

MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

The MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs feature allows MPLS VPN interautonomous (Inter-AS) and MPLS VPN Carrier Supporting Carrier (CSC) networks to load share traffic between adjacent label switch routers (LSRs) that are connected by multiple links. The LSRs can be a pair of Autonomous System Boundary Routers (ASBRs) or a CSC-provider edge (PE) and a CSC-customer edge (CE) device. Using directly connected loopback peering allows load sharing at the Interior Gateway Protocol (IGP) level so only one Border Gateway Protocol (BGP) session is needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs except BGP.

- Prerequisites for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs, on page 1
- Restrictions for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs, on page 1
- Information About MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs, on page 4
- How to Configure MPLS VPN Load Balancing Support for Inter-AS and CSC VPN, on page 4
- Configuration Examples for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN , on page 34
- Additional References, on page 35
- Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN, on page 36

Prerequisites for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

Ensure that your Multiprotocol Label Switching (MPLS) virtual private network (VPN) network, including MPLS VPN interautonomous system (Inter-AS) or Carrier Supporting Carrier (CSC), is configured and working properly.

Restrictions for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

Load sharing using directly connected loopback peering does not apply to Carrier Supported Carrier (CSC) networks that use the Label Distribution Protocol (LDP) and an Interior Gateway Protocol (IGP) to distribute routes and Multiprotocol Label Switching (MPLS) labels.

The software does not support load balancing in interautonomous system (Inter-AS) and CSC when there are multiple links between provider edge (PE) or Autonomous System Boundary Router (ASBR) devices.

When you configure static routes in an MPLS or MPLS virtual private network (VPN) environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in Cisco software releases that support the Tag Forwarding Information Base (TFIB. The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in Cisco software releases that support the MPLS Forwarding Infrastructure (MFI). Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

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

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

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

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

Unsupported Static Routes in an MPLS Environment That Uses the TFIB

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

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

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

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

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

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

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

Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are associated with the same virtual routing and forwarding (VRF) instance:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Information About MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

Load Sharing Using Directly Connected Loopback Peering

You use the MPLS VPN Load Balancing Support for Inter-AS and CSC VPN feature to load share traffic between adjacent label switched routers (LSRs) that are connected by multiple links. The LSRs could be a pair of Autonomous System Boundary Routers (ASBRs) or a carrier supporting carrier provider edge (CSC-PE) and a CSC-customer edge (CE).

Using directly connected loopback peering allows load sharing at the Interior Gateway Protocol (IGP) level so only one Border Gateway Protocol (BGP) session is needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs except BGP.

Directly connected loopback peering enables load sharing of traffic as follows:

- A BGP session is established, using the loopback addresses of the LSRs.
- Multiprotocol Label Switching (MPLS) is enabled on the connecting links.
- Multiple static routes to the loopback address of the adjacent LSR allow IGP load sharing.
- The outgoing label to the loopback address of the adjacent LSR is an implicit null label and is inferred by the LSR.
- Because IGP load sharing is enabled on the loopback address of the adjacent LSR, any traffic destined to a prefix that is learned over the BGP session (and recurses over the loopback) is load shared.

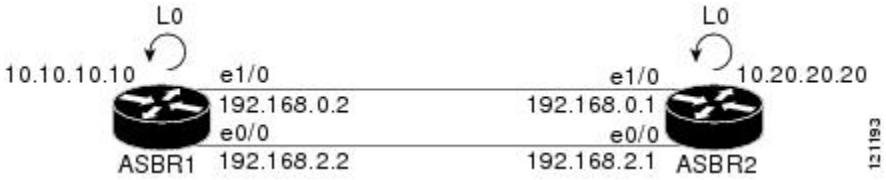
How to Configure MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Configuring Directly Connected Loopback Peering for MPLS VPN Inter-AS using ASBRs to Exchange VPN-IPv4 Addresses

This section describes the following tasks you need to do to configure peering of loopback interfaces of directly connected Autonomous System Boundary Routers (ASBRs):

The figure below shows the loopback configuration for directly connected ASBR1 and ASBR2. This configuration is used as the example in the tasks that follow.

Figure 1: Loopback Interface Configuration for Directly Connected ASBR1 and ASBR2



Configuring Loopback Interface Addresses for Directly Connected ASBRs

Perform this task to configure loopback interface addresses for directly connected Autonomous System Boundary Routers (ASBRs).

