Configuring MPLS VPN InterAS Options

- Information About MPLS VPN InterAS Options, on page 1
- How to Configure MPLS VPN InterAS Options, on page 9
- Verifying MPLS VPN InterAS Options Configuration, on page 56
- Configuration Examples for MPLS VPN InterAS Options, on page 57
- Additional References for MPLS VPN InterAS Options, on page 73
- Feature History for MPLS VPN InterAS Options, on page 73

Information About MPLS VPN InterAS Options

The MPLS VPN InterAS Options feature provides various ways of interconnecting VPNs between different MPLS VPN service providers. This allows sites of a customer to exist on several carrier networks (autonomous systems) and have seamless VPN connectivity between these sites.

Autonomous Systems and ASBRs

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, 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.

An autonomous system boundary router (ASBR) is a device in an AS that is configured by using more than one routing protocol, and exchanges routing information with other ASBRs by using an exterior routing protocol (for example, eBGP), or use static routes, or both.

Separate autonomous systems from different service providers communicate by exchanging information in the form of VPN IP addresses and they use the following protocols to share routing information:

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

  iBGP distributes network layer information for IP prefixes within each VPN and each AS.

- Between autonomous systems, 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 autonomous systems. The primary function of eBGP is to exchange network reachability information between autonomous systems, including information about the list of AS routes. The autonomous systems 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.

MPLS VPN InterAS Options configuration is supported and can include an inter provider VPN, which is MPLS VPNs that include two or more autonomous systems, connected by separate border edge routers. The autonomous systems exchange routes using eBGP, and no iBGP or routing information is exchanged between the autonomous systems.

MPLS VPN InterAS Options

The following options defined in RFC4364 provide MPLS VPN connectivity between different autonomous systems:

- InterAS Option A: This option provides back-to-back virtual routing and forwarding (VRF) connectivity. Here, MPLS VPN providers exchange routes across VRF interfaces.
- InterAS Option B: This option provides VPNv4 route distribution between ASBRs.
- InterAS Option AB: This option combines the best functionality of an interAS option A and interAS option B network to allow an MPLS VPN service provider to interconnect different autonomous systems to provide VPN services.

InterAS Option A

In terms of configuration, interAS Option A is the simplest of all available options.

A typical AS consists of these devices – Provider Edge(PE), Customer Edge(CE) and an Autonomous System Boundary Router(ASBR). The target is to enable VRF connectivity between CE devices (also referred to as VPN sites) in a network. In order to facilitate interAS option A, you have to perform the following for each VPN site:

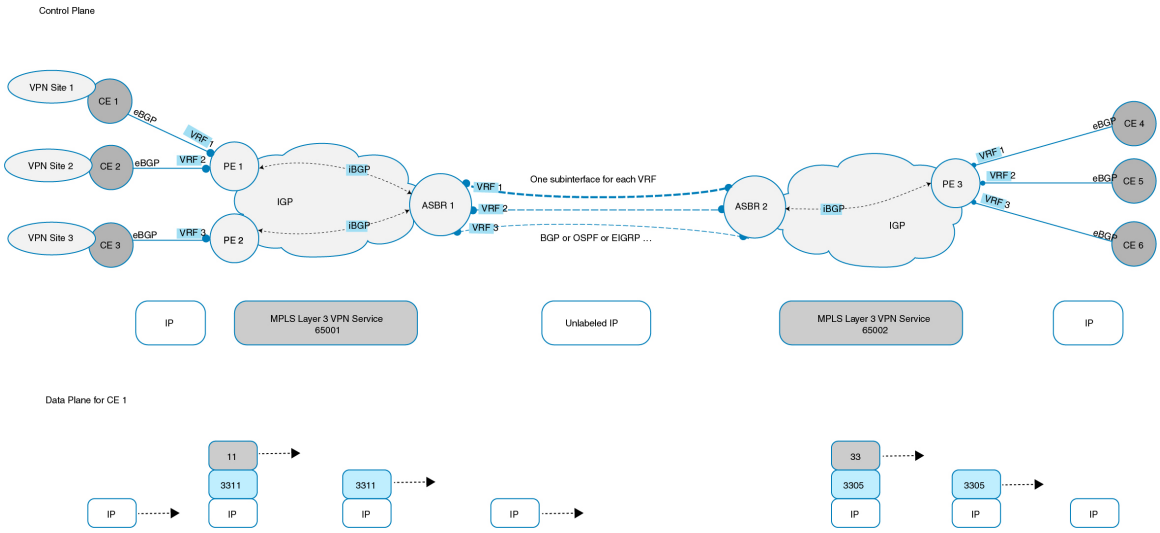
- Assign a VRF interface to each VPN site
- Define an interface or sub-interface for each VRF interface. (If multiple VPN sites are involved, they cannot all be associated with a single interface, and therefore, a sub-interface must be configured for each VRF). Optionally, a dedicated QoS policy may be applied to each subinterface.
- Create a BGP (or other routing protocol) session for each VRF.

With the above configuration in place, traffic flow with option A is as follows: Within the AS, data packets travel like regular Layer 3 VPN traffic. Traffic flow between ASBRs when traversing autonomous systems is in the form of unlabeled IP packets on a VRF interface. Any routing protocol may be used to exchange routing information between the ASBRs in the different autonomous systems.

While this option provides certain advantages (flexibility in terms of the routing protocol that can be used within an AS and between ASBRs, and security by means of a QoS policy on a subinterface), the scale for interAS option A is limited by the scale numbers for subinterfaces and VRFs. This option is therefore suited only to scenarios where the number of VPNs and the number of routes to transfer, is limited (and not likely to increase).

The figure below shows the data packet flow from CE 1, CE 2, CE 3 to CE 4, CE 5, CE 6 respectively. The explanation below takes the instance of the route advertisement and data packet flow from CE1 in AS-65001 to CE 4 in AS-65002.

Figure 1: MPLS VPN InterAS Option A Topology

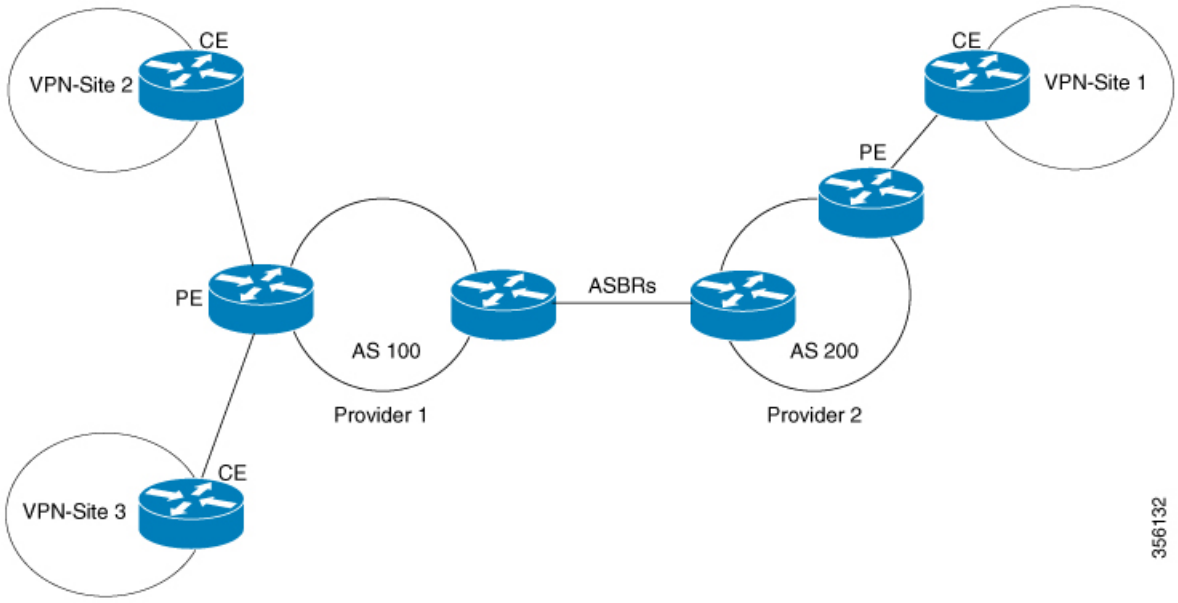


The IP traffic between CE 1 and PE 1 is sent over a VRF sub-interface by using eBGP. Once the packet reaches PE 1 it is sent to ASBR 1 as a two-label MPLS stack. The outermost label is the Interior Gateway protocol (IGP) label and the inner label is the VPN label. Layer 3 VPN traffic is sent from PE 1 to ASBR 1 in AS-65001 and from ASBR 2 to PE 3 in AS-65002 over a MPLS cloud. At ASBR 1, both the labels (IGP and VPN) are popped (removed). From ASBR 1 to ASBR 2 traffic flows as an unlabelled IP packet on a VRF interface. In this example, the routing protocol used between the two ASBRs is eBGP. The two label MPLS stack is pushed once the IP packet reaches ASBR 2. After the packet reaches PE 3, the VPN label is removed. The IGP label is also popped in case of explicit NULL IGP. The VPN packet is sent to CE4 through a VRF interface.

