf-name]` |  
|      | Example:  
|      | `Device(config-router)# address-family ipv4` | |
| 12   | `neighbor [ip-address | ipv6-address | peer-group-name]`  
|      | `activate` | Enables the exchange of information for this address family with the specified neighbor. |
|      | Example:  
|      | `Device(config-router-af)# neighbor 192.168.2.127`  
|      | `activate` | |
| 13   | `neighbor [ip-address | ipv6-address | peer-group-name]`  
|      | `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 LSP. |
|      | Example:  
|      | `Device(config-router-af)# neighbor 192.168.2.127`  
|      | `send-label` | |
| 14   | `exit` | Exits address family configuration mode. |
|      | Example:  
|      | `Device(config-router-af)# exit` | |
## 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} neighbor {ip-address | ipv6-address | peer-group-name} activate`
7. `neighbor {ip-address | ipv6-address | peer-group-name} send-label`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  • Enter your password if prompted.  |
| **Example:**  
  Device> enable | |
| **Step 2** configure terminal | Enters global configuration mode.  |
| **Example:**  
  Device# configure terminal | |
| **Step 3** router bgp autonomous-system-number | Configures the Border Gateway Protocol (BGP) routing process.  |
| **Example:**  
  Device(config)# router bgp 100 | |
| **Step 4** neighbor {ip-address | ipv6-address | peer-group-name} remote-as as-number | Adds an entry to the multiprotocol BGP neighbor table for peering with the ASBR1.  |
| **Example:**  
  Device(config-router)# neighbor 192.168.2.102 remote-as 100 | |
| **Step 5** neighbor {ip-address | ipv6-address | peer-group-name} update-source interface-type interface-number | Enables the BGP session to use a source address on the specified interface.  |
| **Example:**  
  Device(config-router)# neighbor 192.168.2.102 update-source Loopback 0 | |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong></td>
<td>Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router)# address-family ipv4</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Enables the exchange of information for this address family with the specified BGP neighbor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router-af)# neighbor 192.168.2.102 activate</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-router-af)# neighbor 192.168.2.102 send-label</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | **enable** | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
|   | Example: | Device> enable |
| Step 2 | **configure terminal** | Enters global configuration mode. |
|   | Example: | Device# configure terminal |
| Step 3 | **router bgp** `autonomous-system-number` | Configures the Border Gateway Protocol (BGP) routing process. |
|   | Example: | Device(config)# router bgp 100 |
| Step 4 | **neighbor** `{ip-address | ipv6-address | peer-group-name}` 
  `remote-as` `as-number` | Adds an entry to the multiprotocol BGP neighbor table for external BGP (eBGP) peering with RR2. |
|   | Example: | Device(config-router)# neighbor 192.168.2.1 remote-as 100 |
| Step 5 | **neighbor** `{ip-address | ipv6-address | peer-group-name}` 
  `update-source` `interface-type` `interface-number` | Enables the BGP session to use a source address on the specified interface. |
|   | Example: | Device(config-router)# neighbor 192.168.2.1 update-source Loopback 0 |
| Step 6 | **neighbor** `{ip-address | ipv6-address | peer-group-name}` 
  `ebgp-multihop` `[ttl]` | (Optional) Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. |
<p>|   | Example: | Device(config-router)# neighbor 192.168.2.1 ebgp-multihop |
| Step 7 | <strong>address-family vpnv6</strong> <code>[unicast]</code> | (Optional) Places the device in address family configuration mode for configuring routing sessions. |
|   | Example: | Device(config-router)# address-family vpnv6 |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong> neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-router-af)# neighbor 192.168.2.1 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-router-af)# neighbor 192.168.2.1 send-community extended</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths</td>
<td></td>
</tr>
</tbody>
</table>

### 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 [ndt | 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; <code>enable</code></td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# <code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>router bgp autonomous-system-number</code></td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# <code>router bgp 100</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# <code>neighbor 192.168.2.115 remote-as 100</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# <code>neighbor 192.168.2.115 update-source Loopback 0</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>`address-family ipv4 [mdt</td>
<td>multicast</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# <code>address-family ipv4</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router-af)# <code>neighbor 192.168.2.115 activate</code></td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>`neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
</tbody>
</table>
**IPv6 VPN over MPLS**

