Cisco Nexus 3600 NX-OS Label Switching Configuration Guide, Release 9.2(x)

First Published: 2018-07-17

Last Modified: 2018-11-05

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

CONTENTS

PREFACE **Preface** ix

Audience ix

Document Conventions ix

Related Documentation for Cisco Nexus 3000 Series Switches x

Documentation Feedback x

Communications, Services, and Additional Information x

CHAPTER 1 **New and Changed Information** 1

New and Changed Information 1

CHAPTER 2 **Overview** 3

Licensing Requirements 3

Supported Platforms 3

CHAPTER 3 **Configuring Segment Routing** 5

Licensing Requirements 5

Overview of Segment Routing 5

Segment Routing Global Block 6

High Availability for Segment Routing 6

Guidelines and Limitations for Segment Routing 6

Overview of BGP Prefix SID 7

BGP Prefix SID Deployment Example 7

Configuring Segment Routing 8

Enabling MPLS Segment Routing 8

Enabling MPLS on an Interface 9

Configuring Prefix SID Using BGP 10
Configuring the Label Index 10
Configuring the MPLS Label Allocation 12
Configuration Example for BGP Prefix SID 13
Configuring the BGP Link State Address Family 15
Verifying the Segment Routing Configuration 16
Additional References 18
Related Documents 18

CHAPTER 4 **Configuring MPLS Layer 3 VPNs** 19
Information About MPLS Layer 3 VPNs 19
MPLS Layer 3 VPN Definition 19
How an MPLS Layer 3 VPN Works 20
Components of MPLS Layer 3 VPNs 20
Hub-and-Spoke Topology 21
OSPF Sham-Link Support for MPLS VPN 22
Prerequisites for MPLS Layer 3 VPNs 23
Guidelines and Limitations for MPLS Layer 3 VPNs 23
Default Settings for MPLS Layer 3 VPNs 24
Configuring MPLS Layer 3 VPNs 24
Configuring the Core Network 24
Assessing the Needs of MPLS Layer 3 VPN Customers 24
Configuring MPLS in the Core 25
Configuring Multiprotocol BGP on the PE Routers and Route Reflectors 25
Connecting the MPLS VPN Customers 27
Defining VRFs on the PE Routers to Enable Customer Connectivity 27
Configuring VRF Interfaces on PE Routers for Each VPN Customer 29
Configuring Routing Protocols Between the PE and CE Routers 30
Configuring a Hub-and-Spoke Topology 40
Configuring MPLS using Hardware Profile Command 53

CHAPTER 5 **Configuring MPLS Layer 3 VPN Label Allocation** 55
Information About MPLS L3VPN Label Allocation 55
IPv6 Label Allocation 56

Per-VRF Label Allocation Mode 56
About Labeled and Unlabeled Unicast Paths 57
Prerequisites for MPLS L3VPN Label Allocation 57
Guidelines and Limitations for MPLS L3VPN Label Allocation 57
Default Settings for MPLS L3VPN Label Allocation 58
Configuring MPLS L3VPN Label Allocation 58
Configuring Per-VRF L3VPN Label Allocation Mode 58
Allocating Labels for IPv6 Prefixes in the Default VRF 59
Enabling Sending MPLS Labels in IPv6 over an IPv4 MPLS Core Network (6PE) for iBGP Neighbors 61
Advertisement and Withdraw Rules 63
Enabling Local Label Allocation 65
Verifying MPLS L3VPN Label Allocation Configuration 67
Configuration Examples for MPLS L3VPN Label Allocation 67

CHAPTER 6 **Configuring MPLS Layer 3 VPN Load Balancing** 69
Information About MPLS Layer 3 VPN Load Balancing 69
iBGP Load Balancing 69
eBGP Load Balancing 69
Layer 3 VPN Load Balancing 70
Layer 3 VPN Load Balancing with Route Reflectors 71
Layer 2 Load Balancing Coexistence 71
BGP VPNv4 Multipath 72
BGP Cost Community 73
How the BGP Cost Community Influences the Best Path Selection Process 73
Cost Community and EIGRP PE-CE with Back-Door Links 74
Prerequisites for MPLS Layer 3 VPN Load Balancing 74
Guidelines and Limitations for MPLS Layer 3 VPN Load Balancing 74
Default Settings for MPLS Layer 3 VPN Load Balancing 75
Configuring MPLS Layer 3 VPN Load Balancing 75
Configuring BGP Load Balancing for eBGP and iBGP 75
Configuring BGPv4 Multipath 77
Configuration Examples for MPLS Layer 3 VPN Load Balancing 77
Example: MPLS Layer 3 VPN Load Balancing 77

Example: BGP VPNv4 Multipath 78
Example: MPLS Layer 3 VPN Cost Community 78

CHAPTER 7 **InterAS Option B** 79
Information About InterAS 79
InterAS and ASBR 79
Exchanging VPN Routing Information 80
InterAS Options 80
Guidelines and Limitations for Configuring InterAS Option B 81
Configuring the Switch for InterAS Option B 81
Configuring BGP for InterAS Option B 83
Configuring the Switch for InterAS Option B (with RFC 3107 implementation) 85
Configuring BGP for InterAS Option B (with RFC 3107 implementation) 86
Creating an ACL to filter LDP connections between the ASBRs (RFC 3107 implementation) 89
Configuring InterAS Option B (lite Version) 91
Configuring the Switch for InterAS Option B (lite version) 91
Configuring BGP for InterAS Option B (lite Version) 93
Verifying InterAS Option B Configuration 94
Configuration Examples for Configuring InterAS Option B 95

Preface

This preface includes the following sections:

- Audience, on page ix
- Document Conventions, on page ix
- Related Documentation for Cisco Nexus 3000 Series Switches, on page x
- Documentation Feedback, on page x
- Communications, Services, and Additional Information, on page x

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 3000 Series Switches

The entire Cisco Nexus 3000 Series switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3600 Series NX-OS Label Switching Configuration Guide, Release 9.2(x)*.

• New and Changed Information, on page 1

New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 3600 Series NX-OS Label Switching Configuration Guide, Release 9.2(x)* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 9.2(x)

Feature	Description	Changed in Release	Where Documented
Local label allocation	Added support for IPv4 and IPv6 labeled and unlabeled unicast route on a single BGP session. This behavior is the same irrespective of whether one or both SAFI-1 and SAFI-4 are enabled on the same session or not.	9.2(2)	About Labeled and Unlabeled Unicast Paths, on page 57 Advertisement and Withdraw Rules, on page 63 Enabling Local Label Allocation, on page 65
No updates since Cisco NX-OS Release 7.x	First 9.2(x) release	Not applicable	Not applicable

CHAPTER 2

Overview

- Licensing Requirements, on page 3
- Supported Platforms, on page 3

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

Supported Platforms

Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

CHAPTER 3

Configuring Segment Routing

This chapter contains information on how to configure segment routing.

- Licensing Requirements, on page 5
- Overview of Segment Routing, on page 5
- Guidelines and Limitations for Segment Routing, on page 6
- Overview of BGP Prefix SID, on page 7
- Configuring Segment Routing, on page 8
- Verifying the Segment Routing Configuration, on page 16
- Additional References, on page 18

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

Overview of Segment Routing

Segment routing is a technique by which the path followed by a packet is encoded in the packet itself, similar to source routing. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with a segment routing header. Each segment is identified by a segment ID (SID) consisting of a flat unsigned 32-bit integer.

Border Gateway Protocol (BGP) segments, a subclass of segments, identify a BGP forwarding instruction. There are two groups of BGP segments: prefix segments and adjacency segments. Prefix segments steer packets along the shortest path to the destination, using all available equal-cost multi-path (ECMP) paths.

Border Gateway Protocol - Link State (BGP-LS) is an extension to BGP for distributing the network's Link-State (LS) topology model to external entities. BGP-LS advertise routing updates only when they occur which uses bandwidth more effectively. They advertise only the incremental change to all routers as a multicast update. They use variable length subnet masks, which are scalable and use addressing more efficiently.

The segment routing architecture is applied directly to the MPLS data plane.

Segment Routing Global Block

The segment routing global block (SRGB) is the range of local labels reserved for MPLS segment routing. The default label range is from 16000 to 23999.

SRGB is the local property of a segment routing node. Each node can be configured with a different SRGB value, and hence the absolute SID value associated to a prefix segment can change from node to node.

The SRGB must be a proper subset of the dynamic label range and must not overlap the optional MPLS static label range. If dynamic labels in the configured or defaulted SRGB range already have been allocated, the configuration is accepted, and the existing dynamic labels that fall in the SRGB range will remain allocated to the original client. A change to the SRGB range results in the clients deallocating their labels independent of whether the new range can be allocated.

High Availability for Segment Routing

In-service software upgrades (ISSUs) are minimally supported graceful restart. During the graceful restart period, the previously learned route and label state are retained.

Guidelines and Limitations for Segment Routing

Segment routing has the following guidelines and limitations:

- MPLS Segment Routing can be enabled on physical ethernet interfaces and port-channel bundles. It is not supported on ethernet sub-interfaces or Switchedx Virtual Interfaces (SVI).
- BGP allocates a SRGB label for iBGP route-reflector clients only when next-hop-self is in effect (for example, the prefix is advertised with the next hop being one of the local IP/IPv6 addresses on RR). When you have configured next-hop-self on a RR, the next hop is changed for the routes that are being affected (subject to route-map filtering).
- Static MPLS, MPLS segment routing, and MPLS stripping cannot be enabled at the same time.
- Because static MPLS, MPLS segment routing, and MPLS stripping are mutually exclusive, the only segment routing underlay for multi-hop BGP is single-hop BGP. iBGP multi-hop topologies with eBGP running as an overlay are not supported.
- MPLS pop followed by a forward to a specific interface is not supported. The penultimate hop pop (PHP) is avoided by installing the Explicit NULL label as the out-label in the label FIB (LFIB) even when the control plane installs an IPv4 Implicit NULL label.
- BGP labeled unicast and BGP segment routing are not supported for IPv6 prefixes.
- BGP labeled unicast and BGP segment routing are not supported over tunnel interfaces (including GRE and VXLAN) or with vPC access interfaces.
- MTU path discovery (RFC 2923) is not supported over MPLS label switched paths (LSPs) or segment routed paths.
- The BGP configuration commands **neighbor-down fib-accelerate** and **suppress-fib-pending** are not supported for MPLS prefixes.

- Reconfiguration of the segment routing global block (SRGB) results in an automatic restart of the BGP process to update the existing URIB and ULIB entries. Traffic loss occurs for a few seconds, so you should not reconfigure the SRGB in production.
- When the segment routing global block (SRGB) is set to a range but the route-map label-index delta value falls outside the configured range, the allocated label is dynamically generated. For example, if the SRGB is set to a range of 16000-23999 but a route-map label-index is set to 9000, the label is dynamically allocated.
- For network scalability, Cisco recommends using a hierarchical routing design with multi-hop BGP for advertising the attached prefixes from a top-of-rack (TOR) or border leaf switch.
- BGP sessions are not supported over MPLS LSPs or segment routed paths.
- The Layer 3 forwarding consistency checker is not supported for MPLS routes.

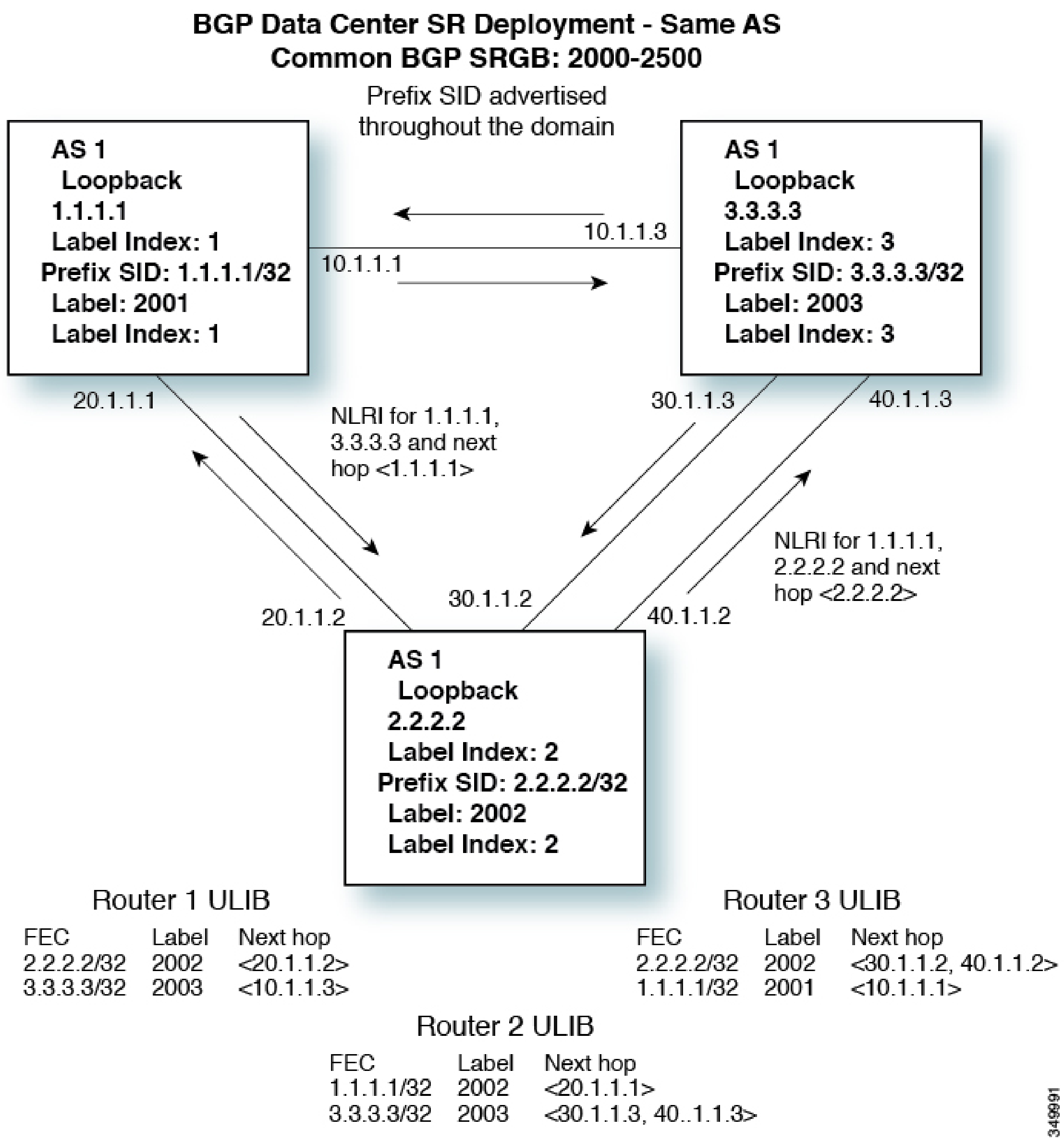
Overview of BGP Prefix SID

In order to support segment routing, BGP requires the ability to advertise a segment identifier (SID) for a BGP prefix. A BGP prefix SID is always global within the segment routing BGP domain and identifies an instruction to forward the packet over the ECMP-aware best path computed by BGP to the related prefix. The BGP prefix SID identifies the BGP prefix segment.

BGP Prefix SID Deployment Example

In the simple example below, all three routers are running iBGP and advertising Network Layer Reachability Information (NRLI) to one another. The routers are also advertising their loopback interface as the next hop, which provides the ECMP between routers 2.2.2.2 and 3.3.3.3.

Figure 1: BGP Prefix SID Simple Example



Configuring Segment Routing

Enabling MPLS Segment Routing

You can enable MPLS segment routing as long as mutually-exclusive MPLS features such as static MPLS are not enabled.

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature mpls segment-routing**
3. (Optional) **show running-config | inc 'feature mpls segment-routing'**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature mpls segment-routing** **Example:** `switch(config)# feature mpls segment-routing`	Enables the MPLS segment routing feature. The **no** form of this command disables the MPLS segment routing feature.
Step 3	(Optional) **show running-config | inc 'feature mpls segment-routing'** **Example:** `switch(config)# show running-config | inc 'feature mpls segment-routing'`	Displays the status of the MPLS segment routing feature.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling MPLS on an Interface

You can enable MPLS on an interface for use with segment routing.

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. [**no**] **mpls ip forwarding**

4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters the interface configuration mode for the specified interface.