InterAS Option B

In an interAS option B network, ASBR ports are connected by one or more interfaces that are enabled to receive MPLS traffic. With this option, the ASBRs peer with each other using eBGP session. The ASBR also functions as a PE router and peers with every PE router in their AS. The ASBR does not hold any VRFs but holds all or a subset of VPNv4 routes from PE router that need to be passed to the other AS. VPNv4 routes are kept unique in ASBR using route-distinguisher and are filtered using route targets. The ASBRs exchange VPNv4 routes and VPN labels using eBGP.

Figure 2: Topology for InterAS Option B



Two methods are supported to distribute the next hop for VPNv4 routes between ASBRs. There is no requirement for LDP or any IGP to be enabled on the link connecting the two ASBRs. The MP-eBGP session between directly connected interfaces on the ASBRs enables the interfaces to forward labeled packets. To ensure this MPLS forwarding for directly connected BGP peers, you must configure mpls bgp forwarding command on the interface connecting to ASBR. This command is implemented in the IOS for directly connected interfaces. Upto 200 BGP neighbors can be configured.

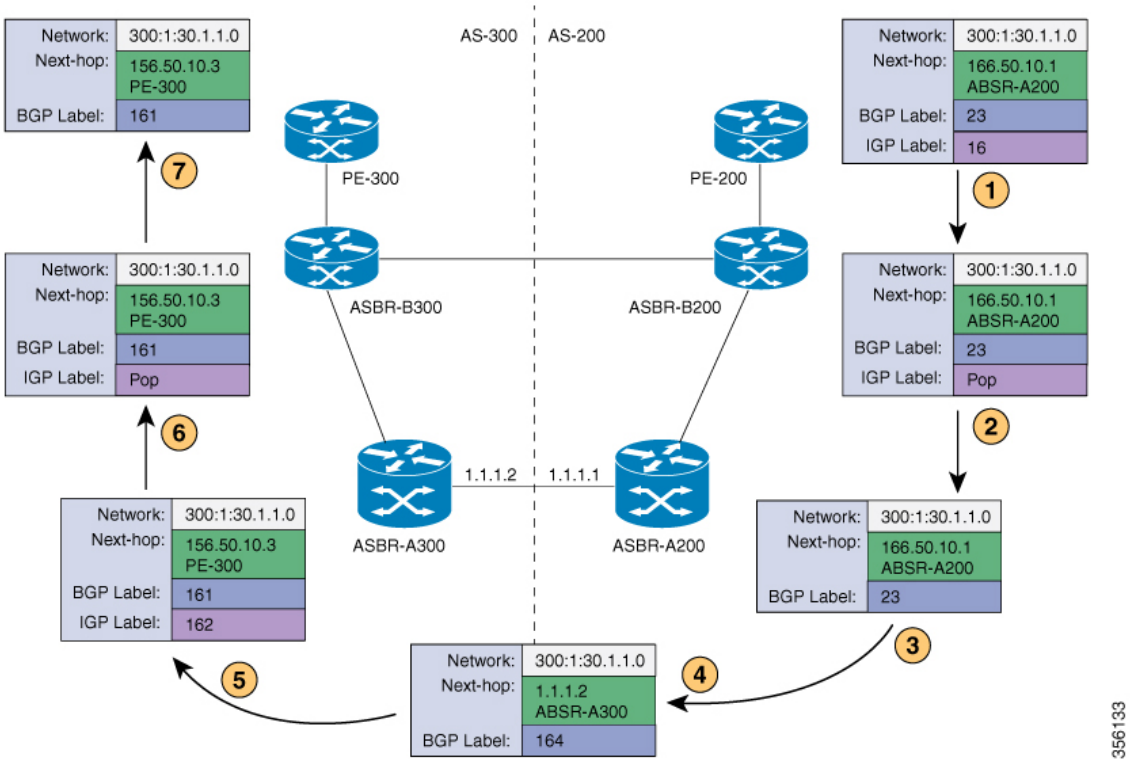
- **Next-hop-self Method:** Changing next-hop to that of the local ASBR for all VPNv4 routes learnt from the other ASBR.
- **Redistribute Connected Subnets Method:** Redistributing the next hop address of the remote ASBR into the local IGP using redistribute connected subnets command , i.e., the next hop is not changed when the VPNv4 routes are redistributed into the local AS.

Note In case of multiple equal paths - ECMP towards remote AS, you have to configure MPLS static label bindings towards remote Loopback on ASBR. Otherwise, you may experience packet loss.

The label switch path forwarding sections described below has AS200 configured with the Next-hop-self method and the AS300 is configured with Redistribute-subnet method.

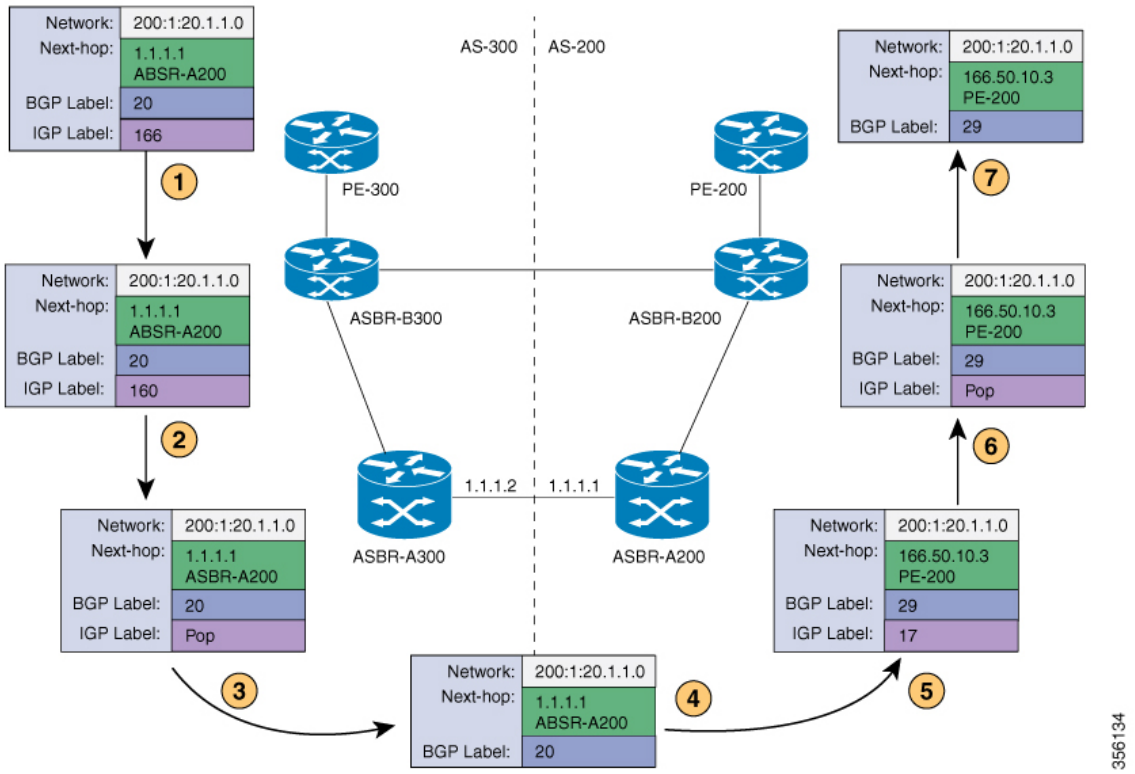
Next-Hop Self Method

The following figure shows the label forwarding path for next-hop-self method. The labels get pushed, swapped and popped on the stack as packet makes its way from PE-200 in AS 200 to PE-300 in AS 300. In step 5, ASBR-A300 receives labeled frame, replaces label 164 with label 161 pushes IGP label 162 onto the label stack.