**Configuring a Multiautonomous-System Backbone for IPv6 VPN**

### Purpose

**Command or Action** | **Purpose**  
--- | ---  
Example:  
Device(config-router-af)# neighbor 192.168.2.115 send-label | 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
| enable | Enables privileged EXEC mode.  
**Example:**  
Device> enable |  
- Enter your password if prompted.  
| **Step 2** | Enters global configuration mode. |
| configure terminal | Enters global configuration mode.  
**Example:**  
Device# configure terminal |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>router bgp <em>autonomous-system-number</em></td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
</tbody>
</table>
|      | **Example:**  
|      | Device(config)# router bgp 100 | |
| 4    | neighbor {ip-address | ipv6-address | peer-group-name}  
|      | remote-as as-number | Adds an entry to the multiprotocol BGP neighbor table for peering with the ASBR2. |
|      | **Example:**  
|      | Device(config-router)# neighbor 192.168.3.1  
|      | remote-as 100 | |
| 5    | neighbor {ip-address | ipv6-address | peer-group-name}  
|      | update-source interface-type interface-number | Enables the BGP session to use a source address on the specified interface. |
|      | **Example:**  
|      | Device(config-router)# neighbor 192.168.3.1  
|      | update-source Loopback 0 | |
| 6    | neighbor {ip-address | ipv6-address | peer-group-name}  
|      | ebgp-multihop [ttl] | Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. |
|      | **Example:**  
|      | Device(config-router)# neighbor 192.168.3.1  
|      | ebgp-multihop | |
| 7    | address-family ipv4 [mdt | multicast | tunnel | unicast  
|      | [vrf vrf-name] | vrf vrf-name] | Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes. |
|      | **Example:**  
|      | Device(config-router)# address-family ipv4 | |
| 8    | neighbor {ip-address | ipv6-address | peer-group-name}  
|      | activate | Enables the exchange of information for this address family with the specified BGP neighbor. |
|      | **Example:**  
|      | Device(config-router-af)# neighbor 192.168.3.1  
|      | activate | |
| 9    | neighbor {ip-address | ipv6-address | peer-group-name}  
|      | send-label | Enables label exchange for this address family to this neighbor in order to receive the remote provider edge (PE) IPv4 loopback with a label in order to set up an end-to-end label switch path (LSP). |
|      | **Example:**  
|      | Device(config-router-af)# neighbor 192.168.3.1  
|      | send-label | |
### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong></td>
</tr>
<tr>
<td>Example:</td>
<td><code>Device&gt; enable</code></td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>configure terminal</td>
</tr>
<tr>
<td></td>
<td>Example: Device# configure terminal</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hostname name</td>
<td>Specifies or modifies the host name for the network server.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# hostname CSC-PE1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>router bgp autonomous-system-number</td>
<td>Configures the Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# router bgp 100</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>address-family ipv6 [vrf vrf-name] [unicast</td>
<td>multicast]</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-router)# address-family ipv6 vrf ISP2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-router-af)# neighbor FE80::866C:99 GigabitEthernet0/0/0 remote-as 200</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-router-af)# neighbor FE80::866C:99 GigabitEthernet0/0/0 activate</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>neighbor {ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-router-af)# neighbor FE80::866C:99 GigabitEthernet0/0/0 send-label</td>
<td></td>
</tr>
</tbody>
</table>
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
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 rrtinfo 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 16:26:22 4

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
  * 12001:DB8:100::/48 ::FFFF:192.168.2.101 0 100 0 10000 ?
  *>l ::FFFF:192.168.2.101 0 100 0 10000 7
  * 12001:DB8::1/128 ::FFFF:192.168.2.101 0 100 0 1
  *>l ::ffff:192.168.2.101 0 100 0 1