Step 3	[**no**] **mpls ip forwarding** **Example:** `switch(config-if)# mpls ip forwarding`	Enables MPLS on the specified interface. The **no** form of this command disables MPLS on the specified interface.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Prefix SID Using BGP

You can set the label index for routes that match the **network** command. Doing so causes the BGP prefix SID to be advertised for local prefixes that are configured with a route map that includes the **set label-index** command, provided the route map is specified in the **network** command that specifies the local prefix. (For more information on the **network** command, see the "Configuring Basic BGP" chapter in the Cisco Nexus 3600 Series NX-OS Unicast Routing Configuration Guide.)

Note Route-map label indexes are ignored when the route map is specified in a context other than the **network** command. Also, labels are allocated for prefixes with a route-map label index independent of whether the prefix has been configured by the **allocate-label route-map** *route-map-name* command.

Configuring the Label Index

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *map-name*
3. [**no**] **set label-index** *index*
4. **exit**
5. **router bgp** *autonomous-system-number*

6. **address-family ipv4 unicast**
7. **network** *ip-prefix* [**route-map** *map-name*]
8. (Optional) **show route-map** [*map-name*]
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map SRmap`
	`switch(config-route-map)#`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	[**no**] **set label-index** *index* **Example:** `switch(config-route-map)# set label-index 10`	Sets the label index for routes that match the **network** command. The range is from 0 to 471788. By default, a label index is not added to the route.

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	**router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	Required: **address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the IPv4 address family.

Step 7	**network** *ip-prefix* [**route-map** *map-name*]
	Example:
	`switch(config-router-af)# network 10.10.10.10/32`
	`route-map SRmap`
		Specifies a network as local to this autonomous system and adds it to the BGP routing table.

Step 8	(Optional) **show route-map** [*map-name*] **Example:** `switch(config-router-af)# show route-map`	Displays information about route maps, including the label index.

	Command or Action	Purpose
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-af)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the MPLS Label Allocation

You can configure MPLS label allocation for the IPv4 unicast address family.

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

You must enable the MPLS segment routing feature.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **router bgp** *autonomous-system-number*
3. **address-family ipv4 unicast**
4. [**no**] **allocate-label** {**all** | **route-map** *route-map-name*}
5. **exit**
6. **neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
7. **address-family ipv4 labeled-unicast**
8. (Optional) **show bgp ipv4 labeled-unicast** *prefix*
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the **no** option with this command to remove the BGP process and the associated configuration.

Step 3	Required: **address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the IPv4 address family.

	Command or Action	Purpose
Step 4	[**no**] **allocate-label** {**all** | **route-map** *route-map-name*} **Example:** `switch(config-router-af)# allocate-label route-map map1`	Configures local label allocation for routes matching the specified route map or for all routes advertised in this address family.

Step 5	Required: **exit**
	Example:
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		Exits global address family configuration mode.

Step 6	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	Example:
	`switch(config-router)# neighbor 10.1.1.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address and AS number for a remote BGP peer.

Step 7	**address-family ipv4 labeled-unicast**
	Example:
	`switch(config-router-neighbor)# address-family ipv4 labeled-unicast`
	`switch(config-router-neighbor-af)#`
		Advertises the labeled IPv4 unicast routes as specified in RFC 3107.

Step 8	(Optional) **show bgp ipv4 labeled-unicast** *prefix* **Example:** `switch(config-router-neighbor-af)# show bgp ipv4 labeled-unicast 10.10.10.10/32`	Displays the advertised label index and the selected local label for the specified IPv4 prefix.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuration Example for BGP Prefix SID

The examples in this section show a common BGP prefix SID configuration between two routers.

This example shows how to advertise a BGP speaker configuration of 10.10.10.10/32 and 20.20.20.20/32 with a label index of 10 and 20, respectively. It uses the default segment routing global block (SRGB) range of 16000 to 23999.

```
hostname s1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing
```

```
segment-routing mpls
vlan 1

route-map label-index-10 permit 10
  set label-index 10
route-map label-index-20 permit 10
  set label-index 20

vrf context management
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 10.10.10.10/32

interface loopback2
  ip address 20.20.20.20/32

line console
line vty

router bgp 1
  address-family ipv4 unicast
    network 10.10.10.10/32 route-map label-index-10
    network 20.20.20.20/32 route-map label-index-20
    allocate-label all
  neighbor 10.1.1.2 remote-as 2
    address-family ipv4 labeled-unicast
```

This example shows how to receive the configuration from a BGP speaker.

```
hostname s2
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.2/24
  ipv6 address 10:1:1::2/64
  no shutdown

interface mgmt0
  ip address dhcp
```

```
  vrf member management

interface loopback1
  ip address 2.2.2.2/32
line console

line vty

router bgp 2
  address-family ipv4 unicast
    allocate-label all
  neighbor 10.1.1.1 remote-as 1
    address-family ipv4 labeled-unicast
```

Configuring the BGP Link State Address Family

With the introduction of RFC 7752 in Cisco Nexus software, you can configure the BGP link state address family for a neighbour session with a controller to advertise the corresponding SIDs. You can configure this feature in global configuration mode and neighbour address family configuration mode.

Before you begin

You must enable BGP.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. [**no**] **address-family link-state**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**router bgp** *autonomous-system-number* **Example:** `switch(config)# router bgp 64497`	
Step 3	[**no**] **address-family link-state** **Example:** `switch (config-router af)# address-family link-state`	Configures the BGP router. **Note** This command can also be configured in neighbour address-family configuration mode. Enters address-family interface configuration mode. **Note** This command can also be configured in neighbour address-family configuration mode.

Verifying the Segment Routing Configuration

To display the segment routing configuration, perform one of the following tasks:

Command	Purpose
show mpls switching	Displays an overview of learned label to prefix to interface mappings.
show bgp ipv4 labeled-unicast *prefix*	Displays the advertised label index and the selected local label for the specified IPv4 prefix.
show bgp link-state *prefix*	Displays the link state of one BGP address family NLRI.
show bgp link-state unicast	Displays all of the BGP address family link-state NLRIs.
show bgp paths	Displays the BGP path information, including the advertised label index.
show bgp {**ipv4** | **ipv6**} **unicast** [*ip-address* | *ipv6-prefix*] **neighbors** [**vrf** *vrf-name*]	Displays information for the BGP peers, including whether egress engineering is enabled and any peer adjacency SIDs.
show mpls label range	Displays the configured SRGB range of labels.
show route-map [*map-name*]	Displays information about a route map, including the label index.
show running-config | inc 'feature mpls segment-routing'	Displays the status of the MPLS segment routing feature.
show {**ip route** | **forwarding** }**vrf** [*vrf-name*]	Displays information about routing and forwarding.

This example shows an overview of learned label to prefix to interface mappings:

```
switch# show mpls switching
Legend:
(P)=Protected, (F)=FRR active, (*)=more labels in stack.

IPV4:
In-Label   Out-Label  FEC name           Out-Interface      Next-Hop
VRF default
16010      Pop Label  1.1.1.1/32         Eth1/11            60.0.0.1
16020      16020      2.2.2.2/32         Eth1/11            60.0.0.1
16050      16050      30.0.0.0/24        Eth1/11            60.0.0.1

In-Label   VRF
492287     default
```

This example shows how to display the configuration from a BGP speaker. The **show bgp ipv4 labeled-unicast** command in this example displays the prefix 10.10.10.10 with label index 10 mapping to label 16010 in the SRGB range of 16000 to 23999.

```
switch# show bgp ipv4 labeled-unicast 10.10.10.10/32
```

```
BGP routing table information for VRF default, address family IPv4 Label Unicast
BGP routing table entry for 10.10.10.10/32, version 7
Paths: (1 available, best #1)
Flags: (0x20c001a) on xmit-list, is in urib, is best urib route, is in HW, , has label
  label af: version 8, (0x100002) on xmit-list
  local label: 16010

  Advertised path-id 1, Label AF advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
  AS-Path: 1 , path sourced external to AS
    10.1.1.1 (metric 0) from 10.1.1.1 (10.10.10.10)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Prefix-SID Attribute: Length: 10
        Label Index TLV: Length 7, Flags 0x0 Label Index 10

  Path-id 1 not advertised to any peer
  Label AF advertisement
  Path-id 1 not advertised to any peer
```

The following is an example of **show ip route vrf 2** command.

```
show ip route vrf 2
IP Route Table for VRF "2"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

41.11.2.0/24, ubest/mbest: 1/0
    *via 1.1.1.9%default, [20/0], 13:26:48, bgp-2, external, tag 11 (mpls-vpn)
42.11.2.0/24, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, direct
42.11.2.1/32, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, local
```

The following is an example of **show forwarding route vrf 2** command.

```
slot  1
=======

IPv4 routes for table 2/base

------------------+----------------------------------------+---------------------+-----------------+-----------------
Prefix            | Next-hop                                | Interface           | Labels
          | Partial Install
------------------+----------------------------------------+---------------------+-----------------+-----------------
0.0.0.0/32           Drop                                      Null0
127.0.0.0/8          Drop                                      Null0
255.255.255.255/32   Receive                                   sup-eth1
*41.11.2.0/24        27.1.31.4                                 Ethernet1/3            PUSH
  30002 492529
                     27.1.32.4                                 Ethernet1/21           PUSH
  30002 492529
                     27.1.33.4                                 port-channel23         PUSH
  30002 492529
                     27.11.31.4                                Ethernet1/3.11         PUSH
  30002 492529
                     27.11.33.4                                port-channel23.11      PUSH
  30002 492529
```

```
                   37.1.53.4                                  Ethernet1/53/1        PUSH
29002 492529
                   37.1.54.4                                  Ethernet1/54/1        PUSH
29002 492529
                   37.2.53.4                                  Ethernet1/53/2        PUSH
29002 492529
                   37.2.54.4                                  Ethernet1/54/2        PUSH
29002 492529
                   80.211.11.1                                Vlan801               PUSH
30002 492529
```

Additional References

Related Documents

Related Topic	Document Title
BGP	*Cisco Nexus 3600 Series Unicast Routing Configuration Guide*

CHAPTER 4

Configuring MPLS Layer 3 VPNs

This chapter describes how to configure Multiprotocol Label Switching (MPLS) Layer 3 virtual private networks (VPNs) on Cisco Nexus 3600 Series Switches.

- Information About MPLS Layer 3 VPNs, on page 19
- Prerequisites for MPLS Layer 3 VPNs, on page 23
- Guidelines and Limitations for MPLS Layer 3 VPNs, on page 23
- Default Settings for MPLS Layer 3 VPNs, on page 24
- Configuring MPLS Layer 3 VPNs, on page 24

Information About MPLS Layer 3 VPNs

An MPLS Layer 3 VPN consists of a set of sites that are interconnected by an MPLS provider core network. At each customer site, one or more customer edge (CE) routers or Layer 2 switches attach to one or more provider edge (PE) routers. This section includes the following topics:

- MPLS Layer 3 VPN Definition
- How an MPLS Layer 3 VPN Works
- Components of MPLS Layer 3 VPNs
- Hub-and-Spoke Topology
- OSPF Sham-Link Support for MPLS VPN

MPLS Layer 3 VPN Definition

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

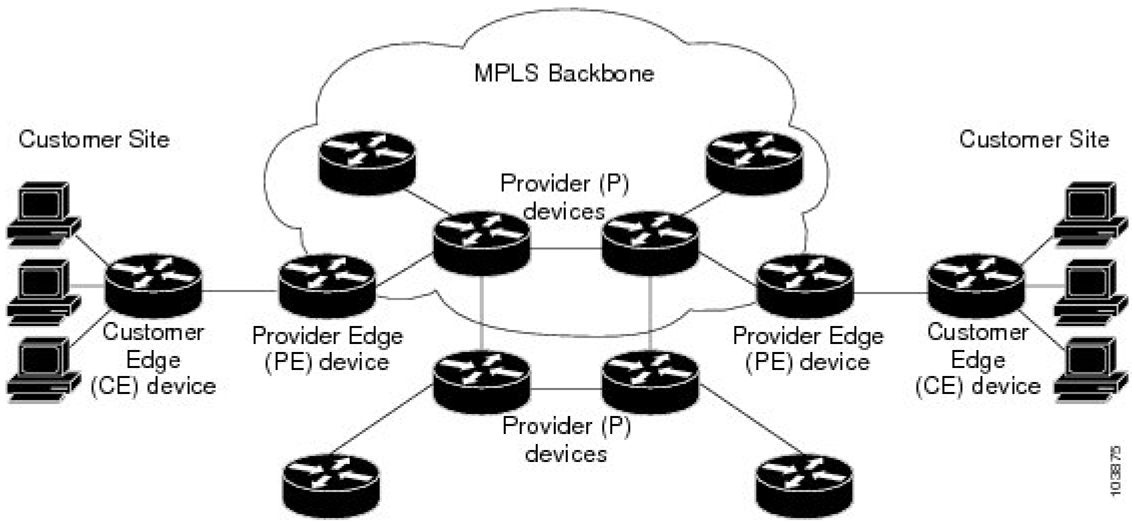
When you add a new site to an MPLS Layer 3 VPN, you must update the provider edge router that provides services to the customer site.

MPLS Layer 3 VPNs include the following components:

- Provider (P) router—A router in the core of the provider network. P routers run MPLS switching and do not attach VPN labels (an MPLS label in each route assigned by the PE router) to routed packets. P routers forward packets based on the Label Distribution Protocol (LDP).

- Provider edge (PE) router—A router that attaches the VPN label to incoming packets that are based on the interface or subinterface on which they are received. A PE router attaches directly to a CE router.
- Customer edge (CE) router—An edge router on the network of the provider that connects to the PE router on the network. A CE router must interface with a PE router.

Figure 2: Basic MPLS Layer 3 VPN Terminology



How an MPLS Layer 3 VPN Works

MPLS Layer 3 VPN functionality is enabled at the edge of an MPLS network. The PE router performs the following tasks:

- Exchanges routing updates with the CE router
- Translates the CE routing information into VPN routes
- Exchanges Layer 3 VPN routes with other PE routers through the Multiprotocol Border Gateway Protocol (MP-BGP)

Components of MPLS Layer 3 VPNs

An MPLS-based Layer 3 VPN network has three components:

1. VPN route target communities—A VPN route target community is a list of all members of a Layer 3 VPN community. You must configure the VPN route targets for each Layer 3 VPN community member.
2. Multiprotocol BGP peering of VPN community PE routers—Multiprotocol BGP propagates VRF reachability information to all members of a VPN community. You must configure Multiprotocol BGP peering in all PE routers within a VPN community.
3. MPLS forwarding—MPLS transports all traffic between all VPN community members across a VPN enterprise or service provider network.

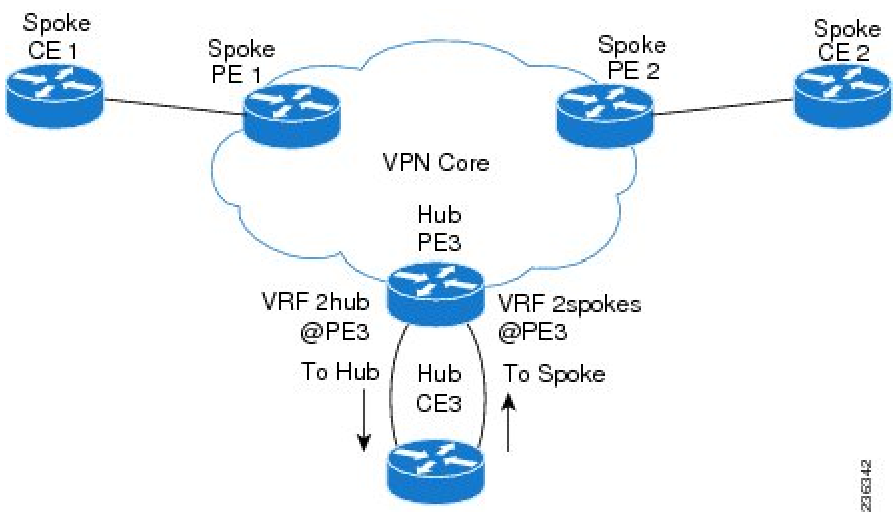
A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A customer-site VRF contains all the routes that are available to the site from the VPNs of which it is a member.

Hub-and-Spoke Topology

A hub-and-spoke topology prevents local connectivity between subscribers at the spoke provider edge (PE) routers and ensures that a hub site provides subscriber connectivity. Any sites that connect to the same PE router must forward intersite traffic using the hub site. This topology ensures that the routing at the spoke sites moves from the access-side interface to the network-side interface or from the network-side interface to the access-side interface but never from the access-side interface to the access-side interface. A hub-and-spoke topology allows you to maintain access restrictions between sites.