Redistribute Connected Subnet Method

The following figure shows the label forwarding path for Redistribute connected subnets method. The labels get pushed, swapped and popped on the stack as packet travels from PE- 300 in AS 300 to PE-200 in AS 200. In step 5, ASBR-A200 receives frame with BGP label 20, swaps it with label 29 and pushes label 17.



InterAS Option AB

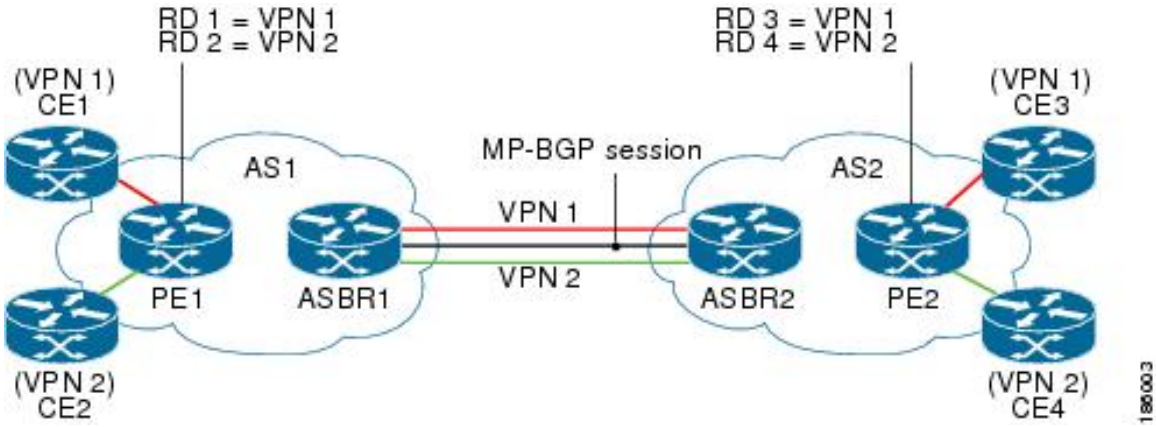
MPLS VPN service providers need to interconnect different autonomous systems to provide service for multiple VPN customers. The MPLS VPN InterAS Option AB feature allows the different autonomous systems to interconnect by using a single MP-BGP session in the global routing table to carry control plane traffic. This MP-BGP session signals VPN prefixes between two ASBRs for each VRF instance. This traffic can either be IP or MPLS.

MPLS BGP forwarding or LDP does not have to be configured between the two ASBRs because the VPN traffic that is IP traffic over a VRF-specific interface.

The interAS option AB feature provides the following benefits for service providers:

- IP QoS functions between ASBR peers are maintained for customer SLAs.
- Dataplane traffic is isolated on a per-VRF basis for security purposes.
- A dedicated QoS policy can be applied on each VRF by attaching the policy on an SVI.

Route Distribution and Packet Forwarding



The following attributes describe the topology of the sample interAS Option AB network shown in the figure above:

- CE1 and CE3 belong to VPN 1.
- CE2 and CE 4 belong to VPN 2.
- PE1 uses route distinguisher 1 (RD 1) for VPN 1 (VRF 1) and RD 2 for VPN 2 (VRF 2).
- PE2 uses RD 3 for VPN 1 (VRF 1) and RD 4 for VPN 2 (VRF 2).
- ASBR1 has VRF 1 provisioned with RD 5 and VRF 2 provisioned with RD 6.
- ASBR2 has VRF 1 provisioned with RD 7 and VRF 2 provisioned and RD 8.
- ASBR1 and ASBR2 have three links between them:
  - VRF 1
  - VRF 2
  - MP-BGP session

Route Distribution for VPN 1

A route distinguisher (RD) is an identifier attached to a route that identifies which VPN belongs to each route. Each routing instance must have a unique RD autonomous system associated with it. The RD is used to place a boundary around a VPN so that the same IP address prefixes can be used in different VPNs without having these IP address prefixes overlap. An RD statement is required if the instance type is a VRF.

The following process describes the route distribution process for VPN 1 in the figure above. Prefix “N” is used in this process to indicate the IP address of a VPN.

ASBR 1

- CE1 advertises the prefix N to PE1.
- PE1 advertises a VPN prefix RD 1:N to ASBR1 through MP-iBGP.
- ASBR1 imports the prefix into VPN 1 and creates a prefix RD 5:N.

- ASBR1 advertises the imported prefix RD 5:N to ASBR2. ASBR1 sets itself as the next hop for prefix RD 5:N and allocates a local label that is signaled with this prefix.
- ASBR1 advertises the route with the export RT configured on the VRF rather than the originally received RTs. By default, ASBR1 does not advertise the source prefix RD 1:N to ASBR2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.

ASBR 2

- ASBR2 receives the prefix RD 5:N and imports it into VPN 1 as RD 7:N.
- ASBR2 advertises the route with the export RT configured on the VRF rather than the originally received RTs.
- While importing the prefix, ASBR2 sets the next hop of RD 7:N to the ASBR1 interface IP address in VRF 1. The next hop table ID is also set to VRF 1. When installing the MPLS forwarding entry for RD 7:N, by default ASBR2 does not install the outgoing label in the forwarding process. This enables the traffic between the ASBRs to be IP.
- ASBR2 advertises the imported prefix RD 7:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signalled with the prefix. By default, ASBR2 does not advertise the source prefix RD 5:N to PE2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.
- PE2 imports the RD 7:N into VRF 1 as RD 3:N.

Packet Forwarding for VPN 1

The following packet forwarding process works the same as it does in an Option A scenario. The ASBR acts like the PE by terminating the VPN and then forwards its traffic as standard IP packets with no VPN label to the next PE, which in turn repeats the VPN process. Each PE device, therefore, treats the adjacent PE device as a CE device, and the standard Layer 3 MPLS VPN mechanisms are used for route redistribution with each autonomous system; that is, the PEs use external BGP (eBGP) to distribute unlabeled IPv4 addresses to each other.

- CE3 sends a packet destined for N to PE2.
- PE2 encapsulates the packet with the VPN label allocated by ASBR2 and the Interior Gateway Protocol (IGP) label needed to tunnel the packet to ASBR2.
- The packet arrives on ASBR2 with the VPN label. ASBR2 removes the VPN label and sends the packet as IP to ASBR1 on the VRF 1 interface.
- The IP packet arrives at ASBR1 on the VRF 1 interface. ASBR1 then encapsulates the packet with the VPN label allocated by PE1 and the IGP label needed to tunnel the packet to PE1.
- The packet arrives on PE1 with the VPN label. PE1 disposes the VPN label and forwards the IP packet to CE1.

Route Distribution for VPN 2

The following information describes the route distribution process for VPN 2 in the figure above:

ASBR 1

- CE2 advertises prefix N to PE1, where N is the VPN IP address.

- PE1 advertises a VPN prefix RD 2:N to ASBR1 through MP-iBGP.
- ASBR1 imports the prefix into VPN 2 and creates a prefix RD 6:N.
- ASBR1 advertises the imported prefix RD 6:N to ASBR2. It sets itself as the next hop for this prefix and also allocates a local label that is signalled with the prefix. By default, ASBR1 does not advertise the source prefix RD 2:N to ASBR2. This advertisement is suppressed as the prefix is being imported into an Option AB VRF.

ASBR 2

- ASBR2 receives the prefix RD 6:N and imports it into VPN 2 as RD 8:N.
- While importing the prefix, ASBR2 sets the next hop of RD 8:N to ASBR1s interface address in VRF 2. The next hop table ID is also set to that of VRF 2. While installing the MPLS forwarding entry for RD 8:N, by default ASBR2 does not install the outgoing label in the forwarding process. This enables traffic between the ASBRs to be IP.
- ASBR2 advertises the imported prefix RD 8:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signalled with the prefix. By default, ASBR2 does not advertise the source prefix RD 6:N to PE2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.
- PE2 imports the RD 8:N into VRF 2 as RD 4:N.

How to Configure MPLS VPN InterAS Options

The following section provides information about how to configure MPLS VPN InterAS Options.