Note Loopback addresses need to be configured for each directly connected ASBR. That is, configure a loopback address for ASBR1 and for ASBR2 in the example shown in the figure above.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface- number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface- number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up and enters interface configuration mode. • The **interface-number** argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.

	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback

Perform this task to configure /32 static routes to the external Border Gateway Protocol (eBGP) neighbor loopback.

Note You need to configure /32 static routes on each of the directly connected ASBRs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	Establishes static routes. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the specified network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route. • The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword and argument name a tag value that can be used as a “match” value for controlling redistribution through the use of route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on Connecting Loopback Interfaces

Perform this task to configure forwarding on the connecting loopback interfaces.

This task is required for sessions between loopbacks. In the “Configuring /32 Static Routes to the eBGP Neighbor Loopback" section, Ethernet 1/0 and Ethernet 0/0 are the connecting interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the Loopbacks

Perform this task to configure an external Border Gateway Protocol (eBGP) session between the loopbacks.

Note You need to configure an eBGP session between loopbacks on each directly connected Autonomous System Boundary Router (ASBR).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family vpnv4** [**unicast**]
9. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
10. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard extended**]
11. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the BGP routing process. • The *as-number* indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**no bgp default route-target filter** **Example:** `Device(config)# no bgp default route-target filter`	Disables BGP route-target filtering, and enters router configuration mode. • All received BGP VPN-IPv4 routes are accepted by the device.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 8	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **unicast** keyword specifies unicast prefixes.
Step 9	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of a BGP peer group.

	Command or Action	Purpose
		• The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard extended**] **Example:** `Device(config-router-af)# neighbor 10.20.20.20 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 11	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

Perform this task to verify that load sharing occurs between loopbacks. You need to ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** {*mask* | *length*} | **labels** *label* [*network label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
3. **disable**

DETAILED STEPS

Procedure

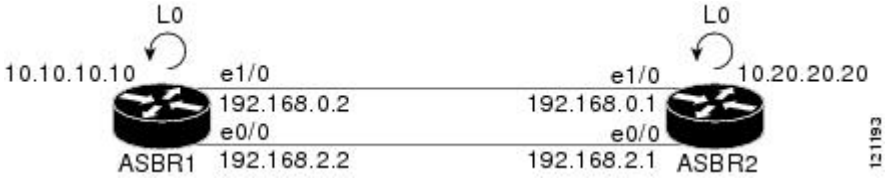
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** {*mask* | *length*} | **labels** *label* [*network label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. • Enter an optional keyword or argument if desired.
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuring Directly Connected Loopback Peering for MPLS VPN Inter-AS Using ASBRs to Exchange IPv4 Routes and Labels

The following sections describe how to configure peering of loopback interfaces of directly connected Autonomous System Boundary Routers (ASBRs) to achieve load sharing in an interautonomous system network:

The figure below shows the loopback configuration for directly connected ASBR1 and ASBR2. This configuration is used as the example in the tasks that follow.

Figure 2: Loopback Interface Configuration for Directly Connected ASBR1 and ASBR2



Configuring Loopback Interface Addresses for Directly Connected ASBRs

Note Loopback addresses need to be configured for each directly connected Autonomous System Boundary Router (ASBR). That is, configure a loopback address for ASBR1 and for ASBR2 as in the example shown in the figure above.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface number*
4. **ip address** *ip-address* [*mask* [**secondary**]]
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up and enters interface configuration mode. • The *interface-number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address* [*mask* [**secondary**]] **Example:** `Device(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback

Perform this task to configure /32 static routes to the external Border Gateway Protocol (eBGP) neighbor loopback.

Note You need to configure /32 static routes on each of the directly connected Autonomous System Boundary Routers (ASBRs).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	Establishes static routes. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the specified network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route.

	Command or Action	Purpose
		• The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword and argument name a tag value that can be used as a "match" value for controlling redistribution through the use of route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on Connecting Loopback Interfaces

This task is required for sessions between loopbacks. In the "Configuring /32 Static Routes to the eBGP Neighbor Loopback" task, Ethernet1/0 and Ethernet0/0 are the connecting interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured.

	Command or Action	Purpose
		• The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the Loopbacks

Note You need to configure an external Border Gateway Protocol (eBGP) session between loopbacks on each directly connected Autonomous System Boundary Router (ASBR).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
9. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
10. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**
12. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the BGP routing process, and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the number of the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:**	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.