A hub-and-spoke topology prevents situations where the PE router locally switches the spokes without passing the traffic through the hub site. This topology prevents subscribers from directly connecting to each other. A hub-and-spoke topology does not require one VRF for each spoke.

Figure 3: Hub-and-Spoke Topology



As shown in the figure, a hub-and-spoke topology is typically set up with a hub PE that is configured with two VRFs:

- VRF 2hub with a dedicated link connected to the hub customer edge (CE)
- VRF 2spokes with another dedicated link connected to the hub CE.

Interior Gateway Protocol (IGP) or external BGP (eBGP) sessions are usually set up through the hub PE-CE links. The VRF 2hub imports all the exported route targets from all the spoke PEs. The hub CE learns all routes from the spoke sites and readvertises them back to the VRF 2spoke of the hub PE. The VRF 2spoke exports all these routes to the spoke PEs.

If you use eBGP between the hub PE and hub CE, you must allow duplicate autonomous system (AS) numbers in the path which is normally prohibited. You can configure the router to allow this duplicate AS number at the neighbor of VRF 2spokes of the hub PE and also for VPN address family neighbors at all the spoke PEs. In addition, you must disable the peer AS number check at the hub CE when distributing routes to the neighbor at VRF 2spokes of the hub PE.

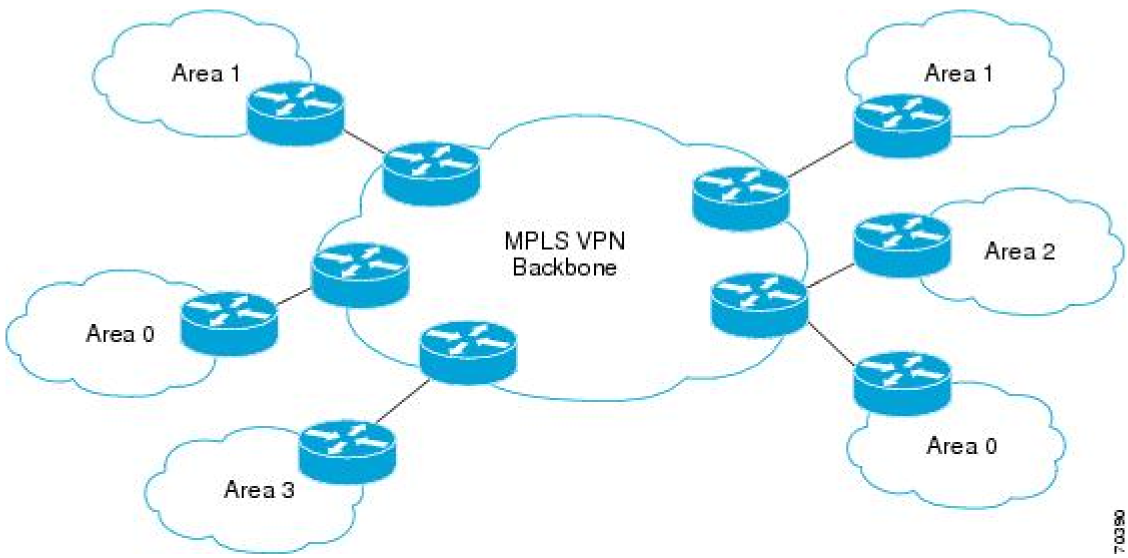
OSPF Sham-Link Support for MPLS VPN

In a Multiprotocol Label Switching (MPLS) VPN configuration, you can use the Open Shortest Path First (OSPF) protocol to connect customer edge (CE) devices to service provider edge (PE) devices in the VPN backbone. Many customers run OSPF as their intrasite routing protocol, subscribe to a VPN service, and want to exchange routing information between their sites using OSPF (during migration or on a permanent basis) over an MPLS VPN backbone.

The benefits of the OSPF sham-link support for MPLS VPN are as follows:

- Client site connection across the MPLS VPN Backbone—A sham link ensures that OSPF client sites that share a backdoor link can communicate over the MPLS VPN backbone and participate in VPN services.
- Flexible routing in an MPLS VPN configuration—In an MPLS VPN configuration, the OSPF cost tha is configured with a sham link allows you to decide if OSPF client site traffic is routed over a backdoor link or through the VPN backbone.

The figure below shows an example of how VPN client sites that run OSPF can connect over an MPLS VPN backbone.



When you use OSPF to connect PE and CE devices, all routing information learned from a VPN site is placed in the VPN routing and forwarding (VRF) instance that is associated with the incoming interface. The PE devices that attach to the VPN use the Border Gateway Protocol (BGP) to distribute VPN routes to each other. A CE device can learn the routes to other sites in the VPN by peering with its attached PE device. The MPLS VPN super backbone provides an additional level of routing hierarchy to interconnect the VPN sites that are running OSPF.

When OSPF routes are propagated over the MPLS VPN backbone, additional information about the prefix in the form of BGP extended communities (route type, domain ID extended communities) is appended to the BGP update. This community information is used by the receiving PE device to decide the type of link-state advertisement (LSA) to be generated when the BGP route is redistributed to the OSPF PE-CE process. In this way, internal OSPF routes that belong to the same VPN and are advertised over the VPN backbone are seen as interarea routes on the remote sites.

Prerequisites for MPLS Layer 3 VPNs

MPLS Layer 3 VPNs has the following prerequisites:

- Ensure that you have configured MPLS and Label Distribution Protocol (LDP) in your network. All routers in the core, including the PE routers, must be able to support MPLS forwarding.
- Ensure that you have installed the correct license for MPLS and any other features you will be using with MPLS.

Guidelines and Limitations for MPLS Layer 3 VPNs

MPLS Layer 3 VPNs have the following configuration guidelines and limitations:

- You can configure MPLS Layer 3 VPN (LDP) on Cisco Nexus 3600-R and Cisco Nexus 9504 and 9508 platform switches with the N9K-X9636C-RX, N9K-X9636C-R, N9K-X96136YC-R, and N9K-X9636Q-R line cards.
- You must enable MPLS IP forwarding on interfaces where the forwarding decisions are made based on the labels of incoming packets. If a VPN label is allocated by per prefix mode, MPLS IP forwarding must be enabled on the link between PE and CE.
- Packets with MPLS Explicit-NULL may not be parsed correctly with default line card profile.
- MPLS Layer 3 VPNs support the following CE-PE routing protocols:
-
    - BGP (IPv4 and IPv6)
    - Enhanced Interior Gateway Protocol (EIGRP) (IPv4)
    - Open Shortest Path First (OSPFv2)
    - Routing Information Protocol (RIPv2)

  Set statements in an import route map are ignored.
- The BGP minimum route advertisement interval (MRAI) value for all iBGP and eBGP sessions is zero and is not configurable.
- In a high scale setup with many BGP routes getting redistributed into EIGRP, modify the EIGRP signal timer to ensure that the EIGRP convergence time is higher than the BGP convergence time. This process allows all the BGP routes to be redistributed into EIGRP, before EIGRP signals convergence.
- When OSPF is used as a protocol between PE and CE devices, the OSPF metric is preserved when routes are advertised over the VPN backbone. The metric is used on the remote PE devices to select the correct route. Do not modify the metric value when OSPF is redistributed to BGP and when BGP is redistributed to OSPF. If you modify the metric value, routing loops might occur.

Default Settings for MPLS Layer 3 VPNs

Table 2: Default MPLS Layer 3 VPN Parameters

Parameters	Default
L3VPN feature	Disabled
L3VPN SNMP notifications	Disabled
allowas-in (for a hub-and-spoke topology)	0
disable-peer-as-check (for a hub-and-spoke topology)	Disabled

Configuring MPLS Layer 3 VPNs

Configuring the Core Network

Assessing the Needs of MPLS Layer 3 VPN Customers

You can identify the core network topology so that it can best serve MPLS Layer 3 VPN customers.

- Identify the size of the network:
    - Identify the following to determine the number of routers and ports you need:
    - How many customers do you need to support?
    - How many VPNs are needed per customer?
    - How many virtual routing and forwarding instances are there for each VPN?
- Determine which routing protocols you need in the core network.
- Determine if you need MPLS VPN high availability support.

Note MPLS VPN nonstop forwarding and graceful restart are supported on select routers and Cisco NX-OS releases. You need to make sure that graceful restart for BGP and LDP is enabled.

- Configure the routing protocols in the core network.
- Determine if you need BGP load sharing and redundant paths in the MPLS Layer 3 VPN core.

Configuring MPLS in the Core

To enable MPLS on all routers in the core, you must configure a label distribution protocol. You can use either of the following as a label distribution protocol:

- MPLS Label Distribution Protocol (LDP).

Configuring Multiprotocol BGP on the PE Routers and Route Reflectors

You can configure multiprotocol BGP connectivity on the PE routers and route reflectors.

Before you begin

Ensure that graceful restart is enabled on all routers for BGP and LDP.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **install feature-set mpls**
4. **feature-set mpls**
5. **feature-set mpls l3vpn**
6. **router bgp** *as - number*
7. **router-id** *ip-address*
8. **neighbor** *ip-address***remote-as** *as-number*
9. **address-family { vpnv4 | vpnv6 } unicast**
10. **send-community extended**
11. **show bgp { vpnv4 | vpnv6 } unicast neighbors**
12. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		Installs the MPLS feature set.

	Command or Action	Purpose

Step 4	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 5	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 6	**router bgp** *as - number* **Example:** `switch(config)# router bgp 1.1`	Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 7	**router-id** *ip-address* **Example:** `switch(config-router)# router-id 192.0.2.255`	(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker. This command triggers an automatic notification and session reset for the BGP neighbor sessions. .

Step 8	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.1`
	
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 9	**address-family { vpnv4 | vpnv6 } unicast**
	Example:
	`switch(config-router-neighbor)# address-family vpnv4 unicast`
	
	`switch(config-router-neighbor-af)#`
		Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 or VPNv6 address prefixes.

Step 10	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 11	**show bgp { vpnv4 | vpnv6 } unicast neighbors** **Example:** `switch(config-router-neighbor-af)# show bgp vpnv4 unicast neighbors`	(Optional) Displays information about BGP neighbors.

	Command or Action	Purpose
Step 12	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Connecting the MPLS VPN Customers

Defining VRFs on the PE Routers to Enable Customer Connectivity

You must create VRFs on the PE routers to enable customer connectivity. You configure route targets to control which IP prefixes are imported into the customer VPN site and which IP prefixes are exported to the BGP network. You can optionally use an import or export route map to provide more fine-grained control over the IP prefixes that are imported into the customer VPN site or exported out of the VPN site. You can use a route map to filter routes that are eligible for import or export in a VRF, based on the route target extended community attributes of the route. The route map might, for example, deny access to selected routes from a community that is on the import route target list.

SUMMARY STEPS

1. **configure terminal**
2. **install feature-set mpls**
3. **feature-set mpls**
4. **feature-set mpls l3vpn**
5. **vrf context** *vrf-name*
6. **rd** *route-distinguisher*
7. **address-family { ipv4 | ipv6 } unicast**
8. **route-target** { **import** | **export** } *route-target-ext-community* **}**
9. **maximum routes** *max-routes* [**threshold** *value*] [**reinstall**]
10. **import** [**vrf default** *max-prefix*] **map** *route-map*
11. **show vrf** *vrf-name*
12. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		Installs the MPLS feature set.

	Command or Action	Purpose

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 5	**vrf context** *vrf-name*
	Example:
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 6	**rd** *route-distinguisher*
	Example:
	`switch(config-vrf)# rd 1.2:1`
	`switch(config-vrf)#`
		Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		• 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		• 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 8	**route-target** { **import** | **export** } *route-target-ext-community* **}** **Example:** `switch(config-vrf-af-ipv4)# route-target import 1.0:1`	Specifies a route-target extended community for a VRF as follows: • 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 route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended communities. You can enter the route-target-ext-community argument in either of these formats: • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3

	Command or Action	Purpose
		• 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1
Step 9	**maximum routes** *max-routes* [**threshold** *value*] [**reinstall**] **Example:** `switch(config-vrf-af-ipv4)# maximum routes 10000`	(Optional) Configures the maximum number of routes that can be stored in the VRF route table. The max-routes range is from 1 to 4294967295. The threshold value range is from 1 to 100.
Step 10	**import** [**vrf default** *max-prefix*] **map** *route-map* **Example:** `switch(config-vrf-af-ipv4)# import vrf default map vpn1-route-map`	(Optional) Configures an import policy for a VRF to import prefixes from the default VRF as follows: • The max-prefix range is from 1 to 2147483647. The default is 1000 prefixes. • The route-map argument specifies the route map to be used as an import route map for the VRF and can be any case-sensitive, alphanumeric string up to 63 characters.
Step 11	**show vrf** *vrf-name* **Example:** `switch(config-vrf-af-ipv4)# show vrf vpn1`	(Optional) Displays information about a VRF. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 12	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring VRF Interfaces on PE Routers for Each VPN Customer

You can associate a virtual routing and forwarding instance (VRF) with an interface or subinterface on the PE routers.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **vrf member** *vrf-name*
4. **show vrf** *vrf-name* **interface**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

Step 2	**interface** *type number*
	Example:
	`switch(config)# interface Ethernet 5/0`
	`switch(config-if)#`
		Specifies the interface to configure and enters interface configuration mode as follows:
		• The type argument specifies the type of interface to be configured.
		• The number argument specifies the port, connector, or interface card number.

Step 3	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vpn1`	Associates a VRF with the specified interface or subinterface. The vrf-name argument is the name assigned to a VRF.
Step 4	**show vrf** *vrf-name* **interface** **Example:** `switch(config-if)# show vrf vpn1 interface`	(Optional) Displays information about interfaces associated with a VRF. The vrf-name argument is any case-sensitive alphanumeric string up to 32 characters.
Step 5	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Routing Protocols Between the PE and CE Routers

Configuring Static or Directly Connected Routes Between the PE and CE Routers

You can configure the PE router for PE-to-CE routing sessions that use static routes.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. { **ip ipv6** } **route** *prefix nexthop*
4. **address-family { ipv4 | ipv6 } unicast**
5. **feature bgp** *as - number*
6. **router bgp** *as - number*
7. **vrf** *vrf-name*
8. **address-family { ipv4 | ipv6 } unicast**
9. **redistribute static route-map** *map-name*
10. **redistribute direct route-map** *map-name*
11. **show { ipv4 | ipv6 } route vrf** *vrf-name*
12. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vrf context** *vrf-name*
	Example:
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 3	{ **ip** **ipv6** } **route** *prefix nexthop* **Example:** `switch(config-vrf)# ip route 192.0.2.1/28 ethernet 2/1`	Defines static route parameters for every PE-to-CE session. The prefix and nexthop are as follows: • IPv4—in dotted decimal notation • IPv6—in hex format.

Step 4	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 5	**feature bgp** *as - number*
	Example:
	`switch(config-vrf-af)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 6	**router bgp** *as - number* **Example:** `switch(config)# router bgp 1.1`	Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 7	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Associates the BGP process with a VRF.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 8	**address-family { ipv4 | ipv6 } unicast** **Example:**	Specifies the IPv4 address family type and enters address family configuration mode.

	Command or Action	Purpose

	`switch(config-vrf)# address-family ipv4 unicast`
	
	`switch(config-vrf-af)#`	

Step 9	**redistribute static route-map** *map-name* **Example:** `switch(config-router-vrf-af)# redistribute static route-map StaticMap`	Redistributes static routes into BGP. The map-name can be any case-sensitive, alphanumeric string up to 63 characters.
Step 10	**redistribute direct route-map** *map-name* **Example:** `switch(config-router-vrf-af)# redistribute direct route-map StaticMap`	Redistributes directly connected routes into BGP. The map-name can be any case-sensitive, alphanumeric string up to 63 characters.
Step 11	**show { ipv4 | ipv6 } route vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show ip ipv4 route vrf vpn1`	(Optional) Displays information about routes. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 12	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

You can use eBGP to configure the PE router for PE-to-CE routing sessions.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **router bgp** *as - number*
4. **vrf** *vrf-name*
5. **neighbor** *ip-address***remote-as** *as-number*
6. **address-family { ipv4 | ipv6 } unicast**
7. **show bgp { vpnv4 | vpnv6 } unicast neighbors vrf** *vrf-name*
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.
		The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 4	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Associates the BGP process with a VRF.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 5	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation. The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 6	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard IPv4 or IPv6 address prefixes.