Configuring MPLS VPN InterAS Option A

Sending AS: Configuring PE

Complete the following tasks to configure the PE which is in the AS sending data to another AS.

Sending AS: Configuring a VRF for a PE

Beginning in user EXEC mode complete the following steps to configure a VRF for a PE which is in the sending AS:

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**vrf definition** *vrf-name*
	Example:
	`Device(config)# vrf definition cu1`
	`Device(config-vrf)#`
		Configures a VRF table and enters VRF configuration mode.

Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF instance.

Step 5	**address-family ipv4**
	Example:
	`Device(config-vrf)# address-family ipv4`
	`Device(config-vrf-af)#`
		Places the device in address family configuration mode, from which you can configure routing sessions that use standard IPv6 address prefixes.

Step 6	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 100:1`	Creates a list of export route target communities for the specified VRF.
Step 7	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 100:2`	Creates a list of import route target communities for the specified VRF.

Step 8	**exit-address-family**
	Example:
	`Device(config-vrf-af)#`
	`exit-address-family`
	`Device(config-vrf)#`
		Exits the address family configuration mode and returns to VRF configuration mode.

Step 9	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Places the device in address family configuration mode, from which you can configure routing sessions that use standard IPv6 address prefixes.

	Command or Action	Purpose
Step 10	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 100:101`	Creates a list of export route target communities for the specified VRF.
Step 11	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 100:102`	Creates a list of import route target communities for the specified VRF.

Step 12	**exit-address-family**
	Example:
	`Device(config-vrf-af)# exit-address-family`
	`Device(config-vrf)#`
		Exits the address family configuration mode and returns to VRF configuration mode.

Sending AS: Configuring a PE-CE Interface

Beginning in privileged EXEC mode complete the following steps to configure a PE-CE interface which is in the sending AS:

Procedure

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

Step 2	**interface** {*interface-id* | *subinterface-id* | *vlan-id*}
	Example:
	`Device(config)# interface Gi1/1/0/13.1`
	`Device(config-if)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

Step 3	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if)# encapsulation dot1q 900`	Enables IEEE 802.1Q encapsulation of traffic on a specified interface.

	Command or Action	Purpose
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding cu1`	Associates the VRF with the Layer 3 interface.
Step 5	**ip address** *ip address mask* [**secondary**] **Example:** `Device(config-if)# ip address 140.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

Step 6	**exit**
	Example:
	`Device(config-if)# exit`
	`Device(config)#`
		Exits interface configuration mode and returns to global configuration mode.

Sending AS: Configuring BGP

Beginning in user EXEC mode complete the following steps to configure a BGP session for a PE which is in the sending AS:

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** *autonomous-system-number*
	Example:
	`Device(config)# router bgp 65001`
	`Device(config-router)#`
		Configures a BGP routing process.

Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 2.2.2.2 remote-as 65001`	Configures an entry to the BGP neighbor table.

	Command or Action	Purpose

Step 5	**address-family** *ipv4* [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | [**vrf** *vrf-name*]
	Example:
	`Device(config-router)# address-family ipv4`
	`Device(config-router-af)#`
		Enters address family configuration mode for configuring BGP routing sessions that use standard IPv4 address prefixes.

Step 6	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 2.2.2.2 activate`	Enables the exchange of information with a BGP neighbor.

Step 7	**exit address-family**
	Example:
	`Device(config-router-af)# exit address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 8	**address-family** *vpnv4*
	Example:
	`Device(config-router)# address-family vpnv4`
	`Device(config-router-af)#`
		Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.

Step 9	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 2.2.2.2 activate`	Enables the exchange of information with a BGP neighbor.
Step 10	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 2.2.2.2 send-community both`	Enables the exchange of information with a BGP neighbor.
Step 11	**exit address-family** **Example:** `Device(config-router-af)# exit`	Exits BGP address-family submode.

	Command or Action	Purpose
	`address-family` `Device(config-router)#`	

Step 12	**address-family** *vpnv6*
	Example:
	`Device(config-router)# address-family vpnv6`
	`Device(config-router-af)#`
		Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv6 address prefixes.

Step 13	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 2.2.2.2 activate`	Enables the exchange of information with a BGP neighbor.
Step 14	**neighbor** *ip-address* **send-community extended** **Example:** `Device(config-router-af)# neighbor 2.2.2.2 send-community extended`	Specifies that a community attribute should be sent to a BGP neighbor.

Step 15	**exit address-family**
	Example:
	`Device(config-router-af)# exit address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 16	**address-family ipv4 vrf** *vrf-name*
	Example:
	`Device(config-router)# address-family ipv4 vrf cu1`
	`Device(config-router-af)#`
		Enters address family configuration mode for configuring BGP routing sessions that use standard IPv4 address prefixes.

Step 17	**redistribute** *protocol* **Example:** `Device(config-router-af)# redistribute connected`	Redistributes routes from one routing domain into another routing domain.
Step 18	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 140.1.1.2 remote-as 65002`	Configures an entry to the BGP neighbor table.

	Command or Action	Purpose
Step 19	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 140.1.1.2 activate`	Enables the exchange of information with a BGP neighbor.

Step 20	**exit address-family**
	Example:
	`Device(config-router-af)# exit address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 21	**exit** **Example:** `Device(config-router)# exit`	Exits router BGP mode.

Sending AS: Configuring a PE-P Interface and IGP

Beginning in user EXEC mode complete the following steps to configure a PE-P interface and IGP which is in the sending AS:

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**
	{*interface-id* | *subinterface-id* | *vlan-id*}
	Example:
	`Device(config)# interface po91`
	`Device(config-if)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Sets the interface to the routed-interface status and erases all Layer 2 configurations.

	Command or Action	Purpose
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 91.1.1.1 255.255.255.248`	Sets a primary or secondary IP address for an interface.
Step 6	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 2 area 0`	Enables OSPF on an interface.
Step 7	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 and IPv6 packets along normally routed paths for a particular interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 9	**router ospf** *process-id* **Example:** `Device(config)# router ospf 2`	Configures an OSPF routing process and assigns a process number.
Step 10	**router-id** *ip-address* **Example:** `Device(config-router)# router-id 1.1.1.1`	Specifies a fixed router ID.
Step 11	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Sending AS: Configuring P

Complete the following tasks to configure the P which is in the AS sending data to another AS.

Sending AS: Configuring P-PE Interface and IGP

Beginning in user EXEC mode complete the following steps to configure a P-PE interface and IGP which is in the sending AS:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

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

Step 3	**interface** {*interface-id* | *subinterface-id* | *vlan-id*}
	Example:
	`Device(config)# interface Port-channel91`
	`Device(config-if)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Sets the interface to the routed-interface status and erases all Layer 2 configuration.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 91.1.1.2 255.255.255.248`	Sets a primary or secondary IP address for an interface.
Step 6	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 2 area 0`	Enables OSPF on an interface.
Step 7	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 and IPv6 packets along normally routed paths for a particular interface.

Step 8	**exit**
	Example:
	`Device(config-if)# exit`
	`Device(config)#`
		Exits interface configuration mode.

Step 9	**interface** {*interface-id* | *subinterface-id* | *vlan-id*} **Example:** `Device(config)# interface Port-channel92`	Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

	Command or Action	Purpose
Step 10	**no switchport** **Example:** `Device(config-if)# no switchport`	Set the interface to the routed-interface status erases all Layer 2 configurations.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 92.1.1.2 255.255.255.248`	Sets a primary or secondary IP address for an interface.
Step 12	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 2 area 0`	Enables OSPF on an interface.
Step 13	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 and IPv6 packets along normally routed paths for a particular interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Step 15	**router ospf** *process-id*
	Example:
	`Device(config)# router ospf 2`
	`Device(config-router)#`
		Configures an OSPF routing process and assign a process number.

Step 16	**router-id** *ip-address* **Example:** `Device(config-router)# router-id 5.5.5.5`	Specifies a fixed router ID.
Step 17	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode, and returns to privileged EXEC mode.

Sending AS: Configuring ASBR

Complete the following tasks to configure the ASBR which is in the AS sending data to another AS.

Sending AS: Configuring VRF for ASBR

Beginning in user EXEC mode complete the following steps to configure a VRF for a ASBR which is in the sending AS:

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	**vrf definition** *vrf-name*
	Example:
	`Device(config)# vrf definition cu1`
	`Device(config-vrf)#`
		Configures a VRF table and enters VRF configuration mode.

Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:2`	Creates routing and forwarding tables for a VRF instance.

Step 5	**address-family ipv4**
	Example:
	`Device(config-vrf)# address-family ipv4`
	`Device(config-vrf-af)#`
		The address-family ipv4 command places the device in address family configuration mode, from which you can configure routing sessions that use standard IPv4 address prefixes.

Step 6	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 100:2`	Creates a list of export route target communities for the specified VRF.
Step 7	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 100:1`	Creates a list of import route target communities for the specified VRF.
Step 8	**exit-address-family** **Example:**	Leaves the address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
	`Device(config-vrf-af)# exit-address-family` `Device(config-vrf)#`	
Step 9	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Places the device in address family configuration mode, from which you can configure routing sessions that use standard IPv6 address prefixes.
Step 10	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 100:102`	Creates a list of export route target communities for the specified VRF.
Step 11	**route-target import***route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 100:101`	Creates a list of import route target communities for the specified VRF.

Step 12	**exit-address-family**
	Example:
	`Device(config-vrf-af)# exit-address-family`
	`Device(config-vrf)#`
		Exits the address family configuration mode and returns to router configuration mode.

Step 13	**exit** **Example:** `Device(config-vrf)# exit`	Exits the router configuration mode and returns to global configuration mode.

Sending AS: Configuring Interface Towards the Receiving ASBR

Beginning in privileged EXEC mode complete the following steps to configure an interface towards the receiving ASBR:

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	

Step 2	**interface** { *interface-id* | *subinterface-id* | *vlan-id* }
	Example:
	`Device(config)# interface fo1/0/10.1`
	`Device(config-subif)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

Step 3	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 900`	Enables IEEE 802.1Q encapsulation of traffic on a specified interface.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-subif)# vrf forwarding cu1`	Associates the VRF with the Layer 3 interface.
Step 5	**ip address** *ip address mask* [**secondary**] **Example:** `Device(config-subif)# ip address 141.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

Sending AS: Configuring BGP

Beginning in privileged EXEC mode complete the following steps to configure a BGP session on the ASBR which is in the sending AS:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number* **Example:** `Device(config-if)# router bgp 65001`	Configures a BGP routing process.

	Command or Action	Purpose
Step 3	**bgp log-neighbor changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 1.1.1.1 remote-as 65001`	Configures an entry to the BGP neighbor table.
Step 5	**neighbor** *ip-address* **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 1.1.1.1 update-source Loopback0`	Allows Cisco IOS software to use a specific operational interface for TCP connections by the BGP sessions.

Step 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | [**vrf** *vrf-name*]
	Example:
	`Device(config-router)# address-family ipv4`
	`Device(config-router-af)#`
		Enters address family configuration mode for configuring BGP routing sessions that use standard IP Version 4 address prefixes.

Step 7	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 1.1.1.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits BGP address-family submode.
Step 9	**address-family** *vpnv4* **Example:** `Device(config-router)# address-family vpnv4`	Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.
Step 10	**neighbor** *ip-address* **activate** **Example:**	Enables the exchange of information with a BGP neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 1.1.1.1 activate`	
Step 11	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 1.1.1.1 send-community both`	Enables the exchange of information with a BGP neighbor.

Step 12	**exit-address-family**
	Example:
	`Device(config-router-af)# exit-address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 13	**address-family** *vpnv6*
	Example:
	`Device(config-router)# address-family vpnv6`
	`Device(config-router-af)#`
		Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv6 address prefixes.

Step 14	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 1.1.1.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 15	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 1.1.1.1 send-community both`	Enables the exchange of information with a BGP neighbor.

Step 16	**exit-address-family**
	Example:
	`Device(config-router-af)# exit-address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

	Command or Action	Purpose
Step 17	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf cu1`	Enters address family configuration mode for configuring BGP routing sessions that use standard IP Version 4 address prefixes.
Step 18	**redistribute** *protocol* **Example:** `Device(config-router-af)# redistribute connected`	Redistributes routes from one routing domain into another routing domain.
Step 19	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 141.1.1.2 remote-as 65002`	Configures an entry to the BGP neighbor table.
Step 20	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 141.1.1.2 activate`	Enables the exchange of information with a BGP neighbor.
Step 21	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits BGP address-family submode.

Sending AS: Configuring a ASBR-P Interface and a IGP

Beginning in privileged EXEC mode complete the following steps to configure a ASBR-P interface and a IGP in the sending AS:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** {*interface-id* | *subinterface-id* | *vlan-id*} **Example:**	Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

	Command or Action	Purpose
	`Device(config)# interface Port-channel92`	
Step 3	**no switchport** **Example:** `Device(config-if)# no switchport`	Set the interface to the routed-interface status erases all Layer 2 configurations.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 92.1.1.1 255.255.255.248`	Sets a primary or secondary IP address for an interface.
Step 5	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 2 area 0`	Enables OSPF on an interface.
Step 6	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables Multiprotocol Label Switching (MPLS) forwarding of IPv4 and IPv6 packets along normally routed paths for a particular interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Receiving AS: Configuring ASBR

Complete the following tasks to configure the ASBR which is in the AS receiving data from another AS.

Receiving AS: Configuring VRF for ASBR

Beginning in user EXEC mode complete the following steps to configure a VRF for a ASBR which is in the receiving AS:

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.

	Command or Action	Purpose

Step 3	**vrf definition** *vrf-name*
	Example:
	`Device(config)# vrf definition cu1`
	`Device(config-vrf)#`
		Configures a VRF table and enters VRF configuration mode.

Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:3`	Creates routing and forwarding tables for a VRF instance.

Step 5	**address-family ipv4**
	Example:
	`Device(config-vrf)# address-family ipv4`
	`Device(config-vrf-af)#`
		The address-family ipv4 command places the device in address family configuration mode, from which you can configure routing sessions that use standard IPv4 address prefixes.

Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 200:2`	Creates a list of export route target communities for the specified VRF.
Step 7	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 200:1`	Creates a list of import route target communities for the specified VRF.
Step 8	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Leaves the address family configuration mode and returns to router configuration mode.

Step 9	**address-family ipv6**
	Example:
	`Device(config-vrf)# address-family ipv6`
	`Device(config-vrf-af)#`
		Places the device in address family configuration mode, from which you can configure routing sessions that use standard IPv6 address prefixes.

Step 10	**route-target export** *route-target-ext-community* **Example:**	Creates a list of export route target communities for the specified VRF.

	Command or Action	Purpose
	`Device(config-vrf-af)# route-target export 200:101`	
Step 11	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 200:102`	Creates a list of import route target communities for the specified VRF.

Step 12	**exit-address-family**
	Example:
	`Device(config-vrf-af)# exit-address-family`
	`Device(config-vrf)#`
		Exits the address family configuration mode and returns to router configuration mode.

Step 13	**exit** **Example:** `Device(config-vrf)# exit`	Exits the router configuration mode and returns to global configuration mode.

Receiving AS: Configuring Interface Towards the Sending ASBR

Beginning in privileged EXEC mode complete the following steps to configure an interface towards the sending ASBR:

Procedure

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

Step 2	**interface** {*interface-id* | *subinterface-id* | *vlan-id*}
	Example:
	`Device(config)# interface fo1/0/10.1`
	`Device(config-subif)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

Step 3	**encapsulation dot1q** *vlan-id* **Example:**	Enables IEEE 802.1Q encapsulation of traffic on a specified interface.

	Command or Action	Purpose
	`Device(config-subif)# encapsulation dot1q 900`	
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-subif)# vrf forwarding cu1`	Associates the VRF with the Layer 3 interface.
Step 5	**ip address** *ip address mask* [**secondary**] **Example:** `Device(config-subif)# ip address 141.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

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

Receiving AS: Configuring BGP

Beginning in privileged EXEC mode complete the following steps to configure a BGP session on the ASBR which is in the receiving AS:

Procedure

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

Step 2	**router bgp** *autonomous-system-number*
	Example:
	`Device(config)# router bgp 65002`
	`Device(config-router)#`
		Configures a BGP routing process.

Step 3	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 30.30.30.30 remote-as 65002`	Configures an entry to the BGP neighbor table.