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 inter-area, 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]
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%
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::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8::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::/128, filtered at the autonomous system edge. Each IPv6 CE configures 2001:DB8::/128, and announces it as part as its less-specific prefix (2001:DB8::/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 ::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

```
 Device 01:DB8:100::/48
  nexthop 172.20.25.1 GigabitEtherent0/0/0 label 38 72
  nexthop 2001:DB8:1/128
  nexthop 172.20.25.1 GigabitEtherent0/0/0 label 38 73
  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 GigabitEtherent0/0/0 label 38, adjacency IP adj out of
  GigabitEtherent0/0/0 0289BEF0
  output chain: label 72 label 38 TAG adj out of GigabitEtherent0/0/0 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 nolabel/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

<table>
<thead>
<tr>
<th>Local Label</th>
<th>Outgoing Prefix</th>
<th>Bytes Label</th>
<th>Outgoing Next Hop</th>
<th>Switched interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Pop Label</td>
<td>192.168.2.114/32</td>
<td>0</td>
<td>GE0/0/0</td>
</tr>
<tr>
<td>17</td>
<td>26</td>
<td>192.168.2.146/32</td>
<td>0</td>
<td>GE0/0/0</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>72</td>
<td>No Label</td>
<td>2001:DB8:100::/48</td>
<td>63121</td>
<td>GE1/0/0</td>
</tr>
<tr>
<td>73</td>
<td>Aggregate</td>
<td>2001:DB8::1/128</td>
<td>24123</td>
<td></td>
</tr>
</tbody>
</table>

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 black-holing.

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
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,
B 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

IPv6 VPN over MPLS VRF

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,
B 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::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**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco Master Command List, All Releases</td>
</tr>
<tr>
<td>IPv6 addressing and connectivity</td>
<td>IPv6 Configuration Guide Library</td>
</tr>
<tr>
<td>IPv6 commands</td>
<td>Cisco IOS IPv6 Command Reference</td>
</tr>
<tr>
<td>Cisco IOS IPv6 features</td>
<td>IPv6 Feature Mapping</td>
</tr>
<tr>
<td>Configuring MPLS Layer 3 VPNs</td>
<td>&quot;MPLS Virtual Private Networks&quot; module in the MPLS Layer 3 VPNs Configuration Guide</td>
</tr>
</tbody>
</table>

**Standards and RFCs**

<table>
<thead>
<tr>
<th>Standard/RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFCs for IPv6</td>
<td>IPv6 RFCs</td>
</tr>
</tbody>
</table>
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 3: Feature Information for IPv6 VPN over MPLS

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 VPN over MPLS</td>
<td>Cisco IOS XE Release 3.13.0S</td>
<td>This feature was introduced on the Cisco ASR 920 Series Aggregation Service Routers (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D).</td>
</tr>
</tbody>
</table>

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.
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.

- Finding Feature Information, page 97
- Prerequisites for IPv6 Switching: Provider Edge Router over MPLS, page 97
- Information About IPv6 Switching: Provider Edge Router over MPLS, page 98
- How to Deploy IPv6 Switching: Provider Edge Router over MPLS, page 99
- Configuration Examples for IPv6 Switching: Provider Edge Router over MPLS, page 104
- Additional References for IPv6 Switching: Provider Edge Router over MPLS, page 107

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.

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**

![IPv6 Prefix: fe80:0:0:0:0:0:fe80::/64](image)

**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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device# configure terminal</td>
</tr>
<tr>
<td>Step 3</td>
<td><strong>ipv6 unicast-routing</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config)# ipv6 unicast-routing</td>
</tr>
<tr>
<td>Step 4</td>
<td><strong>ipv6 cef distributed</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config)# ipv6 cef distributed</td>
</tr>
<tr>
<td>Step 5</td>
<td><strong>interface type number</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config)# interface GigabitEthernet 0/0/0</td>
</tr>
<tr>
<td>Step 6</td>
<td>**ipv6 address ipv6-address / prefix-length</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-if)# ipv6 address 2001:DB8:FFFF::2/64</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
Example:  
Device> enable |
| **Step 2** configure terminal | Enters global configuration mode.  
Example:  
Device# configure terminal |
| **Step 3** router bgp as-number | Enters router configuration mode for the specified routing process.  
Example:  
Device(config)# router bgp 65000 |
| **Step 4** no bgp default ipv4-unicast | Disables the IPv4 unicast address family for the BGP routing process specified in the previous step.  
Example:  
Device(config-router)# no bgp default ipv4-unicast |
| **Step 5** neighbor {ip-address | ipv6-address | peer-group-name} remote-as as-number | Adds the IP address of the neighbor in the specified autonomous system to the BGP neighbor table of the local device.  
Example:  
Device(config-router)# neighbor 192.168.99.70 remote-as 65000 |

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.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong></td>
<td>neighbor { ip-address</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>address-family ipv6 [unicast]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Enters global configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>configure terminal</code></td>
<td>Device# configure terminal</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Enters router configuration mode for the specified routing process.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>router bgp as-number</code></td>
<td>Device(config)# router bgp 65000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Specifies the IPv6 address family and enters address family configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>address-family ipv6 [unicast]</code></td>
<td>Device(config-router)# address-family ipv6</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• The <code>unicast</code> keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the <code>unicast</code> keyword is not specified with the <code>address-family ipv6</code> command.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Controls the maximum number of parallel IBGP routes that can be installed in a routing table.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>maximum-paths ibgp number-of-paths</code></td>
<td>Device(config-router)# maximum-paths ibgp 3</td>
</tr>
</tbody>
</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. If IPv6 packets are generated in the local device, the IPv6 address for MPLS processing will be the address of loopback interface 0.

In the following example, serial interface 0/0 connects to the customer and the IPv6 prefix delegated to the customer is 2001:DB8:ffff::/48, which is determined from the service provider IPv6 prefix. A static route is configured to route IPv6 packets between the 6P route and the CE device.