Step 7	**show bgp { vpnv4 | vpnv6 } unicast neighbors vrf** *vrf-name* **Example:** `switch(config-router-neighbor-af)# show bgp vpnv4 unicast neighbors`	(Optional) Displays information about BGP neighbors. The vrf-name argument is any case-sensitive alphanumeric string up to 32 characters.
Step 8	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring RIPv2 Between the PE and CE Routers

You can use RIP to configure the PE router for PE-to-CE routing sessions.

SUMMARY STEPS

1. **configure terminal**
2. **feature rip**
3. **router rip** *instance-tag*
4. **vrf** *vrf-name*
5. **address-family ipv4 unicast**
6. **redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name vrf-name*
7. **show ip rip vrf** *vrf-name*
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature rip**
	Example:
	`switch(config)# feature rip`
	`switch(config)#`
		Enables the RIP feature.

Step 3	**router rip** *instance-tag* **Example:** `switch(config)# router rip Test1`	Enables RIP and enters router configuration mode. The instance-tag can be any case-sensitive, alphanumeric string up to 20 characters.

Step 4	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Associates the RIP process with a VRF.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 5	**address-family ipv4 unicast**
	Example:
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		Specifies the address family type and enters address family configuration mode.

Step 6	**redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name vrf-name* **Example:** `switch(config-router-vrf-af)# show ip rip vrf vpn1`	Redistributes routes from one routing domain into another routing domain. The as number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. The instance-tag can be any case-sensitive alphanumeric string up to 20 characters

	Command or Action	Purpose
Step 7	**show ip rip vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show ip rip vrf vpn1`	(Optional) Displays information about RIP. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 8	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring OSPF Between the PE and CE Routers

You can use OSPFv2 to configure the PE router for PE-to-CE routing sessions. You can optionally create an OSPF sham link if you have OSPF back door links that are not part of the MPLS network.

SUMMARY STEPS

1. **configure terminal**
2. **feature ospf**
3. **router ospf** *instance-tag*
4. **vrf** *vrf-name*
5. **area** area-id **sham-link** *source-address destination-address*
6.
7. **address-family { ipv4 | ipv6 } unicast**
8. **redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name*
9. **autonomous-system** *as-number*
10.
11. **show ip egrip vrf** *vrf-name*
12. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature ospf**
	Example:
	`switch(config)# feature ospf`
	
	`switch(config)#`
		Enables the OSPF feature.

Step 3	**router ospf** *instance-tag* **Example:** `switch(config)# router ospf Test1`	Enables OSPF and enters router configuration mode. The instance-tag can be any case-sensitive, alphanumeric string up to 20 characters.

	Command or Action	Purpose

Step 4	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Enters router VRF configuration mode.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 5	**area** area-id **sham-link** *source-address destination-address* **Example:** `switch(config-router-vrf)# area 1 sham-link 10.2.1.1 10.2.1.2`	(Optional) Configures the sham link on the PE interface within a specified OSPF area and with the loopback interfaces specified by the IP addresses as endpoints. You must configure the sham link at both PE endpoints.
Step 6		

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		Specifies the address family type and enters address family configuration mode.

Step 8	**redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name* **Example:** `switch(config-router-vrf-af)# redistribute bgp 1.0 route-map BGPMap`	Redistributes BGP into the EIGRP. The autonomous system number of the BGP network is configured in this step. BGP must be redistributed into EIGRP for the CE site to accept the BGP routes that carry the EIGRP information. A metric must also be specified for the BGP network. The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Step 9	**autonomous-system** *as-number*
	Example:
	`switch(config-router-vrf-af)#`
	`autonomous-system 1.3`
		(Optional) Specifies the autonomous system number for this address family for the customer site.
		The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 10		
Step 11	**show ip egrip vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show ipv4 eigrp vrf vpn1`	(Optional) Displays information about EIGRP in this VRF. The vrf-name can be any case-sensitive, alphanumeric string up to 32 characters
Step 12	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring EIGRP Between the PE and CE Routers

You can configure the PE router to use Enhanced Interior Gateway Routing Protocol (EIGRP) between the PE and CE routers to transparently connect EIGRP customer networks through an MPLS-enabled BGP core network so that EIGRP routes are redistributed through the VPN across the BGP network as internal BGP (iBGP) routes.

Before you begin

You must configure BGP in the network core.

SUMMARY STEPS

1. **configure terminal**
2. **feature egrip**
3. **router egrip** *instance-tag*
4. **vrf** *vrf-name*
5. **address-family ipv4 unicast**
6. **redistribute** { **bgp** as-number**route-map** *map-name*
7. **show ip ospf** *instance-tag* **vrf** *vrf-name*
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature egrip**
	Example:
	`switch(config)# feature egrip`
	`switch(config)#`
		Enables the EGRIP feature.

Step 3	**router egrip** *instance-tag* **Example:** `switch(config)# router egrip Test1`	Configures an EIGRP instance and enters router configuration mode. The instance-tag can be any case-sensitive, alphanumeric string up to 20 characters.

Step 4	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Enters router VRF configuration mode.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

	Command or Action	Purpose

Step 5	**address-family ipv4 unicast**
	Example:
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		(Optional) Enters address family configuration mode for configuring routing sessions that use standard IPv4 address prefixes.

Step 6	**redistribute** { **bgp** as-number**route-map** *map-name* **Example:** `switch(config-router-vrf-af)# show ip rip vrf vpn1`	Redistributes routes from one routing domain into another routing domain. The as number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. The instance-tag can be any case-sensitive alphanumeric string up to 20 characters
Step 7	**show ip ospf** *instance-tag* **vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show ip rip vrf vpn1`	(Optional) Displays information about OSPF.
Step 8	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring PE-CE Redistribution in BGP for the MPLS VPN

You must configure BGP to distribute the PE-CE routing protocol on every PE router that provides MPLS Layer 3 VPN services if the PE-CE protocol is not BGP.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **router bgp** *instance-tag*
4. **router id** *ip-address*
5. **router id** *ip-address* **remote-as** *as-number*
6. **update-source loopback [0 | 1]**
7. **address-family { ipv4 | ipv6 } unicast**
8. **send-community extended**
9. **vrf** *vrf-name*
10. **address-family { ipv4 | ipv6 } unicast**
11. **redistribute** { **direct** | { **egrip** | **ospfv3** | **ospfv3** |**rip** } *instance-tag* | static } **route-map** *map-name*
12. **show bgp { ipv4 | ipv6 } unicast vrf** *vrf-name*
13. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**router bgp** *instance-tag*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 4	**router id** *ip-address*
	Example:
	`switch(config-router)# router-id 192.0.2.255 1`
	`switch(config-router)#`
		(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

Step 5	**router id** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table.The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 6	**update-source loopback [0 | 1]** **Example:** `switch(config-router-neighbor)# update-source loopback 0#`	Specifies the source address of the BGP session.

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-neighbor)# address-family vpnv4`
	`switch(config-router-neighbor-af)#`
		Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 or VPNv6 address prefixes. The optional unicast keyword specifies VPNv4 or VPNv6 unicast address prefixes.

Step 8	**send-community extended** **Example:**	Specifies that a communities attribute should be sent to a BGP neighbor.

	Command or Action	Purpose
	`switch(config-router-neighbor-af)# send-community extended`	

Step 9	**vrf** *vrf-name*
	Example:
	`switch(config-router-neighbor-af)# vrf vpn1`
	`switch(config-router-vrf)#`
		Enters router VRF configuration mode.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 10	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		Enters address family configuration mode for configuring routing sessions that use standard IPv4 or IPv6 address prefixes.

Step 11	**redistribute** { **direct** | { **egrip** | **ospfv3** | **ospfv3** |**rip** } *instance-tag* | static } **route-map** *map-name* **Example:** `switch(config-router-af-vrf)# redistribute eigrp Test2 route-map EigrpMap`	Redistributes routes from one routing domain into another routing domain. The as number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. The instance-tag can be any case-sensitive, alphanumeric string up to 20 characters. The map-name can be any case-sensitive alphanumeric string up to 63 characters.
Step 12	**show bgp { ipv4 | ipv6 } unicast vrf** *vrf-name* **Example:** `switch(config-router--vrf-af)# show bgp ipv4 unicast vrf vpn1vpn1`	(Optional) Displays information about BGP. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 13	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring a Hub-and-Spoke Topology

Configuring VRFs on the Hub PE Router

You can configure hub and spoke VRFs on the hub PE router.

SUMMARY STEPS

1. **configure terminal**
2. **install feature-set mpls**
3. **feature-set mpls**
4. **feature-set mpls l3vpn**
5. **vrf context** *vrf-hub*
6. **rd** *route-distinguisher*

7. **address-family { ipv4 | ipv6 } unicast**
8. **route-target** { **import** | **export** } route-target-ext-community **}**
9. **vrf context** *vrf-spoke*
10. **address-family { ipv4 | ipv6 } unicast**
11. **route-target** { **import** | **export** } *route-target-ext-community* **}**
12. **show running-config vrf** *vrf-name*
13. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		Installs the MPLS feature set.

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 5	**vrf context** *vrf-hub*
	Example:
	`switch(config)# vrf context 2hub`
	
	`switch(config-vrf)#`
		Defines the VPN routing instance for the PE hub by assigning a VRF name and enters VRF configuration mode. The vrf-hub argument is any case-sensitive alphanumeric string up to 32 characters.

Step 6	**rd** *route-distinguisher*
	Example:
	`switch(config-vrf)# rd 1.2:1`
	
	`switch(config-vrf)#`
		Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		• 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		• 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

	Command or Action	Purpose

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 8	**route-target** { **import** | **export** } route-target-ext-community **}**
	Example:
	`switch(config-vrf-af-ipv4)# route-target import 1.0:1`
		Specifies a route-target extended community for a VRF as follows:
		• 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 route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended communities. You can enter the route-target-ext-community argument in either of these formats:
		  • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		  • 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 9	**vrf context** *vrf-spoke*
	Example:
	`switch(config-vrf-af-ipv4)# vrf context 2spokes`
	`switch(config-vrf)#`
		Defines the VPN routing instance for the PE spoke by assigning a VRF name and enters VRF configuration mode. The vrf-spoke argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 10	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 11	**route-target** { **import** | **export** } *route-target-ext-community* **}** **Example:** `switch(config-vrf-af-ipv4)# route-target export 1:100`	Specifies a route-target extended community for a VRF as follows: • Creates a route-target extended community for a VRF. 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 route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended

	Command or Action	Purpose
		communities. You can enter the route-target-ext-community argument in either of these formats: • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3 • 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1
Step 12	**show running-config vrf** *vrf-name* **Example:** `switch(config-vrf-af-ipv4)# show running-config vrf 2spokes`	(Optional) Displays the running configuration for the VRF. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters. .
Step 13	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring eBGP on the Hub PE Router

You can use eBGP to configure PE-to-CE hub routing sessions.

Note If all CE sites are using the same BGP AS number, you must perform the following tasks:

- Configure either the BGP **as-override** command at the PE (hub) **or the allowas-in** command at the receiving CE router.
- To advertise BGP routes learned from one ASN back to the same ASN, configure the **disable-peer-as-check** command at the PE router to prevent loopback.

SUMMARY STEPS

1. **configure terminal**
2. **feature-set mpls**
3. **feature mpls l3vpn**
4. **feature bgp**
5. **router bgp** *as - number*
6. **neighbor** *ip-address***remote-as** *as-number*
7. **address-family { ipv4 | ipv6 } unicast**
8. **send-community extended**
9. **vrf** *vrf-hub*
10. **neighbor** *ip-address***remote-as** *as-number*
11. **address-family { ipv4 | ipv6 } unicast**

12. **as-override**
13. **vrf** *vrf-spoke*
14. **neighbor** *ip-address***remote-as** *as-number*
15. **address-family { ipv4 | ipv6 } unicast**
16. **allowas-in** [*number*]
17. **show running-config bgp** *vrf-name*
18. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature-set.
Step 3	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.

Step 4	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 5	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.
		The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 7	**address-family { ipv4 | ipv6 } unicast** **Example:**	Specifies the IP address family type and enters address family configuration mode.

	Command or Action	Purpose
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast` `switch(config-router-neighbor-af)#`	
Step 8	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	(Optional) Configures BGP to advertise extended community lists.

Step 9	**vrf** *vrf-hub*
	Example:
	`switch(config-router-neighbor-af)# vrf 2hub`
	`switch(config-router-vrf)#`
		Enters VRF configuration mode. The *vrf-hub* argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 10	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table for this VRF.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 11	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 12	**as-override** **Example:** `switch(config-router-vrf-neighbor-af)# as-override`	(Optional) Overrides the AS-number when sending an update. If all BGP sites are using the same AS number, of the following commands: • Configure the BGP as-override command at the PE (hub) or • Configure the allowas-in command at the receiving CE router.

Step 13	**vrf** *vrf-spoke*
	Example:
	`switch(config-router-vrf-neighbor-af)# vrf 2spokes`
	`switch(config-router-vrf)#`
		Enters VRF configuration mode. The vrf-spoke argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 14	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table for this VRF.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

	Command or Action	Purpose
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 15	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 16	**allowas-in** [*number*] **Example:** `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(Optional) Allows duplicate AS numbers in the AS path. Configure this parameter in the VPN address family configuration mode at the PE spokes and at the neighbor mode at the PE hub.
Step 17	**show running-config bgp** *vrf-name* **Example:** `switch(config-router-vrf-neighbor-af)# show running-config bgp`	(Optional) Displays the running configuration for BGP.
Step 18	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring eBGP on the Hub CE Router

You can use eBGP to configure PE-to-CE hub routing sessions.

Note Note If all CE sites are using the same BGP AS number, you must perform the following tasks:

- Configure either the as-override command at the PE (hub) or the allowas-in command at the receiving CE router.
- Configure the disable-peer-as-check command at the CE router.
- To advertise BGP routes learned from one ASN back to the same ASN, configure the disable-peer-as-check command at the PE router to prevent loopback.

SUMMARY STEPS

1. **configure terminal**
2. **feature-set mpls**
3. **feature mpls l3vpn**
4. **feature bgp**
5. **router bgp** *as - number*
6. **neighbor** *ip-address***remote-as** *as-number*

7. **address-family { ipv4 | ipv6 } unicast**
8. **send-community extended**
9. **vrf** *vrf-hub*
10. **neighbor** *ip-address***remote-as** *as-number*
11. **address-family { ipv4 | ipv6 } unicast**
12. **as-override**
13. **vrf** *vrf-spoke*
14. **neighbor** *ip-address***remote-as** *as-number*
15. **address-family { ipv4 | ipv6 } unicast**
16. **allowas-in** [*number*]
17. **show running-config bgp** *vrf-name*
18. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature-set.
Step 3	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.

Step 4	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 5	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.
		The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	**neighbor** *ip-address***remote-as** *as-number* **Example:**	Adds an entry to the iBGP neighbor table. • The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

	Command or Action	Purpose
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2` `switch(config-router-neighbor)#`	• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 8	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	(Optional) Configures BGP to advertise extended community lists.

Step 9	**vrf** *vrf-hub*
	Example:
	`switch(config-router-neighbor-af)# vrf 2hub`
	`switch(config-router-vrf)#`
		Enters VRF configuration mode. The *vrf-hub* argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 10	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table for this VRF.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 11	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 12	**as-override** **Example:** `switch(config-router-vrf-neighbor-af)# as-override`	(Optional) Overrides the AS-number when sending an update. If all BGP sites are using the same AS number, of the following commands: • Configure the BGP as-override command at the PE (hub) or • Configure the allowas-in command at the receiving CE router.

Step 13	**vrf** *vrf-spoke*
	Example:
	`switch(config-router-vrf-neighbor-af)# vrf 2spokes`
	`switch(config-router-vrf)#`
		Enters VRF configuration mode. The vrf-spoke argument is any case-sensitive, alphanumeric string up to 32 characters.

	Command or Action	Purpose

Step 14	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table for this VRF.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 15	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 16	**allowas-in** [*number*] **Example:** `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(Optional) Allows duplicate AS numbers in the AS path. Configure this parameter in the VPN address family configuration mode at the PE spokes and at the neighbor mode at the PE hub.
Step 17	**show running-config bgp** *vrf-name* **Example:** `switch(config-router-vrf-neighbor-af)# show running-config bgp`	(Optional) Displays the running configuration for BGP.
Step 18	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring VRFs on the Spoke PE Router

You can configure hub and spoke VRFs on the spoke PE router.