	Command or Action	Purpose

Step 4	**address-family** *ipv4* [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | [**vrf** *vrf-name*]
	Example:
	`Device(config-router)# address-family ipv4`
	`Device(config-router-af)#`
		Enters address family configuration mode for configuring BGP routing sessions that use standard IP Version 4 address prefixes.

Step 5	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 30.30.30.30 activate`	Enables the exchange of information with a BGP neighbor.

Step 6	**exit**
	Example:
	`Device(config-router-af)# exit`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 7	**address-family** *ipv6*
	Example:
	`Device(config-router)# address-family ipv6`
	`Device(config-router-af)#`
		Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.

Step 8	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 30.30.30.30 activate`	Enables the exchange of information with a BGP neighbor.

Step 9	**exit address-family**
	Example:
	`Device(config-router-af)# exit address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 10	**address-family** *vpnv4*
	Example:
	`Device(config-router)# address-family vpnv4`
	`Device(config-router-af)#`
		Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv6 address prefixes.

	Command or Action	Purpose
Step 11	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 30.30.30.30 activate`	Enables the exchange of information with a BGP neighbor.
Step 12	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send−community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 30.30.30.30 send-community both`	Enables the exchange of information with a BGP neighbor.

Step 13	**exit**
	Example:
	`Device(config-router-af)# exit`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 14	**address-family** *vpnv6*
	Example:
	`Device(config-router)# address-family vpnv6`
	`Device(config-router-af)#`
		Enters address family configuration mode for configuring BGP routing sessions that use standard IP Version 4 address prefixes.

Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 30.30.30.30 activate`	Enables the exchange of information with a BGP neighbor.
Step 16	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send−community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 30.30.30.30 send-community both`	Enables the exchange of information with a BGP neighbor.

Step 17	**exit**
	Example:
	`Device(config-router-af)# exit`
	`Device(config-router)#`
		Exits BGP address-family submode.

	Command or Action	Purpose

Step 18	**address-family** *ipv4*
	Example:
	`Device(config-router)# address-family ipv4 vrf cu1`
	`Device(config-router-af)#`
		Enters address family configuration mode for configuring BGP routing sessions that use standard IP Version 4 address prefixes.

Step 19	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 141.1.1.1 remote-as 65001`	Configures an entry to the BGP neighbor table.
Step 20	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 141.1.1.1 activate`	Enables the exchange of information with a BGP neighbor.

Step 21	**exit address-family**
	Example:
	`Device(config-router-af)# exit address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 22	**end** **Example:** `Device(config-router)# end`	Exits router BGP mode and returns to privileged EXEC mode.

Receiving AS: Configuring a ASBR-P Interface and a IGP

Beginning in privileged EXEC mode complete the following steps to configure a ASBR-P interface and a IGP which is in the receiving AS:

Procedure

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

	Command or Action	Purpose
Step 2	**interface** {*interface-id* | *subinterface-id* | *vlan-id*} **Example:** `Device(config)# interface FortyGigabitEthernet1/0/13`	Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.
Step 3	**no switchport** **Example:** `Device(config-if)# no switchport`	Set the interface to the routed-interface status erases all Layer 2 configurations.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 10 area 0`	Enables OSPF on an interface.
Step 6	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables Multiprotocol Label Switching (MPLS) forwarding of IPv4 and IPv6 packets along normally routed paths for a particular interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Receiving AS: Configuring P

Complete the following tasks to configure the P which is in the AS receiving data from another AS.

Receiving AS: Configuring ASBR-P Interface and IGP

Beginning in user EXEC mode complete the following steps to configure a ASBR-P interface and IGP which is in the receiving AS:

Procedure

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

Step 2	**interface** {*interface-id* | *subinterface-id* | *vlan-id*}
	Example:
	`Device(config)# interface HundredGigE1/0/13`
	`Device(config-if)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

Step 3	**no switchport** **Example:** `Device(config-if)# no switchport`	Set the interface to the routed-interface status erases all Layer 2 configurations.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.2 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 10 area 0`	Enables OSPF on an interface.
Step 6	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables Multiprotocol Label Switching (MPLS) forwarding of IPv4 and IPv6 packets along normally routed paths for a particular interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Step 8	**interface** {*interface-id* | *subinterface-id* | *vlan-id*}
	Example:
	`Device(config)# interface HundredGigE1/0/4`
	`Device(config-if)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

	Command or Action	Purpose
Step 9	**no switchport** **Example:** `Device(config-if)# no switchport`	Set the interface to the routed-interface status and erases all Layer 2 configurations.
Step 10	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 20.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 11	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 10 area 0`	Enables OSPF on an interface.
Step 12	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 and IPv6 packets along normally routed paths for a particular interface.

Step 13	**exit**
	Example:
	`Device(config-if)# exit`
	`Device(config)#`
		Exits interface configuration mode and returns to global configuration mode.

Step 14	**exit** **Example:** `Device(config)# exit`	Exits router configuration mode, and returns to privileged EXEC mode.

Receiving AS: Configuring PE

Complete the following tasks to configure the PE which is in the AS receiving data from another AS.

Configuring VRF for PE2

Beginning in privileged EXEC mode complete the following steps to configure a VRF for a PE:

Procedure

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

	Command or Action	Purpose

Step 2	**vrf definition** *vrf-name*
	Example:
	`Device(config)# vrf definition cu1`
	`Device(config-vrf)#`
		Configures a VRF table and enters VRF configuration mode.

Step 3	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:4`	Creates routing and forwarding tables for a VRF instance.

Step 4	**address-family ipv4**
	Example:
	`Device(config-vrf)# address-family ipv4`
	`Device(config-vrf-af)#`
		The address-family ipv4 command places the device in address family configuration mode, from which you can configure routing sessions that use standard IPv4 address prefixes.

Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 200:2`	Creates a list of export route target communities for the specified VRF.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 200:1`	Creates a list of import route target communities for the specified VRF.

Step 7	**exit-address-family**
	Example:
	`Device(config-vrf-af)#`
	`exit-address-family`
	`Device(config-vrf)#`
		Leaves the address family configuration mode and returns to router configuration mode.

Step 8	**address-familyipv6**
	Example:
	`Device(config-vrf)# address-family ipv6`
	`Device(config-vrf-af)#`
		Places the device in address family configuration mode, from which you can configure routing sessions that use standard IPv6 address prefixes.

Step 9	**route-target export** *route-target-ext-community* **Example:**	Creates a list of export route target communities for the specified VRF.

	Command or Action	Purpose
	`Device(config-vrf-af)# route-target export 200:102`	
Step 10	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 200:101`	Creates a list of import route target communities for the specified VRF.

Step 11	**exit-address-family**
	Example:
	`Device(config-vrf-af)# exit-address-family`
	`Device(config-vrf)#`
		Exits the address family configuration mode and returns to router configuration mode.

Step 12	**exit**
	Example:
	`Device(config-vrf)# exit`
	`Device(config)#`
		Exits the router configuration mode and returns to global configuration mode.

Receiving AS: Configuring PE-CE Interface

Beginning in privileged EXEC mode complete the following steps to configure a PE-CE interface which is in the receiving AS:

Procedure

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

Step 2	**interface** {*interface-id* | *subinterface-id* | *vlan-id*}
	Example:
	`Device(config)# interface FortyGigabitEthernet1/0/5.1`
	`Device(config-subif)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

Step 3	**encapsulation dot1q** *vlan-id* **Example:**	Enables IEEE 802.1Q encapsulation of traffic on a specified interface.

	Command or Action	Purpose
	`Device(config-subif)# encapsulation dot1q 900`	
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-subif)# vrf forwarding cu1`	Associates the VRF with the Layer 3 interface.
Step 5	**ip address** *ip address mask* [**secondary**] **Example:** `Device(config-subif)# ip address 151.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

Step 6	**exit**
	Example:
	`Device(config-subif)# exit`
	`Device(config)#`
		Exits interface configuration mode and returns to global configuration mode.

Receiving AS: Configuring BGP

Beginning in privileged EXEC mode complete the following steps to configure a BGP session on a PE which is in the receiving AS:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number* **Example:** `Device(config-if)# router bgp 65002`	Configures a BGP routing process.
Step 3	**bgp log-neighbor changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.

	Command or Action	Purpose
Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.10.10.10 remote-as 65002`	Configures an entry to the BGP neighbor table.
Step 5	**neighbor** *ip-address* **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.10.10.10 update-source Loopback30`	Allows Cisco IOS software to use a specific operational interface for TCP connections by the BGP sessions.