	Command or Action	Purpose
	`Device(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255`	• The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *ttl* argument the time-to-live in the range from 1 to 255 hops.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 9	**address-family ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a VPN routing/forwarding instance (VRF) to associate with submode commands.
Step 10	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note**

	Command or Action	Purpose
		This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of the BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 12	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

To verify that load sharing can occur between loopbacks, ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** [*network* {*mask* |*length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
3. **disable**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

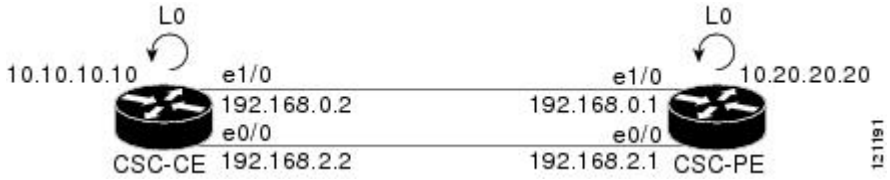
	Command or Action	Purpose
	`Device> enable`	
Step 2	**show mpls forwarding-table** [*network* {*mask* |*length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. • Enter a keyword or argument, if desired.
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuring Directly Connected Loopback Peering on MPLS VPN Carrier Supporting Carrier

The following sections explain how to load balance Carrier Supporting Carrier (CSC) traffic by peering loopback interfaces of directly connected CSC-provider edge (PE) and CSC-customer edge (CE) devices:

The figure below shows the loopback configuration for directly connected CSC-PE and CSC-CE devices. This configuration is used as the example in the tasks that follow.

Figure 3: Loopback Interface Configuration for Directly Connected CSC-PE and CSC-CE Devices



Configuring Loopback Interface Addresses on CSC-PE Devices

Note Configuration of a loopback interface address on the Carrier Supporting Carrier (CSC)-provider edge (PE) device requires the enabling of a virtual routing and forwarding (VRF) instance. The CSC-customer edge (CE) device loopback interface does not require enabling a VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface number*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]

6. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up, and enters interface configuration mode. • The *interface-number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.20.20.20 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 6	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Loopback Interface Addresses for CSC-CE Routers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface-number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up. • The *interface-number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback on the CSC-PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route vrf** *vrf-name prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**global**] [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route vrf** *vrf-name prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**global**] [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet 1/0 172.16.0.2`	Establishes static routes for a virtual routing and forwarding (VRF) instance. • The *vrf-name* argument is the name of the VRF for the static route. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the destination network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The **global** keyword specifies that the given next hop address is in the nonVRF routing table. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route.

	Command or Action	Purpose
		• The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword and argument name a tag value that can be used as a "match" value for controlling redistribution via route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback on the CSC-CE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	Establishes static routes. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the destination network.

	Command or Action	Purpose
		• The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route. • The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword and argument name a tag value that can be used as a "match" value for controlling redistribution via route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on CSC-PE Interfaces That Connect to the CSC-CE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **mpls bgp forwarding**
7. **exit**
8. Repeat Steps 3 through 6 for another connecting interface (Ethernet 0/0).
9. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vpn1`	Associates a virtual routing and forwarding (VRF) instance with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.16.0.1 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 6	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	Repeat Steps 3 through 6 for another connecting interface (Ethernet 0/0).	
Step 9	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on CSC-CE Interfaces That Connect to the CSC-PE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *typeslot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *typeslot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.
Step 5	**exit** **Example:**	Exits to global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the CSC-PE Device and the CSC-CE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
9. **ip vrf forwarding** *vrf-name*
10. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
11. **neighbor** *ip-address* **send-label**
12. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:**	Configures the Border Gateway Protocol (BGP) routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 200`	• The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.10.10.10 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(config-router)# neighbor 10.10.10.10 disable-connected-check`	Allows peering between loopbacks. • The *ip-address*argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.10.10.10 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.

	Command or Action	Purpose
Step 8	**address-family ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **ipv4** keyword configures sessions that carry standard IPv4 address prefixes. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a virtual routing and forwarding (VRF) instance to associate with submode commands.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-router-af)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 10	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 11	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 send-label`	Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device. • The *ip-address* argument is the IP address of the neighboring device.
Step 12	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the CSC-CE Device and the CSC-PE Loopback

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family ipv4** [**unicast**] [**vrf** *vrf-name*]
9. **neighbor** {*ip-address* | *peer-group-name*|*ipv6-address*] **activate**
10. **neighbor** *ip-address* **send-label**
11. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the Border Gateway Protocol (BGP) routing process. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.