SUMMARY STEPS

1. **configure terminal**
2. **install feature-set mpls**
3. **feature-set mpls**
4. **feature-set mpls l3vpn**
5. **vrf context** *vrf-spoke*
6. **rd** *route-distinguisher*
7. **address-family { ipv4 | ipv6 } unicast**
8. **route-target** { **import** | **export** } route-target-ext-community **}**
9. **show running-config vrf** *vrf-name*
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		Installs the MPLS feature set.

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 5	**vrf context** *vrf-spoke*
	Example:
	`switch(config)# vrf context spoke`
	`switch(config-vrf)#`
		Defines the VPN routing instance for the PE spoke by assigning a VRF name and enters VRF configuration mode. The vrf-spoke argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 6	**rd** *route-distinguisher*
	Example:
	`switch(config-vrf)# rd 1.101`
	`switch(config-vrf)#`
		Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		• 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		• 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 8	**route-target** { **import** | **export** } route-target-ext-community **}** **Example:**	Specifies a route-target extended community for a VRF as follows: • The **import** keyword imports routing information from the target VPN extended community.

	Command or Action	Purpose

	`switch(config-vrf-af-ipv4)# route-target import 1.0:1`	• The **export** keyword 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 or export route-target extended communities. You can enter the route-target-ext-community argument in either of these formats:
		  • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		  • 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 9	**show running-config vrf** *vrf-name* **Example:** `switch(config-vrf-af-ipv4)# show running-config vrf 2spokes`	(Optional) Displays the running configuration for the VRF. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters. .
Step 10	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring eBGP on the Spoke PE Router

You can use eBGP to configure PE spoke routing sessions.

Note If all CE sites are using the same BGP AS number, you must perform the following tasks:

- Configure the the allowas-in command at the receiving spoke router.

SUMMARY STEPS

1. **configure terminal**
2. **feature-set mpls**
3. **feature mpls l3vpn**
4. **feature bgp**
5. **router bgp** *as - number*
6. **neighbor** *ip-address***remote-as** *as-number*
7. **address-family { ipv4 | ipv6 } unicast**
8. **allowas-in** *number*
9. **send-community extended**

10. **show running-config bgp**
11. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature-set.
Step 3	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.

Step 4	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 5	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 100`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.
		The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 63.63.0.63 remote-as 100`
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Specifies the IPv4 or IPv6 address family type and enters address family configuration mode.

	Command or Action	Purpose
Step 8	**allowas-in** *number* **Example:** `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(Optional) Allows an AS path with the PE ASN for a specified number of times. • The range is from 1 to 10 • If all BGP sites are using the same AS number, of the following commands: **Note** Configure the BGP as-override command at the PE (hub) or Configure the allowas-in command at the receiving CE router. The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 9	**send-community extended** **Example:** `switch(config-router-neighbor)# send-community extended`	(Optional) Configures BGP to advertise extended community lists.
Step 10	**show running-config bgp** **Example:** `switch(config-router-vrf-neighbor-af)# show running-config bgp`	(Optional) Displays the running configuration for BGP.
Step 11	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring MPLS using Hardware Profile Command

Beginning with release 7.0(3)F3(3), Cisco Nexus 3600 supports multiple hardware profiles. You can configure MPLS and/or VXLAN using hardware profile configuration command in a switch. The hardware profile configuration command invokes appropriate configuration files that are available on the switch. VXLAN is enabled by default

Before you begin

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **hardware profile** [**vxlan** | **mpls**] **module all**

4. **show hardware profile module** [**all** | *number*]
5. **show module internal sw info** | [**i** | **mpls**]
6. **show running configuration** | [**i** | **mpls**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**hardware profile** [**vxlan** | **mpls**] **module all** **Example:** `switch(config)# hardware profile mpls  module all`	Enables MPLS on all the switch modules. .

Step 4	**show hardware profile module** [**all** | *number*]
	Example:
	`switch(config)# show hardware profile module all`
	`switch(config)#`
		Displays the hardware profile of all the modules or specific module.

Step 5	**show module internal sw info** | [**i** | **mpls**] **Example:** `switch(config)# show module internal sw info`	Displays the switch software information.
Step 6	**show running configuration** | [**i** | **mpls**] **Example:** `switch(config)# show module internal sw info`	Displays the running configuration.

CHAPTER 5

Configuring MPLS Layer 3 VPN Label Allocation

This chapter describes how to configure label allocation for Multiprotocol Label Switching (MPLS) Layer 3 virtual private networks (L3VPNs) on Cisco Nexus 3600 series Switches.

- Information About MPLS L3VPN Label Allocation, on page 55
- Prerequisites for MPLS L3VPN Label Allocation, on page 57
- Guidelines and Limitations for MPLS L3VPN Label Allocation, on page 57
- Default Settings for MPLS L3VPN Label Allocation, on page 58
- Configuring MPLS L3VPN Label Allocation, on page 58
- Advertisement and Withdraw Rules, on page 63
- Enabling Local Label Allocation, on page 65
- Verifying MPLS L3VPN Label Allocation Configuration, on page 67
- Configuration Examples for MPLS L3VPN Label Allocation, on page 67

Information About MPLS L3VPN Label Allocation

The MPLS provider edge (PE) router stores both local and remote routes and includes a label entry for each route. By default, Cisco NX-OS uses per-prefix label allocation which means that each prefix is assigned a label. For distributed platforms, the per-prefix labels consume memory. When there are many VPN routing and forwarding instances (VRFs) and routes, the amount of memory that the per-prefix labels consume can become an issue.

You can enable per-VRF label allocation to advertise a single VPN label for local routes throughout the entire VRF. The router uses a new VPN label for the VRF decoding and IP-based lookup to learn where to forward packets for the PE or customer edge (CE) interfaces.

You can enable different label allocation modes for Border Gateway Protocol (BGP) Layer 3 VPN routes to meet different requirements and to achieve trade-offs between scalability and performance. All labels are allocated within the global label space. Cisco NX-OS supports the following label allocation modes:

- Per-prefix—A label is allocated for each VPN prefix. VPN packets received from remote PEs can be directly forwarded to the connected CE that advertised the prefix, based on the label forwarding table. However, this mode also uses many labels. This mode is the only mode available when VPN packets sent from PE to CE are label switched. This is the default label allocation mode.

- Per-VRF—A single label is assigned to all local VPN routes in a VRF. This mode requires an IPv4 or IPv6 lookup in the VRF forwarding table once the VPN label is removed at the egress PE. This mode is the most efficient in terms of label space as well as BGP advertisements, and the lookup does not result in any performance degradation. Cisco NX-OS uses the same per-VRF label for both IPv4 and IPv6 prefixes.

Note EIBGP load balancing is not supported for a VRF that uses per-VRF label mode

- Aggregate Labels—BGP can allocate and advertise a local label for an aggregate prefix. Forwarding requires an IPv4 or IPv6 lookup that is similar to the per-VRF scenario. A single per-VRF label is allocated and used for all prefixes that need a lookup.
- VRF connected routes—When directly connected routes are redistributed and exported, an aggregate label is allocated for each route. The packets that come in from the core are decapsulated and a lookup is done in the VRF IPv4 or IPv6 table to determine whether the packet is for the local router or for another router or host that is directly connected. A single per-VRF label is allocated for all such routes.
- Label hold down—When a local label is no longer associated with a prefix, to allow time for updates to be sent to other PEs, the local label is not released immediately. A ten minute hold down timer is started per label. Within this hold down period, the label can be reclaimed for the prefix. When the timer expires, BGP releases the label.

IPv6 Label Allocation

IPv6 prefixes are advertised with the allocated label to iBGP peers that have the labeled-unicast address-family enabled. The received eBGP next hop is not propagated to such peers; instead, the local IPv4 session address is sent as an IPv4-mapped IPv6 next hop. The remote peer resolves this next hop through one or more IPv4 MPLS LSPs in the core network.

You can use a route reflector to advertise the labeled 6PE prefixes between PEs. You must enable the labeled-unicast address-family between the route reflector and all such peers. The route reflector does not need to be in the forwarding path and propagates the received next hop as is to iBGP peers and route reflector clients.

Note 6PE also supports both per-prefix and per-VRF label allocation modes, as in 6VPE

Per-VRF Label Allocation Mode

The following conditions apply when you configure per-VRF label allocation:

- The VRF uses one label for all local routes.
- When you enable per-VRF label allocation, any existing per-VRF aggregate label is used. If no per-VRF aggregate label is present, the software creates a new per-VRF label.

  The CE does not lose data when you disable per-VRF label allocation because the configuration reverts to the default per-prefix labeling configuration.
- A per-VRF label forwarding entry is deleted only if the VRF, BGP, or address family configuration is removed.

About Labeled and Unlabeled Unicast Paths

Subsequent Address Family Identifier (SAFI) is an indication of the BGP route. Example 1 is for an unlabeled route and 4 for a labeled route.

- Unlabeled unicast (U) for IPv4 is SAFI 1.
- Labeled unicast (LU) for IPv4 is SAFI 4.
- Unlabeled unicast (U) for IPv6 is AFI 2 and SAFI 1.
- Labeled unicast (LU) for IPv6 is AFI 2 and SAFI 4.

Cisco NX-OS Release 9.2(2) supports both, IPv4 and IPv6 unlabeled and labeled unicast on one BGP session. This behavior is the same irrespective of whether one or both SAFI-1 and SAFI-4 are enabled on the same session or not.

This behavior is applicable for all eBGP, iBGP, and redistributed paths and the eBGP and iBGP neighbors.

Prerequisites for MPLS L3VPN Label Allocation

L3VPN label allocation has the following prerequisites:

- Ensure that you have configured MPLS, and LDP in your network. All routers in the core, including the PE routers, must be able to support MPLS forwarding.
- Ensure that you have installed the correct license for MPLS and any other features you will be using with MPLS.
- Ensure that you disable the external/internal Border Gateway Protocol (BGP) multipath feature if it is enabled before you configure per-VRF label allocation mode.
- Before configuring a 6VPE per VRF label, ensure that the IPv6 address family is configured on that VRF.

Guidelines and Limitations for MPLS L3VPN Label Allocation

L3VPN label allocation has the following configuration guidelines and limitations:

- Layer 3 VPN label allocation is also supported on the Cisco Nexus 3600 platform switches.
- Enabling per-VRF label allocation causes BGP reconvergence, which can result in data loss for traffic coming from the MPLS VPN core.

Note You can minimize network disruption by enabling per-VRF label allocation during a scheduled MPLS maintenance window. Also, if possible, avoid enabling this feature on a live router.

- Aggregate labels and per-VRF labels are global across all virtual device contexts (VDCs) and are in a separate, dedicated label range.

- Aggregate prefixes for per-prefix label allocation share the same label in a given VRF.

Default Settings for MPLS L3VPN Label Allocation

Table 3: Default L3VPN Label Allocation Parameters

Parameters	Default
L3VPN feature	Disabled
Label allocation mode	Per prefix

Configuring MPLS L3VPN Label Allocation

Configuring Per-VRF L3VPN Label Allocation Mode

You can configure per-VRF L3VPN label allocation mode for Layer 3 VPNs.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **feature-set mpls**
4. **feature-set mpls l3vpn**
5. **router bgp** *as - number*
6. **vrf** *vrf-name*
7. **address-family { ipv4 | ipv6 } unicast | multicast }**
8. **label-allocation-mode per-vrf**
9. **show bgp l3vpn detail vrf** *vrf-name*
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

	Command or Action	Purpose

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 5	**router bgp** *as - number* **Example:** `switch(config)# router bgp 1.1`	Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 6	**vrf** *vrf-name* **Example:** `switch(config-router)# vrf vpn1`	Enters router VRF configuration mode. The vrf-name can be any case-sensitive, alphanumeric string up to 32 characters..
Step 7	**address-family { ipv4 | ipv6 } unicast | multicast }** **Example:** `switch(config-router-vrf)# address-family ipv6 unicast`	Specifies the IP address family type and enters address family configuration mode.
Step 8	**label-allocation-mode per-vrf** **Example:** `switch(config-router-vrf-af)# label-allocation-mode per-vrf`	Allocates labels on a per-VRF basis.
Step 9	**show bgp l3vpn detail vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show bgp l3vpn detail vrf vpn1`	(Optional) Displays information about Layer 3 VPN configuration on BGP for this VRF. The vrf-name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 10	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Allocating Labels for IPv6 Prefixes in the Default VRF

If you are running IPv6 over an IPv4 MPLS core network (6PE), you can allocate labels for the IPv6 prefixes in the default VRF.

Note By default, labels are not allocated for IPv6 prefixes in the default VRF.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **feature-set mpls**
4. **feature-set mpls l3vpn**
5. **router bgp** *as - number*
6. **address-family { ipv4 | ipv6 } unicast | multicast }**
7. **allocate-label { all | route-map** *route-map* **}**
8. **show running-config bgp**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 5	**router bgp** *as - number* **Example:** `switch(config)# router bgp 1.1`	Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

	Command or Action	Purpose
Step 6	**address-family { ipv4 | ipv6 } unicast | multicast }** **Example:** `switch(config-router-vrf)# address-family ipv6 unicast`	Specifies the IP address family type and enters address family configuration mode.
Step 7	**allocate-label { all | route-map** *route-map* **}** **Example:** `switch(config-router-af)# allocate-label all`	Allocates labels for IPv6 prefixes in the default VRF. • The **all** keyword allocates labels for all IPv6 prefixes. • The **route-map** keyword allocates labels for IPv6 prefixes matched in the specified route map. The route-map can be any case-sensitive alphanumeric string up to 63 characters.
Step 8	**show running-config bgp** **Example:** `switch(config-router-af)# show running-config bgp`	(Optional) Displays information about the BGP configuration.
Step 9	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling Sending MPLS Labels in IPv6 over an IPv4 MPLS Core Network (6PE) for iBGP Neighbors

6PE advertises IPv6 prefixes in global VRF over IPv4 based MPLS network with the allocated label to iBGP peers that have the labeled-unicast address-family enabled. 6PE requires LDP enabled on core facing interfaces to transport IPv6 traffic over IPv4 based MPLS network and "address-family ipv6 labeled-unicast" under BGP to exchange label for IPv6 prefixes between PEs.

Note The **address-family ipv6 labeled-unicast** command is supported only for iBGP neighbors. You cannot use this command with the **address-family ipv6 unicast** command.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **feature-set mpls**
4. **feature-set mpls l3vpn**
5. **router bgp** *as - number*
6. **neighbor** *ip-address*
7. **address-family ipv6 labeled-unicast**
8. **show running-config bgp**

9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 5	**router bgp** *as - number* **Example:** `switch(config)# router bgp 1.1`	Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 7	**address-family ipv6 labeled-unicast**
	Example:
	`switch(config-router-neighbor)# address-family ipv6 labeled-unicast`
	
	`switch(config-router-neighbor-af)#`
		Specifies IPv6 labeled unicast address prefixes. This command is accepted only for iBGP neighbors.

Step 8	**show running-config bgp** **Example:** `switch(config-router-af)# show running-config bgp`	(Optional) Displays information about the BGP configuration.

	Command or Action	Purpose
Step 9	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

What to do next

-

Advertisement and Withdraw Rules

The following table shows the advertisement and withdraw behavior for different scenarios.