```plaintext
ip cef
ipv6 cef
ipv6 unicast-routing
!
mpls ipv6 source-interface Loopback0
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 192.168.99.5 255.255.255.255
ipv6 address 2001:DB8:1000:1::1/64
!
interface GigabitEthernet0/0/0
description to_P_router
 ip address 192.168.99.1 255.255.255.252
 ip router isis
tag-switching ip
!
interface GigabitEthernet0/1/0
description to_CE_router
no ip address
ipv6 address 2001:DB8:FFFF::1/64
!
routing isis
 passive-interface Loopback0
 net 49.0001.1921.6809.9005.00
!
routing bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.99.70 remote-as 65000
 neighbor 192.168.99.70 description to_6PE2
 neighbor 192.168.99.70 update-source Loopback0
!
 address-family ipv6
 neighbor 192.168.99.70 activate
 neighbor 192.168.99.70 send-label
 network 2001:DB8:FFFF::/48
 exit-address-family
!
ipv6 route 2001:DB8:FFFF::/48 GigabitEthernet0/0/0 2001:DB8:FFFF::2
```

**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 00:00:00, 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
```

<table>
<thead>
<tr>
<th>Local Label</th>
<th>Outgoing Prefix</th>
<th>Bytes Label</th>
<th>Outgoing Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 Pop Label</td>
<td>1.1.1.1/32</td>
<td>0</td>
<td>Et0/0 10.0.0.1</td>
</tr>
<tr>
<td>18 No Label</td>
<td>nh-id(1)</td>
<td>0</td>
<td>Et2/0 10.0.2.2</td>
</tr>
<tr>
<td>19 No Label</td>
<td>nh-id(2)</td>
<td>0</td>
<td>Et1/0 10.0.1.2</td>
</tr>
<tr>
<td>20 No Label</td>
<td>nh-id(3)</td>
<td>0</td>
<td>Et1/0 10.0.1.2</td>
</tr>
<tr>
<td>22 No Label</td>
<td>nh-id(5)</td>
<td>0</td>
<td>Et2/0 10.0.2.2</td>
</tr>
</tbody>
</table>

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:

```
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
  next hop ::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.

```
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::/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

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 addressing and connectivity</td>
<td>IPv6 Configuration Guide</td>
</tr>
</tbody>
</table>
### Related Topic

<table>
<thead>
<tr>
<th>Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Master Commands List, All Releases</td>
</tr>
<tr>
<td>IPv6 commands</td>
<td>IPv6 Command Reference</td>
</tr>
<tr>
<td>Cisco IOS IPv6 features</td>
<td>IPv6 Feature Mapping</td>
</tr>
</tbody>
</table>

### Standards and RFCs

<table>
<thead>
<tr>
<th>Standard/RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFCs for IPv6</td>
<td>IPv6 RFCs</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
<tr>
<td>download documentation, software, and tools. Use these resources to install</td>
<td></td>
</tr>
<tr>
<td>and configure the software and to troubleshoot and resolve technical</td>
<td></td>
</tr>
<tr>
<td>issues with Cisco products and technologies. Access to most tools on the</td>
<td></td>
</tr>
<tr>
<td>Cisco Support and Documentation website requires a Cisco.com user ID and</td>
<td></td>
</tr>
<tr>
<td>password.</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 5

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.

- Finding Feature Information, page 109
- Prerequisites for Multi-VRF Support, page 109
- Restrictions for Multi-VRF Support, page 110
- Information About Multi-VRF Support, page 110
- How to Configure Multi-VRF Support, page 112
- Configuration Examples for Multi-VRF Support, page 121
- Additional References, page 125
- Feature Information for Multi-VRF Support, page 125

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.

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 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.
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.