Step 6	**address-family ipv4**
	Example:
	`Device(config-router)# address-family ipv4`
	`Device(config-router-af)#`
		Enters address family configuration mode for configuring BGP routing sessions that use standard IP Version 4 address prefixes.

Step 7	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 activate`	Enables the exchange of information with a BGP neighbor.

Step 8	**exit-address-family**
	Example:
	`Device(config-router-af)# exit address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 9	**address-family** *vpnv4*
	Example:
	`Device(config-router)# address-family vpnv4`
	`Device(config-router-af)#`
		Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.

Step 10	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 activate`	Enables the exchange of information with a BGP neighbor.

	Command or Action	Purpose
Step 11	**neighbor** { *ip-address* | *ipv6-address* | *peer-group-name* } **send−community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 10.10.10.10 send-community both`	Enables the exchange of information with a BGP neighbor.

Step 12	**exit-address-family**
	Example:
	`Device(config-router-af)# exit-address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 13	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv6 address prefixes.
Step 14	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 activate`	Enables the exchange of information with a BGP neighbor.

Step 15	**exit-address-family**
	Example:
	`Device(config-router-af)# exit-address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 16	**address-family** *vpnv6* **Example:** `Device(config-router)# address-family vpnv6`	Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.
Step 17	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 activate`	Enables the exchange of information with a BGP neighbor.

	Command or Action	Purpose
Step 18	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send−community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 10.10.10.10 send-community both`	Enables the exchange of information with a BGP neighbor.
Step 19	**exit address-family** **Example:** `Device(config-router-af)# exit address-family`	Exits BGP address-family submode.

Step 20	**address-family ipv4 vrf** *vrf-name*]
	Example:
	`Device(config-router)# address-family ipv4 vrf cu1`
	`Device(config-router-af)#`
		Enters address family configuration mode for configuring BGP routing sessions that use standard IP Version 4 address prefixes.

Step 21	**redistribute** *protocol* **Example:** `Device(config-router-af)# redistribute connected`	Redistributes routes from one routing domain into another routing domain.
Step 22	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 151.1.1.2 remote-as 65003`	Configures an entry to the BGP neighbor table.
Step 23	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 151.1.1.2 activate`	Enables the exchange of information with a BGP neighbor.

Step 24	**exit address-family**
	Example:
	`Device(config-router-af)# exit address-family`
	`Device(config-router)#`
		Exits BGP address-family submode.

Step 25	**exit** **Example:**	Exits router configuration mode.

	Command or Action	Purpose
	`Device(config-router)# exit`	

Receiving AS: Configuring a PE-P Interface and IGP

Beginning in user EXEC mode complete the following steps to configure a PE-P interface and IGP which is in the receiving AS:

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** {*interface-id* | *subinterface-id* | *vlan-id*}
	Example:
	`Device(config)# interface FortyGigabitEthernet1/0/4`
	`(config-if)#`
		Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.

Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Set the interface to the routed-interface status erases all Layer 2 configurations.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 20.1.1.2 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 6	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 10 area 0`	Enables OSPF on an interface.

Step 7	**end**
	Example:
	`Device(config-if)# end`
	`Device(config)#`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring MPLS VPN InterAS Option B

The following section provides information about how to configure interAS option B using next-hop-self method and redistribute connected method.

Configuring InterAS Option B using the Next-Hop-Self Method

To configure interAS Option B on ASBRs using the next-hop-self method, complete the following 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 ospf** *process-id* **Example:** `Device(config)# router ospf 1`	Configures an OSPF routing process and assign a process number.
Step 4	**router-id** *ip-address* **Example:** `Device(config)# router-id 4.1.1.1`	Specifies a fixed router ID.
Step 5	**nsr** **Example:** `Device(config-router)# nsr`	Configures OSPF non-stop routing (NSR).
Step 6	**nsf** **Example:** `Device(config-router)# nsf`	Confgures OSPF non-stop forwarding (NSF).
Step 7	**redistribute bgp** *autonomous-system-number* **Example:** `Device(config-router)# redistribute bgp 200`	Redistributes routes from a BGP autonomous system into and OSPF routing process.

	Command or Action	Purpose

Step 8	**passive-interface** *interface-type interface-number*
	Example:
	`Device(config-router)# passive-interface GigabitEthernet 1/0/10`
	`Device(config-router)# passive-interface Tunnel0`
		Disables Open Shortest Path First (OSPF) routing updates on an interface.

Step 9	**network** *ip-address wildcard-mask* **aread** *area-id* **Example:** `Device(config-router)# network 4.1.1.0 0.0.0.255 area 0`	Defines an interface on which OSPF runs and defines the area ID for that interface.
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.
Step 11	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 200`	Configures a BGP routing process.
Step 12	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id 4.1.1.1`	Configures a fixed router ID for the BGP routing process.
Step 13	**bgp log-neighbor changes** **Example:** `Device(config-router)# bgp log-neighbor changes`	Enables logging of BGP neighbor resets.
Step 14	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables advertisement of routing information for address family IPv4.
Step 15	**no bgp default route-target filter** **Example:** `Device(config-router)# no bgp default route-target filter`	Disables automatic BGP route-target community filtering.

	Command or Action	Purpose
Step 16	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 4.1.1.3 remote-as 200`	Configures an entry to the BGP neighbor table.
Step 17	**neighbor** *ip-address* **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 4.1.1.3 update-source Loopback0`	Allows Cisco IOS software to use a specific operational interface for TCP connections by the BGP sessions.
Step 18	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 4.1.1.3 remote-as 300`	Configures an entry to the BGP neighbor table.
Step 19	**address-family** *ipv4* **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for configuring BGP routing sessions that use standard IP Version 4 address prefixes.
Step 20	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.32.1.2 activate`	Enables the exchange of information with a BGP neighbor.
Step 21	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.32.1.2 send-label`	Sends MPLS labels with BGP routes to a neighboring BGP router.
Step 22	**exit address-family** **Example:** `Device(config-router-af)# exit address-family`	Exits BGP address-family submode.
Step 23	**address-family** *vpnv4* **Example:** `Device(config-router)# address-family vpnv4`	Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.

	Command or Action	Purpose
Step 24	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 4.1.1.3 activate`	Enables the exchange of information with a BGP neighbor.
Step 25	**neighbor** *ip-address* **send-community extended** **Example:** `Device(config-router-af)# neighbor 4.1.1.3 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 26	**neighbor** *ip-address* **next-hop-self** **Example:** `Device(config-router-af)# neighbor 4.1.1.3 next-hop-self`	Configure a router as the next hop for a BGP-speaking neighbor. This is the command that implements the next-hop-self method.
Step 27	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.30.1.2 activate`	Enables the exchange of information with a BGP neighbor.
Step 28	**neighbor** *ip-address* **send-community extended** **Example:** `Device(config-router-af)# neighbor 10.30.1.2 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 29	**exit address-family** **Example:** `Device(config-router-af)# exit address-family`	Exits BGP address-family submode.
Step 30	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id 4.1.1.3`	Configures a fixed router ID for the BGP routing process.
Step 31	**bgp log-neighbor changes** **Example:** `Device(config-router)# bgp log-neighbor changes`	Enables logging of BGP neighbor resets.

	Command or Action	Purpose
Step 32	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 4.1.1.1 remote-as 200`	Configures an entry to the BGP neighbor table.
Step 33	**neighbor** *ip-address* **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 4.1.1.1 update-source Loopback0`	Allows Cisco IOS software to use a specific operational interface for TCP connections by the BGP sessions.
Step 34	**address-family** *vpnv4* **Example:** `Device(config-router)# address-family vpnv4`	Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.
Step 35	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 4.1.1.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 36	**neighbor** *ip-address* **send-community extended** **Example:** `Device(config-router-af)# neighbor 4.1.1.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 37	**exit address-family** **Example:** `Device(config-router-af)# exit address-family`	Exits BGP address-family submode.