Table 4: Advertisement and Withdraw Rules

Case	Bestpath/ Addpath Type	Local Label Present?	NHS or NHU	Update-group SAFI	Advertise o withdraw?
1	Unlabeled path. For example, no RX label.	Yes	NHS	SAFI-1	Advertise b default.
2				SAFI-4	Advertise
3			NHU	SAFI-1	Advertise

Case	Bestpath/ Addpath Type	Local Label Present?	NHS or NHU	Update-group SAFI	Advertise or withdraw?
4				SAFI-4	Withdraw
5		No	NHS	SAFI-1	Advertise
6				SAFI-4	Withdraw
7			NHU	SAFI-1	Advertise
8				SAFI-4	Withdraw
9	Labeled path. For example, with an RX label.	Yes	NHS	SAFI-1	Advertise by default. Withdraw with NbrKnob.
10				SAFI-4	Advertise
11			NHU	SAFI-1	Withdraw
12				SAFI-4	Advertise
13		No	NHS	SAFI-1	Advertise
14				SAFI-4	Withdraw
15			NHU	SAFI-1	Withdraw

Case	Bestpath/ Addpath Type	Local Label Present?	NHS or NHU	Update-group SAFI	Advertise o withdraw?
				SAFI-4	Advertise

Enabling Local Label Allocation

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **feature-set mpls**
4. **router bgp** *as - number*
5. **address-family { ipv4 | ipv6 } unicast | multicast }**
6. **allocate-label { all | route-map** *route-map* **}**
7. **neighbor** *ip-address*
8. [**no**] **advertise local-labeled-route**
9. **address-family { ipv4 | ipv6 } unicast | multicast }**
10. [**no**] **advertise local-labeled-route**
11. **route-map label_routemap permit 10**
12. **show running-config bgp**
13. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

	Command or Action	Purpose
Step 4	**router bgp** *as - number* **Example:** `switch(config)# router bgp 1.1`	Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 5	**address-family { ipv4 | ipv6 } unicast | multicast }** **Example:** `switch(config-router-vrf)# address-family ipv4 unicast`	Specifies the IP address family type and enters the address family configuration mode.
Step 6	**allocate-label { all | route-map** *route-map* **}** **Example:** `switch(config-router-af)# allocate-label all`	Allocates labels for IPv6 prefixes in the default VRF. • The **all** keyword allocates labels for all IPv6 prefixes. • The **route-map** keyword allocates labels for IPv6 prefixes matched in the specified route map. The route-map can be any case-sensitive alphanumeric string up to 63 characters.

Step 7	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 8	[**no**] **advertise local-labeled-route** **Example:** `switch(config-router-neighbor)# advertise local-labeled-route`	Indicates whether to advertise an IPv4 or IPv6 route with a local label to the BGP neighbor via the IPv4 or IPv6 unicast SAFI (SAFI-1). The default is enabled so that it can be advertised to the BGP neighbor.
Step 9	**address-family { ipv4 | ipv6 } unicast | multicast }** **Example:** `switch(config-router-vrf)# address-family ipv6 unicast`	Specifies the IP address family type and enters the address family configuration mode.
Step 10	[**no**] **advertise local-labeled-route** **Example:** `switch(config-router-neighbor)# advertise local-labeled-route`	Indicates whether to advertise an IPv4 or IPv6 route with a local label to the BGP neighbor via the IPv4 or IPv6 unicast SAFI (SAFI-1). The default is enabled so that it can be advertised to the BGP neighbor.
Step 11	**route-map label_routemap permit 10** **Example:** `switch(config-router-vrf)# route-map label_routemap permit 10`	

	Command or Action	Purpose
Step 12	**show running-config bgp** **Example:** `switch(config-router-af)# show running-config bgp`	(Optional) Displays information about the BGP configuration.
Step 13	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying MPLS L3VPN Label Allocation Configuration

To display the L3VPN label allocation configuration, perform one of the following tasks:

Table 5: Verifying MPLS L3VPN Label Allocation Configuration

Command	Purpose
show bgp l3vpn [**detail**] [**vrf** *v rf-name*]	Displays Layer 3 VPN information for BGP in a VRF.
show bgp vpnv4 unicast labels [**vrf** *v rf-name*]	Displays label information for BGP.
show ip route [**vrf** *v rf-name*]	Displays label information for routes.

Configuration Examples for MPLS L3VPN Label Allocation

The following example shows how to configure per-VRF label allocation for an IPv4 MPLS network.

```
PE1
-----
vrf context vpn1
rd 100:1
address-family ipv4 unicast
route-target export 200:1
router bgp 100
neighbor 10.1.1.2 remote-as 100
address-family vpnv4 unicast
send-community extended
update-source loopback10
vrf vpn1
address-family ipv4 unicast
label-allocation-mode per-vrf
neighbor 36.0.0.2 remote-as 300
address-family ipv4 unicast
```

CHAPTER 6

Configuring MPLS Layer 3 VPN Load Balancing

This chapter describes how to configure load balancing for Multiprotocol Label Switching (MPLS) Layer 3 virtual private networks (VPNs) on Cisco NX-OS devices.

- Information About MPLS Layer 3 VPN Load Balancing, on page 69
- Prerequisites for MPLS Layer 3 VPN Load Balancing, on page 74
- Guidelines and Limitations for MPLS Layer 3 VPN Load Balancing, on page 74
- Default Settings for MPLS Layer 3 VPN Load Balancing, on page 75
- Configuring MPLS Layer 3 VPN Load Balancing, on page 75
- Configuration Examples for MPLS Layer 3 VPN Load Balancing, on page 77

Information About MPLS Layer 3 VPN Load Balancing

Load balancing distributes traffic so that no individual router is overburdened. In an MPLS Layer 3 network, you can achieve load balancing by using the Border Gateway Protocol (BGP). When multiple iBGP paths are installed in a routing table, a route reflector advertises only one path (next hop). If a router is behind a route reflector, all routes that are connected to multihomed sites are not advertised unless a different route distinguisher is configured for each virtual routing and forwarding instance (VRF). (A route reflector passes learned routes to neighbors so that all iBGP peers do not need to be fully meshed.)

iBGP Load Balancing

When a BGP-speaking router configured with no local policy receives multiple network layer reachability information (NLRI) from the internal BGP (iBGP) for the same destination, the router chooses one iBGP path as the best path and installs the best path in its IP routing table. iBGP load balancing enables the BGP-speaking router to select multiple iBGP paths as the best paths to a destination and to install multiple best paths in its IP routing table.

eBGP Load Balancing

When a router learns two identical eBGP paths for a prefix from a neighboring autonomous system, it chooses the path with the lower route ID as the best path. The router installs this best path in the IP routing table. You can enable eBGP load balancing to install multiple paths in the IP routing table when the eBGP paths are learned from a neighboring autonomous system instead of picking one best path.

During packet switching, depending on the switching mode, the router performs either per-packet or per-destination load balancing among the multiple paths.

Layer 3 VPN Load Balancing

ayer 3 VPN load balancing for both eBGP and iBGP allows you to configure multihomed autonomous systems and provider edge (PE) routers to distribute traffic across both external BGP (eBGP) and iBGP multipaths.

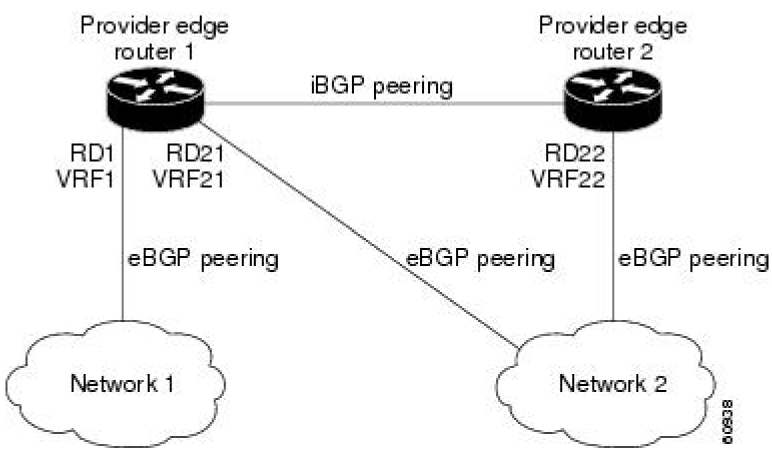
Layer 3 VPN load balancing supports IPv4 and IPv6 for the PE routers and VPNs.

BGP installs up to the maximum number of multipaths allowed. BGP uses the best path algorithm to select one path as the best path, inserts the best path into the routing information base (RIB) and advertises the best path to BGP peers. The router can insert other paths into the RIB but selects only one path as the best path.

Layer 3 VPNs load balance on a per-packet or per-source or destination pair basis. To enable load balancing, configure the router with Layer 3 VPNs that contain VPN routing and forwarding instances (VRFs) that import both eBGP and iBGP paths. You can configure the number of paths separately for each VRF.

The following figure shows an MPLS provider network that uses BGP. In the figure, two remote networks are connected to PE1 and PE2, which are both configured for VPN unicast iBGP peering. Network 2 is a multihomed network that is connected to PE1 and PE2. Network 2 also has extranet VPN services configured with Network 1. Both Network 1 and Network 2 are configured for eBGP peering with the PE routers.

Figure 4: Provider MPLS Network Using BGP



You can configure PE1 so that it can select both iBGP and eBGP paths as multipaths and import these paths into the VPN routing and forwarding instance (VRF) of Network 1 to perform load balancing.

Traffic is distributed as follows:

- IP traffic that is sent from Network 2 to PE1 and PE2 is sent across the eBGP paths as IP traffic.
- IP traffic that is sent from PE1 to PE2 is sent across the iBGP path as MPLS traffic.
- Traffic that is sent across an eBGP path is sent as IP traffic.

Any prefix that is advertised from Network 2 will be received by PE1 through route distinguisher (RD) 21 and RD22.

- The advertisement through RD21 is carried in IP packets.
- The advertisement through RD22 is carried in MPLS packets.

The router can select both paths as multipaths for VRF1 and insert these paths into the VRF1 RIB.

Layer 3 VPN Load Balancing with Route Reflectors

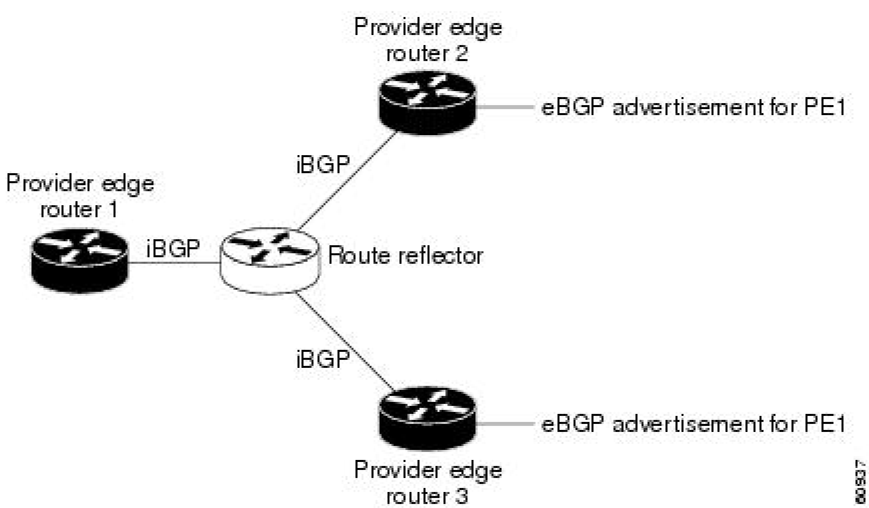
Route reflectors reduce the number of sessions on PE routers and increase the scalability of Layer 3 VPN networks. Route reflectors hold on to all received VPN routes to peer with PE routers. Different PEs can require different route target-tagged VPNv4 and VPNv6 routes. The route reflector may also need to send a refresh for a specific route target to a PE when the VRF configuration has changed. Storing all routes increases the scalability requirements on a route reflector. You can configure a route reflector to only hold routes that have a defined set of route target communities.

You can configure route reflectors to service a different set of VPNs and configure a PE to peer with all route reflectors that service the VRFs configured on the PE. When you configure a new VRF with a route target that the PE does not already hold routes for, the PE issues route refreshes to the route reflectors and retrieves the relevant VPN routes.

The following figure shows a topology that contains three PE routers and a route reflector, all configured for iBGP peering. PE2 and PE3 each advertise an equal preference eBGP path to PE1. By default, the route reflector chooses only one path and advertises PE1.

Note The route reflectors do not need to be in the forwarding path, but you must configure unique route distinguisher (RDs) for VPN sites that are multihomed.

Figure 5: Topology with a Route Reflector



For all equal preference paths to PE1 to be advertised through the route reflector, you must configure each VRF with a different RD. The prefixes received by the route reflector are recognized differently and advertised to PE1.

Layer 2 Load Balancing Coexistence

The load balance method that is required in the Layer 2 VPN is different from the method that is used for Layer 3 VPN. Layer 3 VPN and Layer 2 VPN forwarding is performed independently using two different types of adjacencies. The forwarding is not impacted by using a different method of load balancing for the Layer 2 VPN.

Note Load balancing is not supported at the ingress PE for Layer 2 VPNs

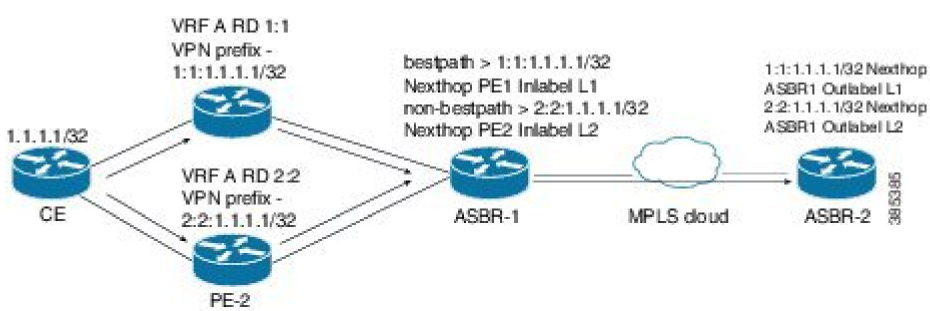
BGP VPNv4 Multipath

BGP VPNv4 Multipath feature helps to achieve Equal Cost Multi-Path (ECMP) for traffic flowing from an Autonomous System Border Router (ASBR) towards the Provider Edge (PE) device in an Multi-Protocol Label Switching (MPLS) cloud network by using a lower number of prefixes and MPLS labels. This feature configures the maximum number of multipaths for both eBGP and iBGP paths. This feature can be configured on PE devices and Route Reflectors in an MPLS topology.

Consider a scenario in which a dual homed Customer Edge (CE) device is connected to 2 PE devices and you have to utilize both the PE devices for traffic flow from ASBR-2 to the CE device.

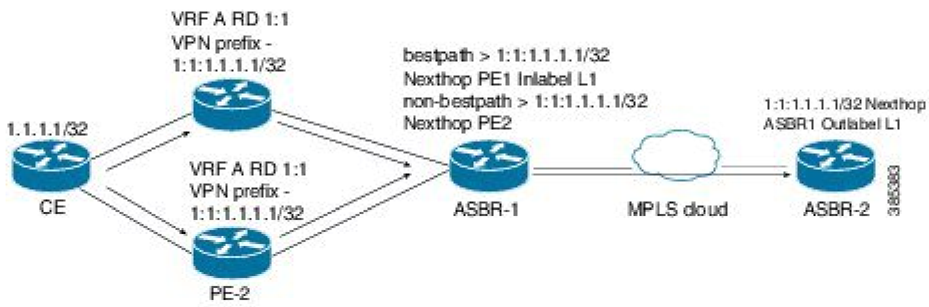
Currently, as shown in following figure, Virtual Routing and Forwarding (VRF) on each PE is configured using separate Route Distinguishers (RD). The CE device generates a BGP IPv4 prefix. The PE devices are configured with 2 separate RDs and generate two different VPN-IPv4 prefixes for the BGP IPv4 prefix sent by the CE device. ASBR-1 receives both the VPN-IPv4 prefixes and adds them to the routing table. ASBR-1 allocates Inter-AS option-B labels, Inlabel L1 and Inlabel L2, to both the VPN routes and then advertises both VPN routes to ASBR-2. To use both PE devices to maintain traffic flow, ASBR-1 has to utilize two Inter-AS option-B labels and two prefixes which limits the scale that can be supported.

Figure 6: Virtual Routing and Forwarding (VRF) on each PE configured using separate Route Distinguishers



Using the BGP VPN Multipath feature, as shown in Figure 22-4, you can enable the VRF on both PE devices to use the same RD. In such a scenario, ASBR-1 receives the same prefix from both the PE devices. ASBR-1 allocates only one Inter-AS option-B label, Inlabel L1, to the received prefix and advertises the VPN route to ASBR-2. In this case, the scale is enhanced as traffic flow using both PE devices is established with only one prefix and label on ASBR-1.

Figure 7: Enabling the VRF on both PE devices to use the same RD



BGP Cost Community

The BGP cost community is a nontransitive extended community attribute that is passed to iBGP and confederation peers but not to eBGP peers. (A confederation is a group of iBGP peers that use the same autonomous system number to communicate to external networks.) The BGP cost community attributes includes a cost community ID and a cost value. You can customize the BGP best path selection process for a local autonomous system or confederation by configuring the BGP cost community attribute. You configure the cost community attribute in a route map with a community ID and cost value. BGP prefers the path with the lowest community ID, or for identical community IDs, BGP prefers the path with the lowest cost value in the BGP cost community attribute.