The following are the supported flavors of multicast over VRF on Cisco ASR 920 RSP2 module:

• Multicast with multi-VRF (MPLS VPN/MLDP)
• Multicast with GRE tunnel (MVPN GRE)
• Multicast with VRF-lite
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip routing</td>
<td>Enables IP routing.</td>
</tr>
<tr>
<td>Example: Device(config)# ip routing</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip vrf vrf-name</td>
<td>Names the VRF, and enters VRF configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# ip vrf v1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> rd route-distinguisher</td>
<td>Creates a VRF table by specifying a route distinguisher.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-vrf)# rd 100:1</td>
<td>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).</td>
</tr>
<tr>
<td><strong>Step 6</strong> route-target {export</td>
<td>import</td>
</tr>
<tr>
<td><strong>Step 7</strong> import map route-map</td>
<td>(Optional) Associates a route map with the VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-vrf)# import map importmap1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> exit</td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-vrf)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> interface type slot/subslot/port[,subinterface]</td>
<td>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 BDI.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# interface GigabitEthernet 0/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> ip vrf forwarding vrf-name</td>
<td>Associates the VRF with the Layer 3 interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-if)# ip vrf forwarding v1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-if)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> show ip vrf</td>
<td>Displays the settings of the VRFs.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show ip vrf</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

<p>| Step 2            | Enters global configuration mode. |
| configure terminal |         |
| Example:          |         |
| Device# configure terminal | |</p>
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>router bgp <strong>autonomous-system-number</strong></td>
<td>Configures the BGP routing process with the autonomous system number passed to other BGP devices, and enters router configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config)# router bgp 100</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>network <strong>ip-address mask network-mask</strong></td>
<td>Specifies a network and mask to announce using BGP.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# network 10.0.0.0 mask 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>redistribute ospf <strong>process-id match internal</strong></td>
<td>Sets the device to redistribute OSPF internal routes.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# redistribute ospf 2 match internal</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>network <strong>ip-address wildcard-mask area area-id</strong></td>
<td>Identifies the network address and mask on which OSPF is running, and the area ID of that network address.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# network 10.0.0.0 255.255.255.0 area 0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>address-family ipv4 vrf <strong>vrf-name</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# address-family ipv4 vrf v12</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>neighbor **{ip-address</td>
<td>peer-group-name} remote-as as-number**</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router)# neighbor 10.0.0.3 remote-as 100</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>neighbor <strong>address activate</strong></td>
<td>Activates the advertisement of the IPv4 address-family neighbors.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-router-af)# neighbor 10.0.0.3 activate</td>
<td></td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
  enable
devic>
  enable | Enables privileged EXEC mode.  
• Enter your password if prompted. |
| **Step 2**
  configure terminal
  devic> | Enters global configuration mode. |
| **Step 3**
  router bgp autonomous-system-number
  devic> | Configures the BGP routing process with the autonomous system number passed to other BGP devices and enters router configuration mode. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td><code>address-family ipv4 vrf vrf-name</code></td>
<td>Identifies the name of the VRF instance that will be associated with the next two commands and enters address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router)# address-family ipv4 vrf v12</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>neighbor address send-label</code></td>
<td>Enables the device to use BGP to distribute MPLS labels along with the IPv4 routes to the peer devices.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router-af)# neighbor 10.0.0.3 send-label</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>neighbor address activate</code></td>
<td>Activates the advertisement of the IPv4 address-family neighbors.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router-af)# neighbor 10.0.0.3 activate</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><code>end</code></td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-router-af)# end</code></td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td><code>interface type slot/subslot/port[,subinterface]</code></td>
<td>Enters interface configuration mode for the interface to be used for the BGP session. The interface can be a routed port or an BDI.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config)# interface GigabitEthernet 0/0/0</code></td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td><code>mpls bgp forwarding</code></td>
<td>Enables MPLS forwarding on the interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-if)# mpls bgp forwarding</code></td>
<td></td>
</tr>
</tbody>
</table>
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.