	Command or Action	Purpose
		• The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 8	**address-family ipv4** [**unicast**] [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing protocols such as BGP, RIP, and static routing. • The **ipv4** keyword configures sessions that carry standard IPv4 address prefixes. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a virtual routing and forwarding (VRF) instance to associate with submode commands.
Step 9	**neighbor** {*ip-address* | *peer-group-name*|*ipv6-address*] **activate** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor.

	Command or Action	Purpose
		Note This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 10	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 send-label`	Enables a BGP device to send Multiprotocol Label Switching (MPLS) labels with BGP routes to a neighboring BGP device. • The *ip-address* argument is the IP address of the neighboring device.
Step 11	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

To verify that load sharing occurs between loopbacks, ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | [**interface**] *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
3. **disable**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask*	*length*}	**labels** *label* [*-label*]	[**interface**] *interface*	**next-hop** *address*	**lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB.

	Command or Action	Purpose
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuration Examples for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Examples: Configuring a 32 Static Route from an ASBR to the Loopback Address of Another ASBR

The following example configures a /32 static route from ASBR1 to the loopback address of ASBR2:

```
Device# configure terminal
Device(config)# ip route 10.20.20.20 255.255.255.255 e1/0 168.192.0.1
Device(config)# ip route 10.20.20.20 255.255.255.255 e0/0 168.192.2.1
```

The following example configures a /32 static route from ASBR2 to the loopback address of ASBR1:

```
Device# configure terminal
Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 e1/0 168.192.0.2
Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 e0/0 168.192.2.2
```

Example: Configuring BGP MPLS Forwarding on the Interfaces Connecting ASBRs

The following example configures the Border Gateway Protocol (BGP) and Multiprotocol Label Switching (MPLS) forwarding on the interfaces connecting ASBR2 with ASBR1:

```
Device# configure terminal
Device(config)# interface ethernet 1/0
Device(config-if)# ip vrf forwarding vpn1
Device(config-if)# ip address 168.192.0.1 255.255.255.255
Device(config-if)# mpls bgp forwarding
Device(config-if)# exit
Device(config)# interface ethernet 0/0
Device(config-if)# ip vrf forwarding vpn1
Device(config-if)# ip address 168.192.2.1 255.255.255.255
Device(config-if)# mpls bgp forwarding
Device(config-if)# exit
```

Example: Configuring VPNv4 Sessions on an ASBR

The following example configures VPNv4 sessions on ASBR2:

```
Device# configure terminal
Device(config)# router bgp 200
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# neighbor 10.10.10.10 remote-as 100
Device(config-router)# neighbor 10.10.10.10 disable-connected-check
Device(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255
Device(config-router)# neighbor 10.10.10.10 update-source Loopback0
!
Device(config-router)# address-family vpnv4
Device(config-router-af)# neighbor 10.10.10.10 activate
Device(config-router-af)# neighbor 10.10.10.10 send-community extended
Device(config-router-af)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring MPLS VPN CSC with BGP	"MPLS VPN Carrier Supporting Carrier with BGP" module in the *MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide*
Configuring BGP	"Configuring BGP" module in the *IP Routing: BGP Configuration Guide*
Configuring BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN	"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN" module in the *IP Routing: BGP Configuration Guide*

RFCs

RFC	Title
RFC 1164	Application of the Border Gateway Protocol in the Internet
RFC 1171	A Border Gateway Protocol 4
RFC 1700	Assigned Numbers
RFC 1966	BGP Route Reflection: An Alternative to Full Mesh IBGP
RFC 2283	Multiprotocol Extensions for BGP-4
RFC 2373	IP Version 6 Addressing Architecture
RFC 2547	BGP/MPLS VPNs
RFC 2842	Capabilities Advertisement with BGP-4

RFC	Title
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 3107	Carrying Label Information in BGP-4

Technical Assistance

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

Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

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

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

Table 1: Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Feature Name	Releases	Feature Information
MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs	12.0(29)S 12.4(20)T 12.2(33)SRA 12.2(33)SXH Cisco IOS XE Release 2.2	The MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs feature allows MPLS VPN Inter-AS and MPLS VPN CSC networks to load share traffic between adjacent LSRs that are connected by multiple links. The LSRs can be a pair of ASBRs or a CSC-PE and a CSC-CE. Using directly connected loopback peering allows load sharing at the IGP level, so more than one BGP session is not needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs than BGP. In Cisco IOS Release 12.0(29)S, this feature was introduced. In Cisco IOS Release 12.4(20)T, 12.2(33)SRA, and 12.2(33)SXH, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. No commands were introduced or modified.