BGP uses the best path selection process to determine which path is the best where multiple paths to the same destination are available. You can assign a preference to a specific path when multiple equal cost paths are available.

Since the administrative distance of iBGP is worse than the distance of most Interior Gateway Protocols (IGPs), the unicast Routing Information Base (RIB) may apply the same BGP cost community compare algorithm before using the normal distance or metric comparisons of the protocol or route. VPN routes that are learned through iBGP can be preferred over locally learned IGP routes.

The cost extended community attribute is propagated to iBGP peers when an extended community exchange is enabled.

How the BGP Cost Community Influences the Best Path Selection Process

The cost community attribute influences the BGP best path selection process at the point of insertion (POI). The POI follows the IGP metric comparison. When BGP receives multiple paths to the same destination, it uses the best path selection process to determine which path is the best path. BGP automatically makes the decision and installs the best path into the routing table. The POI allows you to assign a preference to a specific path when multiple equal cost paths are available. If the POI is not valid for local best path selection, the cost community attribute is silently ignored.

You can configure multiple paths with the cost community attribute for the same POI. The path with the lowest cost community ID is considered first. All of the cost community paths for a specific POI are considered, starting with the one with the lowest cost community ID. Paths that do not contain the cost community (for the POI and community ID being evaluated) are assigned with the default community cost value.

Applying the cost community attribute at the POI allows you to assign a value to a path originated or learned by a peer in any part of the local autonomous system or confederation. The router can use the cost community as a tie breaker during the best path selection process. You can configure multiple instances of the cost community for separate equal cost paths within the same autonomous system or confederation. For example, you can apply a lower cost community value to a specific exit path in a network with multiple equal cost exits points, and the BGP best path selection process prefers that specific exit path.

Cost Community and EIGRP PE-CE with Back-Door Links

BGP prefers back-door links in an Enhanced Interior Gateway Protocol (EIGRP) Layer 3 VPN topology if the back-door link is learned first. A back-door link, or a route, is a connection that is configured outside of the Layer 3 VPN between a remote and main site.

The pre-best path point of insertion (POI) in the BGP cost community supports mixed EIGRP Layer 3 VPN network topologies that contain VPN and back-door links. This POI is applied automatically to EIGRP routes that are redistributed into BGP. The pre-best path POI carries the EIGRP route type and metric. This POI influences the best-path calculation process by influencing BGP to consider this POI before any other comparison step.

Prerequisites for MPLS Layer 3 VPN Load Balancing

MPLS Layer 3 VPN load balancing has the following prerequisites:

- You must enable the MPLS and L3VPN features.
- You must install the correct license for MPLS.

Guidelines and Limitations for MPLS Layer 3 VPN Load Balancing

MPLS Layer 3 VPN load balancing has the following configuration guidelines and limitations:

- MPLS Layer 3 VPN load balancing is supported on Cisco Nexus 3600 platform switches.
- If you place a router behind a route reflector and it is connected to multihomed sites, the router will not be advertised unless separate VRFs with different RDs are configured for each VRF.
- Each IP routing table entry for a BGP prefix that has multiple iBGP paths uses additional memory. We recommend that you do not use this feature on a router with a low amount of available memory or when it is carrying a full Internet routing table.
- You should not ignore the BGP cost community when a back-door link is present and EIGRP is the PE-CE routing protocol.
- A maximum of 16K VPN prefixes is supported on Cisco Nexus 3600 platform switches with N3K-C3636C-R and N3K-C36180YC-R line cards.
- 4K VRFs are supported.

Default Settings for MPLS Layer 3 VPN Load Balancing

The following table lists the default settings for MPLS Layer 3 VPN load balancing parameters.

Table 6: Default MPLS Layer 3 VPN Load Balancing Parameters

Parameters	Default
Layer 3 VPN feature	Disabled
BGP cost community ID	128
BGP cost community cost	2147483647
maximum multipaths	1
BGP VPNv4 Multipath	Disabled

Configuring MPLS Layer 3 VPN Load Balancing

Configuring BGP Load Balancing for eBGP and iBGP

You can configure a Layer 3 VPN load balancing for an eBGP or iBGP network.

SUMMARY STEPS

1. **configure terminal**
2. **feature-set mpls**
3. **feature mpls l3vpn**
4. **feature bgp**
5. **router bgp** *as - number*
6. **bestpath cost-community ignore remote-as** *as-number*
7. **address-family { ipv4 | ipv6 } unicast**
8. **maximum-paths** [**bgp**] *number-of-paths*
9. **show running-config bgp**
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature-set.
Step 3	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.

Step 4	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 5	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.
		The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	**bestpath cost-community ignore remote-as** *as-number* **Example:** `switch(config-router)# bestpath cost-community ignore#`	(Optional) Ignores the cost community for BGP bestpath calculations.

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters address family configuration mode for configuring IP routing sessions.

Step 8	**maximum-paths** [**bgp**] *number-of-paths* **Example:** `switch(config-router-af)# maximum-paths 4`	Configures the maximum number of multipaths allowed. Use the ibgp keyword to configure **iBGP** load balancing. The range is from 1 to 16.
Step 9	**show running-config bgp** **Example:** `switch(config-router-vrf-neighbor-af)# show running-config bgp`	(Optional) Displays the running configuration for BGP.
Step 10	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring BGPv4 Multipath

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **router bgp** *as - number*
4. **address-family vpnv4 unicast**
5. **maximum-paths eibgp** *parallel-paths*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp** **Example:** `switch(config)# feature bgp`	Enables the BGP feature.

Step 3	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 2`
	`switch(config-router)#`
		Assigns an autonomous system (AS) number to a router and enter the router BGP configuration mode.

Step 4	**address-family vpnv4 unicast**
	Example:
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)#`
		Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.

Step 5	**maximum-paths eibgp** *parallel-paths* **Example:** `switch(config-router-af)# maximum-paths eibgp 3`	Specifies the maximum number of BGP VPNv4 multipaths for both eBGP and iBGP paths. The range is from 1 to 32.

Configuration Examples for MPLS Layer 3 VPN Load Balancing

Example: MPLS Layer 3 VPN Load Balancing

The following example shows how to configure iBGP load balancing:

```
 configure terminal
feature-set mpls
```

```
feature mpls l3vpn
feature bgp
router bgp 1.1
bestpath cost-community ignore
address-family ipv6 unicast
maximum-paths ibgp 4
```

Example: BGP VPNv4 Multipath

The following example shows how to configure a maximum of 3 BGP VPNv4 multipaths:

```
 configure terminal
router bgp 100
address-family vpnv4 unicast
maximum-paths eibgp 3
```

Example: MPLS Layer 3 VPN Cost Community

The following example shows how to configure the BGP cost community:

```
 configure terminal
feature-set mpls
feature mpls l3vpn
feature bgp
route-map CostMap permit
set extcommunity cost 1 100
router bgp 1.1
router-id 192.0.2.255
neighbor 192.0.2.1 remote-as 1.1
address-family vpnv4 unicast
send-community extended
route-map CostMap in
```

CHAPTER 7

InterAS Option B

This chapter explains the different InterAS option B configuration options. The available options are InterAS option B, InterAS option B (with RFC 3107), and InterAS option B lite. The InterAS option B (with RFC 3107) implementation ensures complete IGP isolation between the data centers and WAN. When BGP advertises a particular route to ASBR, it also distributes the label which is mapped to that route.

- Information About InterAS, on page 79
- InterAS Options, on page 80
- Guidelines and Limitations for Configuring InterAS Option B, on page 81
- Configuring the Switch for InterAS Option B, on page 81
- Configuring BGP for InterAS Option B, on page 83
- Configuring the Switch for InterAS Option B (with RFC 3107 implementation), on page 85
- Configuring BGP for InterAS Option B (with RFC 3107 implementation), on page 86
- Creating an ACL to filter LDP connections between the ASBRs (RFC 3107 implementation), on page 89
- Configuring InterAS Option B (lite Version), on page 91
- Verifying InterAS Option B Configuration, on page 94
- Configuration Examples for Configuring InterAS Option B, on page 95

Information About InterAS

An autonomous system (AS) is a single network or group of networks that is controlled by a common system administration group and using a single, clearly defined protocol. In many cases, virtual private networks (VPNs) extend to different ASes in different geographical areas. Some VPNs must extend across multiple service providers; these VPNs are called overlapping VPNs. The connection between ASes must be seamless to the customer, regardless of the complexity or location of the VPNs.

InterAS and ASBR

Separate ASes from different service providers can communicate by exchanging information in the form of VPN IP addresses. The ASBRs use EBGP to exchange that information. The IBGP distributes the network layer information for IP prefixes throughout each VPN and each AS. The following protocols are used for sharing routing information:

- Within an AS, routing information is shared using IBGP.

- Between ASes, routing information is shared using EBGP. EBGP allows service providers to set up an interdomain routing system that guarantees loop-free exchange of routing information between separate ASes.

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

InterAS configuration supported in this MPLS VPN can include an interprovider VPN, which is MPLS VPNs that include two or more ASes, connected by separate border edge routers. The ASes exchange routes use EBGP, and no IBGP or routing information is exchanged between the ASes.

Exchanging VPN Routing Information

ASes exchange VPN routing information (routes and labels) to establish connections. To control connections between ASes, the PE routers and EBGP border edge routers maintain a label forwarding information base (LFIB). The LFIB manages the labels and routes that the PE routers and EBGP border edge routers receive during the exchange of VPN information.

The ASes use the following guidelines to exchange VPN routing information:

- Routing information includes:
    - The destination network.
    - The next-hop field associated with the distributing router.
    - A local MPLS label
- A route distinguisher (RD1) is part of a destination network address. It makes the VPN IP route globally unique in the VPN service provider environment.

The ASBRs are configured to change the next-hop when sending VPN NLRIs to the IBGP neighbors. Therefore, the ASBRs must allocate a new label when they forward the NLRI to the IBGP neighbors.

InterAS Options

Nexus 3600 series switches support the following InterAS options:

- **InterAS option A** - In an interAS option A network, autonomous system border router (ASBR) peers are connected by multiple subinterfaces with at least one interface VPN that spans the two ASes. These ASBRs associate each subinterface with a VPN routing and forwarding (VRF) instance and a BGP session to signal unlabeled IP prefixes. As a result, traffic between the back-to-back VRFs is IP. In this scenario, the VPNs are isolated from each other and, because the traffic is IP Quality of Service (QoS) mechanisms that operate on the IP traffic can be maintained. The downside of this configuration is that one BGP session is required for each subinterface (and at least one subinterface is required for each VPN), which causes scalability concerns as the network grows.
- **InterAS option B** - In an interAS option B network, ASBR ports are connected by one or more subinterfaces that are enabled to receive MPLS traffic. A Multiprotocol Border Gateway Router (MP-BGP) session distributes labeled VPN prefixes between the ASBRs. As a result, the traffic that flows between the ASBRs is labeled. The downside of this configuration is that, because the traffic is MPLS, QoS mechanisms that are applied only to IP traffic cannot be carried and the VRFs cannot be isolated. InterAS

option B provides better scalability than option A because it requires only one BGP session to exchange all VPN prefixes between the ASBRs. Also, this feature provides nonstop forwarding (NSF) and Graceful Restart. The ASBRs must be directly connected in this option.

Some functions of option B are noted below:

- You can have an IBGP VPNv4/v6 session between Nexus 3600 series switches within an AS and you can have an EBGP VPNv4/v6 session between data center edge routers and WAN routers.
- There is no requirement for a per VRF IBGP session between data center edge routers, like in the lite version.
- – LDP distributes IGP labels between ASBRs.
- **InterAS option B (with BGP-3107 or RFC 3107 implementation)**
- You can have an IBGP VPNv4/v6 implementation between Nexus 3600 platform switches within an AS and you can have an EBGP VPNv4/v6 session between data center edge routers and WAN routers.
- BGP-3107 enables BGP packets to carry label information without using LDP between ASBRs.
- The label mapping information for a particular route is piggybacked in the same BGP update message that is used to distribute the route itself.
- When BGP is used to distribute a particular route, it also distributes an MPLS label which is mapped to that route. Many ISPs prefer this method of configuration since it ensures complete IGP isolation between the data centers.
- **InterAS option B lite** – Support for the InterAS option B feature is restricted in the Cisco NX-OS 6.2(2) release. Details are noted in the Configuring InterAS Option B (lite version) section.

Guidelines and Limitations for Configuring InterAS Option B

The InterAS option B feature is not supported with BGP confederation AS. However, the Option B implementation is supported on Cisco Nexus 3600 platform switches.

Configuring the Switch for InterAS Option B

You enable certain features on the switch to run InterAS option B.

Before you begin

The install feature-set mpls command is available only in the default VDC, and you must enable it in default VDC.

Configure VRFs on the DC edge switches with following steps:

SUMMARY STEPS

1. **configure terminal**
2. **install feature-set mpls**

3. **feature mpls ldp**
4. **feature mpls l3vpn**
5. **feature bgp**
6. **vrf-context** *vrf-name*
7. **rd** *route-target-ext-community*
8. **address-family {ipv4 | ipv6} unicast**
9. **route-target {import | export}** *route-target-ext-community*
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls** **Example:** `switch(config)# install feature-set mpls`	Installs the MPLS feature set in the default VDC. **Note** You can only install and enable MPLS in the default VDC. Use the no form of this command to uninstall the MPLS feature set
Step 3	**feature mpls ldp** **Example:** `switch(config)# feature mpls ldp`	Enables the MPLS LDP feature on the device.. **Note** When the MPLS LDP feature is disabled on the device, no LDP commands are available.
Step 4	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.
Step 5	**feature bgp** **Example:** `switch(config)# feature bgp`	Enables the BGP feature.
Step 6	**vrf-context** *vrf-name* **Example:** `switch(config)# vrf context VPN1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 7	**rd** *route-target-ext-community* **Example:** `switch(config-vrf)# rd100:1`	Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.
Step 8	**address-family {ipv4 | ipv6} unicast** **Example:** `switch(config-vrf)# address-family ipv4 unicast`	Specifies the IPv4 or IPv6 address family type and enters address family configuration mode.

	Command or Action	Purpose
Step 9	**route-target {import | export}** *route-target-ext-community* **Example:** `switch(config-vrf-af-ip4)# route-target import 1:1`	Specifies a route-target extended community for a VRF as follows: • 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 route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended communities.
Step 10	**copy running-config startup-config** **Example:** `switch(config-vrf-af-ip4)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring BGP for InterAS Option B

Configure DC Edge switches with IBGP & EBGP VPNv4/v6 with the following steps:

Before you begin

To configure BGP for InterAS option B, you need to enable this configuration on both the IBGP and EBGP sides. Refer to Figure 1 for reference.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **address-family {vpnv4 | vpnv6} unicast**
6. **send-community {both | extended}**
7. **retain route-target all**
8. **vrf** *vrf-name*
9. **address-family {ipv4 | ipv6} unicast**
10. **exit**
11. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**router bgp** *as-number* **Example:** `switch(config)# router bgp 100`	Enters the router BGP configuration mode and assigns an autonomous system (AS) number to the local BGP speaker device.
Step 3	**neighbor** *ip-address* **Example:** `switch(config-router)# neighbor 10.0.0.2`	Adds an entry to the BGP or multiprotocol BGP neighbor table, and enters router BGP neighbor configuration mode.
Step 4	**remote-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 200`	The as-number argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family {vpnv4 | vpnv6} unicast** **Example:** `switch(config-router-neighbor)# address-family vpnv4 unicast`	Enters address family configuration mode for configuring IP VPN sessions.
Step 6	**send-community {both | extended}** **Example:** `switch(config-router-neighbor-af)# send-community both`	Specifies that a communities attribute should be sent to both BGP neighbors.
Step 7	**retain route-target all** **Example:** `switch(config-router-neighbor-af)# retain route-target all`	(Optional). Retains VPNv4/v6 address configuration on the ASBR without VRF configuration. **Note** If you have a VRF configuration on the ASBR, this command is not required.
Step 8	**vrf** *vrf-name* **Example:** `switch(config-router-neighbor-af)# vrf VPN1`	Associates the BGP process with a VRF.
Step 9	**address-family {ipv4 | ipv6} unicast** **Example:** `switch(config-router-vrf)# address-family ipv4 unicast`	Specifies the IPv4 or IPv6 address family and enters address family configuration mode.
Step 10	**exit** **Example:** `switch(config-vrf-af)# exit`	Exits IPv4 address family.