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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables OSPF routing, specifies a virtual routing and forwarding (VRF) table, and enters router configuration mode.</td>
</tr>
<tr>
<td>router ospf process-id [vrf vpn-name]</td>
<td>Enables OSPF routing, specifies a virtual routing and forwarding (VRF) table, and enters router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# router ospf 100 vrf v1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>(Optional) Logs changes in the adjacency state.</td>
</tr>
<tr>
<td>log-adjacency-changes</td>
<td>(Optional) Logs changes in the adjacency state.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# log-adjacency-changes</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Sets the device to redistribute information from the Border Gateway Protocol (BGP) network to the OSPF network.</td>
</tr>
<tr>
<td>redistribute bgp autonomous-system-number subnets</td>
<td>Sets the device to redistribute information from the Border Gateway Protocol (BGP) network to the OSPF network.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# redistribute bgp 800 subnets</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Indicates the network address and mask on which OSPF runs, and the area ID of that network address.</td>
</tr>
<tr>
<td>network ip-address subnet-mask area area-id</td>
<td>Indicates the network address and mask on which OSPF runs, and the area ID of that network address.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# network 10.0.0.0 255.255.255.0 area 0</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-router)# end</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Displays information about the OSPF routing processes.</td>
</tr>
<tr>
<td>show ip ospf</td>
<td>Displays information about the OSPF routing processes.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# show ip ospf</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td><code>Device&gt;</code> <code>enable</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>interface type slot/subslot/port[/subinterface]</code></td>
<td>Enters interface configuration mode for the interface associated with the VRF. The interface can be a routed port or an BDI.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config)# interface GigabitEthernet 0/0/0</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>mpls ip</code></td>
<td>Enables MPLS forwarding of IPv4 packets along normally routed paths for this interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-if)# mpls ip</code></td>
<td></td>
</tr>
</tbody>
</table>

### 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 GigabitEthernet3/0/0
  service instance 10 ethernet
  encapsulation dot1q 10
  rewrite ingress tag pop 1 symmetric
  bridge-domain 10
!

interface GigabitEthernet3/0/0
  service instance 20 ethernet
  encapsulation dot1q 20
  rewrite ingress tag pop 1 symmetric
  bridge-domain 20
```

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 GigabitEthernet3/0/0
  service instance 10 ethernet
  encapsulation dot1q 10
  rewrite ingress tag pop 1 symmetric
  bridge-domain 10
!```
Example: Configuring Multi-VRF Support on the CE Device

The following example shows how to configure VRFs:

```
cfg-term
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 GigabitEthernet 3/8/0
  ip vrf forwarding v11
  ip address 10.0.0.8 255.255.255.0
exit
interface GigabitEthernet 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
```

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:

```
routing 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 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 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

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco Master Command List, All Releases</td>
</tr>
<tr>
<td>MPLS and MPLS applications commands</td>
<td>Cisco IOS Multiprotocol Label Switching Command Reference</td>
</tr>
<tr>
<td>OSPF with Multi-VRF</td>
<td>“OSPF Support for Multi-VRF in CE Routers” module in the OSPF Configuration Guide.</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

Feature Information for Multi-VRF Support

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 4: Feature Information for Multi-VRF Support

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-VRF Support</td>
<td>Cisco IOS XE Release 3.13.0S</td>
<td>This feature was introduced on the Cisco ASR 920 Series Aggregation Service Routers (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D).</td>
</tr>
</tbody>
</table>
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**

- Finding Feature Information, page 128
- Restrictions for ECMP Load Balancing, page 128
- Configuring ECMP Load Balancing, page 129

**Finding Feature Information**

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**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 on the RSP3 module.
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**

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

**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 run-configuration | in platform loadbalance platform loadbalance max-paths 8
Verifying ECMP Load Balancing

Use the following commands to verify ECMP load balancing.

Building configuration...

Current configuration : 10710 bytes
!
! Last configuration change at 00:29:01 IST Sat Jan 17 2015
!
version 15.5
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
platform loadbalance max-paths 8
no platform punt-keepalive disable-kernel-core
platform bfd-debug-trace 1
platform tcam-parity-error enable
platform tcam-threshold alarm-frequency 1
platform shell
!
hostname RM-PE1
!
boot-start-marker
boot-end-marker
!
vrf definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
no aaa new-model
clock timezone IST 5 30
facility-alarm critical exceed-action shutdown
no ip routing protocol purge interface
!
ip vrf test
rd 100:100
route-target export 1000:1000
route-target import 1000:1000
!
no ip domain lookup
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback106
ip address 65.1.106.1 255.255.255.255
!
interface Loopback107
ip address 65.1.107.1 255.255.255.255
!
interface Loopback108
ip address 65.1.108.1 255.255.255.255
!
interface Loopback109
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
lp-refresh-interval 65000
spf-interval 5 50 200
pnce-interval 5 50 200
lap-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
!
mls 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