	Command or Action	Purpose
Step 11	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the Switch for InterAS Option B (with RFC 3107 implementation)

You enable certain features on the switch to run InterAS option B.

Before you begin

Configure VRFs on the DC edge switches with following steps:

SUMMARY STEPS

1. **configure terminal**
2. **install feature-set mpls**
3. **feature mpls ldp**
4. **feature mpls l3vpn**
5. **feature bgp**
6. **vrf-context** *vrf-name*
7. **rd** *route-distinguisher*
8. **address-family {ipv4 | ipv6} unicast**
9. **route-target {import | export}** *route-target-ext-community*
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls** **Example:** `switch(config)# install feature-set mpls`	Installs the MPLS feature set in the default VDC. **Note** You can only install and enable MPLS in the default VDC. Use the no form of this command to uninstall the MPLS feature set
Step 3	**feature mpls ldp** **Example:** `switch(config)# feature mpls ldp`	Enables the MPLS LDP feature on the device.. **Note** When the MPLS LDP feature is disabled on the device, no LDP commands are available.

	Command or Action	Purpose
Step 4	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.
Step 5	**feature bgp** **Example:** `switch(config)# feature bgp`	Enables the BGP feature.
Step 6	**vrf-context** *vrf-name* **Example:** `switch(config)# vrf context VPN1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 7	**rd** *route-distinguisher* **Example:** `switch(config-vrf)# rd100:1`	Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.
Step 8	**address-family {ipv4 | ipv6} unicast** **Example:** `switch(config-vrf)# address-family ipv4 unicast`	Specifies the IPv4 or IPv6 address family type and enters address family configuration mode.
Step 9	**route-target {import | export}** *route-target-ext-community* **Example:** `switch(config-vrf-af-ip4)# route-target import 1:1`	Specifies a route-target extended community for a VRF as follows: • 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 route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended communities.
Step 10	**copy running-config startup-config** **Example:** `switch(config-vrf-af-ip4)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring BGP for InterAS Option B (with RFC 3107 implementation)

Configure DC Edge switches with IBGP & EBGP VPNv4/v6 along with BGP labeled unicast family with following steps:

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family {vpnv4 | vpnv6} unicast**
4. **redistribute direct route-map** *tag*
5. **allocate-label all**
6. **exit**
7. **neighbor** *ip-address*
8. **remote-as** *as-number*
9. **address-family {ipv4|ipv6} labeled-unicast**
10. **retain route-target all**
11. **exit**
12. **neighbor** *ip-address*
13. **remote-as** *as-number*
14. **address-family {vpnv4|vpnv6} unicast**
15. **exit**
16. **address-family {vpnv4|vpnv6} unicast**
17. **Repeat the process with ASBR2**
18. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**router bgp** *as-number* **Example:** `switch(config)# router bgp 100`	Enters the router BGP configuration mode and assigns an autonomous system (AS) number to the local BGP speaker device.
Step 3	**address-family {vpnv4 | vpnv6} unicast** **Example:** `switch(config-router-neighbor)# address-family vpnv4 unicast`	Enters address family configuration mode for configuring IP VPN sessions.
Step 4	**redistribute direct route-map** *tag* **Example:** `switch(config-router-af)# redistribute direct route-map loopback`	Redistributes directly connected routes using the Border Gateway Protocol.

	Command or Action	Purpose
Step 5	**allocate-label all** **Example:** `switch(config-router-af)# allocate-label all`	Configures ASBRs with the BGP labeled unicast address family to advertise labels for the connected interface.
Step 6	**exit** **Example:** `switch(config-router-af)# exit`	Exits address family router configuration mode and enters router BGP configuration mode.
Step 7	**neighbor** *ip-address* **Example:** `switch(config-router)# neighbor 10.1.1.1`	Configures the BGP neighbour's IP address, and enters router BGP neighbour configuration mode.
Step 8	**remote-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 100`	Specifies the BGP neighbour's AS number.
Step 9	**address-family {ipv4|ipv6} labeled-unicast** **Example:** `switch(config-router-neighbor)# address-family ipv4 labeled-unicast`	Configures the ASBR with the BGP labeled unicast address family to advertise labels for the connected interface. **Note** This is the command that implements RFC 3107.
Step 10	**retain route-target all** **Example:** `switch(config-router-neighbor-af)# retain route-target all`	(Optional). Retains VPNv4/v6 address configuration on the ASBR without VRF configuration. **Note** If you have a VRF configuration on the ASBR, this command is not required.
Step 11	**exit** **Example:** `Switch(config-router-neighbor-af)# exit`	Exits router BGP neighbour address family configuration mode and returns to router BGP configuration mode.
Step 12	**neighbor** *ip-address* **Example:** `switch(config-router)# neighbor 10.1.1.1`	Configures a loopback IP address, and enters router BGP neighbor configuration mode..
Step 13	**remote-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 100`	Specifies the BGP neighbour's AS number.
Step 14	**address-family {vpnv4|vpnv6} unicast** **Example:** `switch(config-router-vrf)# address-family ipv4 unicast`	Configures the ASBR with the BGP VPNv4 unicast address family.

	Command or Action	Purpose
Step 15	**exit** **Example:** `switch(config-vrf-af)# exit`	Exits IPv4 address family.
Step 16	**address-family {vpnv4|vpnv6} unicast** **Example:** `switch(config-router-vrf)# address-family ipv4 unicast`	Configures the ASBR with the BGP VPNv4 unicast address family.
Step 17	**Repeat the process with ASBR2**	Configures ASBR2 with option B (RFC 3107) settings and implements complete IGP isolation between the two data centers DC1 and DC2.
Step 18	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Creating an ACL to filter LDP connections between the ASBRs (RFC 3107 implementation)

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list** *name*
3. **[** sequence-number **]deny tcp any any eq** *packet-length*
4. **[** sequence-number **] deny tcp any eq** *packet-length* **any**
5. **[** sequence-number **] deny udp any any eq** *packet-length*
6. **[** sequence-number **] deny udp any eq** *packet-length* **any**
7. **[** sequence-number **] permit ip any any**
8. **exit**
9. **interface** *type number*
10. **mpls ip**
11. **ip access-group** *name* **in**
12. **ip access-group** *name* **out**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	**ip access-list** *name* **Example:** `switch(config)# ip access-list LDP`	Creates an access list and enters ACL configuration mode.
Step 3	**[** sequence-number **]deny tcp any any eq** *packet-length* **Example:** `switch(config-acl)# 10 deny tcp any any eq 646`	Executes the ACL instruction as per the specified sequence.
Step 4	**[** sequence-number **] deny tcp any eq** *packet-length* **any** **Example:** `switch(config-acl)# 20 deny tcp any eq 646 any`	Executes the ACL instruction as per the specified sequence.
Step 5	**[** sequence-number **] deny udp any any eq** *packet-length* **Example:** `switch(config-acl)# 30 deny udp any any eq 646`	Executes the ACL instruction as per the specified sequence.
Step 6	**[** sequence-number **] deny udp any eq** *packet-length* **any** **Example:** `switch(config-acl)# 20 deny udp any eq 646 any`	Executes the ACL instruction as per the specified sequence.
Step 7	**[** sequence-number **] permit ip any any** **Example:** `switch(config-acl)# 50 permit ip any any`	Executes the ACL instruction as per the specified sequence.
Step 8	**exit** **Example:** `switch(config-acl)# exit`	Exits ACL configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `switch(config)# interface ethernet 2/20`	Enters interface configuration mode.
Step 10	**mpls ip** **Example:** `switch(config-if)# mpls ip`	Configures MPLS hop-by-hop forwarding on this interface.
Step 11	**ip access-group** *name* **in** **Example:** `switch(config-if)# ip access-group LDP in`	Specifies that the ACL (named LDP created in the earlier steps) be applied to inbound traffic on the interface.

	Command or Action	Purpose
Step 12	**ip access-group** *name* **out** **Example:** `switch(config-if)# ip access-group LDP out`	Specifies that the ACL (named LDP created in the earlier steps) be applied to the outbound traffic on the interface.
Step 13	**end** **Example:** `switch(config-if)# end`	Exits interface configuration mode and returns to the privileged EXEC mode

Configuring InterAS Option B (lite Version)

Guidelines and Limitations for Configuring InterAS Option B lite

- The aggregation switch supports only local VRFs, and Nexus devices within an autonomous system (AS) are connected through a VRF implementation.
- Routes learned from the IBGP peer are not sent to the EBGP peer and routes learned from an EBGP peer are not sent to IBGP VPNv4/VPNv6 peers.
- The interAS option B with MP-BGP on the EBGP side does not work with MP-BGP on the IBGP side. One interface goes to the core and one interface goes to the Layer 3 VPN.
- MP-BGP Layer 3 VPN does not work within an AS.

Configuring the Switch for InterAS Option B (lite version)

You enable certain features on the switch to run interAS option B.

Before you begin

The install feature-set mpls command is available only in the default VDC, and you must enable it in default VDC.

SUMMARY STEPS

1. **configure terminal**
2. **install feature-set mpls**
3. **feature mpls ldp**
4. **feature mpls l3vpn**
5. **feature bgp**
6. **vrf-context** *vrf-name*
7. **rd** *route-distinguisher*
8. **address-family {ipv4 | ipv6} unicast**
9. **route-target {import | export}** *route-target-ext-community*
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls** **Example:** `switch(config)# install feature-set mpls`	Installs the MPLS feature set in the default VDC. **Note** You can only install and enable MPLS in the default VDC. Use the no form of this command to uninstall the MPLS feature set.
Step 3	**feature mpls ldp** **Example:** `switch(config)# feature mpls ldp`	Enables the MPLS LDP feature on the device. When the MPLS LDP feature is disabled on the device, no LDP commands are available.
Step 4	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.
Step 5	**feature bgp** **Example:** `switch(config)# feature bgp`	Enables the BGP feature.
Step 6	**vrf-context** *vrf-name* **Example:** `switch(config)# vrf-context VPN1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 7	**rd** *route-distinguisher* **Example:** `switch(config-vrf)# rd 100:1`	Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.
Step 8	**address-family {ipv4 | ipv6} unicast** **Example:** `switch(config-vrf)# address-family ipv4 unicast`	Specifies the IPv4 or IPv6 address family type and enters address family configuration mode.
Step 9	**route-target {import | export}** *route-target-ext-community* **Example:**	Specifies a route-target extended community for a VRF as follows:

	Command or Action	Purpose
	`switch(config-vrf-af-ip4)# route-target import 1:1`	**Note** • 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 *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended communities.
Step 10	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring BGP for InterAS Option B (lite Version)

Configure EBGP VPNv4/v6 on the DC Edge switches using the following steps:

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **address-family {vpnv4 | vpnv6} unicast**
6. **send-community {both | extended}**
7. **vrf** *vrf-name*
8. **address-family {ipv4 | ipv6} unicast**
9. **exit**
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**router bgp** *as-number* **Example:** `switch(config)# router bgp 100`	Enters the router BGP configuration mode and assigns an autonomous system (AS) number to the local BGP speaker device.
Step 3	**neighbor** *ip-address* **Example:** `switch(config-router)# neighbor 10.0.0.2`	Adds an entry to the BGP or multiprotocol BGP neighbor table, and enters router BGP neighbor configuration mode.
Step 4	**remote-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 200`	The as-number argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family {vpnv4 | vpnv6} unicast** **Example:** `switch(config-router-neighbor)# address-family vpnv4 unicast`	Enters address family configuration mode for configuring IP VPN sessions.
Step 6	**send-community {both | extended}** **Example:** `switch(config-router-neighbor-af)# send-community both`	Specifies that a communities attribute should be sent to both BGP neighbors.
Step 7	**vrf** *vrf-name* **Example:** `switch(config-router-neighbor-af)# vrf VPN1`	Associates the BGP process with a VRF.
Step 8	**address-family {ipv4 | ipv6} unicast** **Example:** `switch(config-router-vrf)# address-family ipv4 unicast`	Specifies the IPv4 or IPv6 address family and enters address family configuration mode.
Step 9	**exit** **Example:** `switch(config-vrf-af)# exit`	Exits IPv4 address family.
Step 10	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying InterAS Option B Configuration

To verify InterAS option B configuration information, perform one of the following tasks:

Command	Purpose
show bgp { vpnv4 | vpnv6 } unicast [*ip-prefix/length* [**neighbors** *neighbor*]] {**vrf**{*vrf-name* | **all** } | **rd** *route-distinguisher* }	Displays VPN routes from the BGP table.
show bgp ipv6 unicast [**vrf***vrf-name*]	Displays information about BGP on a VRF for 6VPE.
show forwarding { ip | ipv6 } route vrf *vrf-name*	Displays the IP forwarding table that is associated with a VRF. Check that the loopback addresses of the local and remote CE routers are in the routing table of the PE routers.
show { ip | ipv6 } bgp[**vrf** *vrf-name*]	Displays information about BGP on a VRF..
show ip route[*ip-address* [mask]] [*protocol*] **vrf** *vrf-name*	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.
show {**ip** | **ipv6**} **routevrf** *vrf-name*	Displays the IP routing table that is associated with a VRF. Check that the loopback addresses of the local and remote CE routers are in the routing table of the PE routers.
show running-config bgp	Displays the running configuration for BGP.
show running-config vrf *vrf-name*	Displays the running configuration for VRFs.
show vrf *vrf-name* **interface** *if-type*	Verifies the route distinguisher (RD) and interface that are configured for the VRF.
trace *trace destination* **vrf** *vrf-name*	Discovers the routes that packets take when traveling to their destination. The trace command can help isolate a problem if two routers cannot communicate.

Configuration Examples for Configuring InterAS Option B

This example shows how to configure InterAS Option B

```
!--Configure VRFs on the DC edge switches --!

configure terminal
install feature-set mpls
feature mpls ldp
feature mpls l3vpn
feature bgp
vrf context VPN1
rd 100:1
address-family ipv4 unicast
route-target import 1:1
copy running-config startup-config

!--Configure DC Edge switches with IBGP & EBGP VPNv4/v6 --!
```

```
configure terminal
router bgp 100
neighbor 10.0.0.2
remote-as 200
address-family vpnv4 unicast
send-community both
retain route-target all
vrf VPN1
address-family ipv4 unicast
exit
copy running-config startup-config
```

This example shows how to configure InterAS Option B (RFC 3107)

```
!--Configure VRFs on the DC edge switches --!

configure terminal
install feature-set mpls
feature mpls ldp
feature mpls l3vpn
feature bgp
vrf context VPN1
rd 100:1
address-family ipv4 unicast
route-target import 1:1
copy running-config startup-config

!--Configure DC Edge switches with IBGP & EBGP VPNv4/v6 --!

configure terminal
router bgp 100
address-family ipv4 unicast
redistribute direct route-map loopback
allocate-label all
exit
neighbor 10.1.1.1
remote-as 100
address-family ipv4 labeled-unicast
retain route-target all
exit
neighbor 1.1.1.1
remote-as 100
address-family vpnv4 unicast
address-family vpnv6 unicast
!--Repeat the process with ASBR2. --!
copy running-config startup-config

!--Creating an ACL to filter LDP connection between the ASBRs (RFC 3107 implementation)--!

configure terminal
ip access-list LDP
10 deny tcp any any eq 646
20 deny tcp any eq 646 any
30 deny udp any any eq 646
40 deny udp any eq 646 any
50 permit ip any any
exit
interface ethernet 2/20
mpls ip
ip access-group LDP in
ip access-group LDP out
```

```
end
```

INDEX

A

address-family ipv4 labeled-unicast **13**
address-family ipv4 unicast **11, 13**
allocate-label {all | route-map} **13**

F

feature mpls static **9**

M

mpls ip forwarding **10**

N

neighbor **13**
network **11**

R

route-map **11**

S

set label-index **11**
show bgp {ip | ipv6} vrf **95**
show bgp {ipv4 | ipv6} unicast neighbors **16**
show bgp ipv4 labeled-unicast **13, 16**
show bgp link-state **16**
show bgp link-state unicast **16**
show bgp paths **16**
show ip route **95**
show ipv6 bgp **95**
show mpls label range **16**
show mpls switching **16**
show route-map **11, 16**
show running-config | inc 'feature mpls segment-routing' **9, 16**
show running-config bgp **95**
show running-config vrf **95**
show vrf vrf-name **95**