Configuring InterAS Option B using Redistribute Connected Method

To configure interAS Option B on ASBRs using the redistribute connected method, complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 1`	Configures an OSPF routing process and assign a process number.
Step 4	**router-id** *ip-address* **Example:** `Device(config)# router-id 5.1.1.1`	Specifies a fixed router ID.
Step 5	**nsr** **Example:** `Device(config-router)# nsr`	Configures OSPF non-stop routing (NSR).
Step 6	**nsf** **Example:** `Device(config-router)# nsf`	Confgures OSPF non-stop forwarding (NSF).
Step 7	**redistribute connected** **Example:** `Device(config-router)# redistribute connected`	Redistributes the next hop address of the remote ASBR into the local IGP. This is the command that implements redistribute connected method.

Step 8	**passive-interface** *interface-type interface-number*
	Example:
	`Device(config-router)# passive-interface GigabitEthernet 1/0/10`
	`Device(config-router)# passive-interface Tunnel0`
		Disables Open Shortest Path First (OSPF) routing updates on an interface.

Step 9	**network** *ip-address wildcard-mask* **aread** *area-id* **Example:**	Defines an interface on which OSPF runs and defines the area ID for that interface.

	Command or Action	Purpose
	`Device(config-router)# network 5.1.1.0 0.0.0.255 area 0`	
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.
Step 11	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 300`	Configures a BGP routing process.
Step 12	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id 5.1.1.1`	Configures a fixed router ID for the BGP routing process.
Step 13	**bgp log-neighbor changes** **Example:** `Device(config-router)# bgp log-neighbor changes`	Enables logging of BGP neighbor resets.
Step 14	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables advertisement of routing information for address family IPv4.
Step 15	**no bgp default route-target filter** **Example:** `Device(config-router)# no bgp default route-target filter`	Disables automatic BGP route-target community filtering.
Step 16	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 5.1.1.3 remote-as 300`	Configures an entry to the BGP neighbor table.
Step 17	**neighbor** *ip-address* **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 4.1.1.3 update-source Loopback0`	Allows Cisco IOS software to use a specific operational interface for TCP connections by the BGP sessions.

	Command or Action	Purpose
Step 18	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.30.1.2 remote-as 200`	Configures an entry to the BGP neighbor table.
Step 19	**address-family** *vpnv4* **Example:** `Device(config-router)# address-family vpnv4`	Configures the device in address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.
Step 20	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 5.1.1.3 activate`	Enables the exchange of information with a BGP neighbor.
Step 21	**neighbor** *ip-address* **send-community extended** **Example:** `Device(config-router-af)# neighbor 5.1.1.3 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 22	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.30.1.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 23	**neighbor** *ip-address* **send-community extended** **Example:** `Device(config-router-af)# neighbor 10.30.1.2 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 24	**exit address-family** **Example:** `Device(config-router-af)# exit address-family`	Exits BGP address-family submode.
Step 25	**mpls ldp router-id** *interface-id* [**force**] **Example:** `Device(config-router)# mpls ldp router-id Loopback0 force`	Specifies the preferred interface for determining the LDP router ID.

Configuring MPLS VPN Inter-AS Option AB

The following sections describe how to configure the interAS option AB feature on an ASBR for an MPLS VPN:

Configuring the VRFs on the ASBR Interface for Each VPN Customer

Use the following steps to configure the VRFs on the ASBR interface for each VPN customer so that these VPNs have connectivity over the interAS Option AB network.

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 number* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface to configure and enters interface configuration mode.
Step 4	**ip vrf forwarding***vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface. • The vrf-name argument is the name assigned to a VRF.
Step 5	**end** **Example:** `Device(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the MP-BGP Session Between ASBR Peers

BGP propagates reachability information for VPN-IPv4 prefixes among PE devices by means of the BGP multiprotocol extensions (see RFC 2283, Multiprotocol Extensions for BGP-4), which define support for

address families other than IPv4. Using the extensions ensures that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

Follow the steps in this section to configure the MP-BGP session on the ASBR.

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 100`	Configures a BGP routing process and places the device in router configuration mode. • The as-number argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The ip-address argument specifies the IP address of the neighbor. • The peer-group-name argument specifies the name of a BGP peer group. • The as-number argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The unicast keyword specifies IPv4 unicast address prefixes.

	Command or Action	Purpose
Step 6	**neighbor** { *ip-address* | *peer-group-name* } **activate** **Example:** `Device(config-router-af)# neighbor 192.168.0.1 activate`	Enables the exchange of information with a neighboring device. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** { *ip-address* | *peer-group-name* } **inter-as-hybrid** **Example:** `Device(config-router-af)# neighbor 192.168.0.1 inter-as-hybrid`	Configures eBGP peer device (ASBR) as an Inter-AS Option AB peer. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • If any prefixes are imported into Option AB VRFs, then the imported paths are advertised to this peer. • If any prefixes are received from this peer and are imported into Option AB VRFs, then the imported paths are advertised to iBGP peers. **Note** Advertised routes have RTs that are configured on the VRF. Advertised routes do not have their original RTs.
Step 8	**exit-address-family** **Example:** `Device(config-router)# exit-address-family`	Exits from address family configuration mode.

Configuring the Routing Policy for VPNs that Need Inter-AS Connections

Use the steps in this section to configure VRFs for the VPNs that need Inter-AS connections between ASBR peers, by configuring the appropriate routing policy and Option AB configuration.

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vpn1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.

Step 4	**rd** *route-distinguisher*
	Example:
	`Device(config-vrf)# rd 100:1`
		Creates routing and forwarding tables.
		• 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 autonomous system number: your 32-bit number, for example, 101:3.
		  • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1.

Step 5	**address-family ipv4**
	Example:
	`Device(config-vrf)# address-family ipv4`
		Enters VRF address family configuration mode to specify an address family for a VRF.
		• The **ipv4** keyword specifies an IPv4 address family for a VRF.
		  • 16-bit autonomous system number: your 32-bit number, for example, 101:3.
		  • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1.

Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 100:1`	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 **both** keyword imports routing information from and exports routing

	Command or Action	Purpose
		information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-targetextended community attributes to the VRF list of import, export, or both (import and export) route-target extended communities.
Step 7	**inter-as-hybrid** **Example:** `Device(config-vrf-af)# inter-as-hybrid`	Specifies the VRF as an option AB VRF, which has the following effects: • Routes imported to this VRF can be advertised to option AB peers and VPNv4 iBGP peers. • When routes received from option AB peers and are imported into the VRF, the next hop table ID of the route is set to the table ID of the VRF.
Step 8	**inter-as-hybrid** [**next-hop***ip-address*] **Example:** `Device(config-vrf-af)# inter-as-hybrid next-hop 192.168.1.0`	(Optional) Specifies the next hop IP address to be set on paths that are imported into the VRF and that are received from an Option AB peer. • The next hop context is also set to the VRF, which imports these paths.
Step 9	**end** **Example:** `Device(config-vrf-af)# end`	(Optional) Exits to privileged EXEC mode.

Changing an Inter-AS Option A Deployment to an Option AB Deployment

In an option A deployment, the VRF instances are back-to-back between the ASBR devices and there is direct connectivity between PE devices of different autonomous systems. The PE devices are attached by multiple physical or logical interfaces, each of which is associated with a given VPN (through a VRF instance).

In the Option AB deployment, the different autonomous systems interconnect by using a single MP-BGP session in the global routing table to carry control plane traffic.

Use the following steps to change an MPLS VPN Inter-AS Option A deployment to an Option AB deployment.

1. Configure the MP-BGP session on the ASBR. BGP multiprotocol extensions are used to define support for address families other than IPv4 so that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.
2. Identify the VRFs that need an upgrade from Option A and configure them for Option AB by using the **inter-as-hybrid** command.
3. Use the following steps in this section to remove the configuration for the eBGP (peer ASBR) neighbor.

4. Repeat all the steps in the following procedure to remove the configuration for additional eBGP (peer ASBR) neighbors.

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 100`	Configures a BGP routing process and places the device in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf vpn4`	Configures each VRF that is identified in the MP-BGP session on the ASBR so that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other. • Enters address family configuration mode to specify an address family for a VRF.
Step 5	**no neighbor** { *ip-address* | *peer-group-name* } **Example:** `Device(config-router-af)# no neighbor 192.168.0.1`	Removes the configuration for the exchange of information with the neighboring eBGP (ASBR) device. • The *ip-address* argument specifies the IP address of theneighbor.
Step 6	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits from address family configuration mode.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits to privileged EXEC mode.

Verifying MPLS VPN InterAS Options Configuration

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

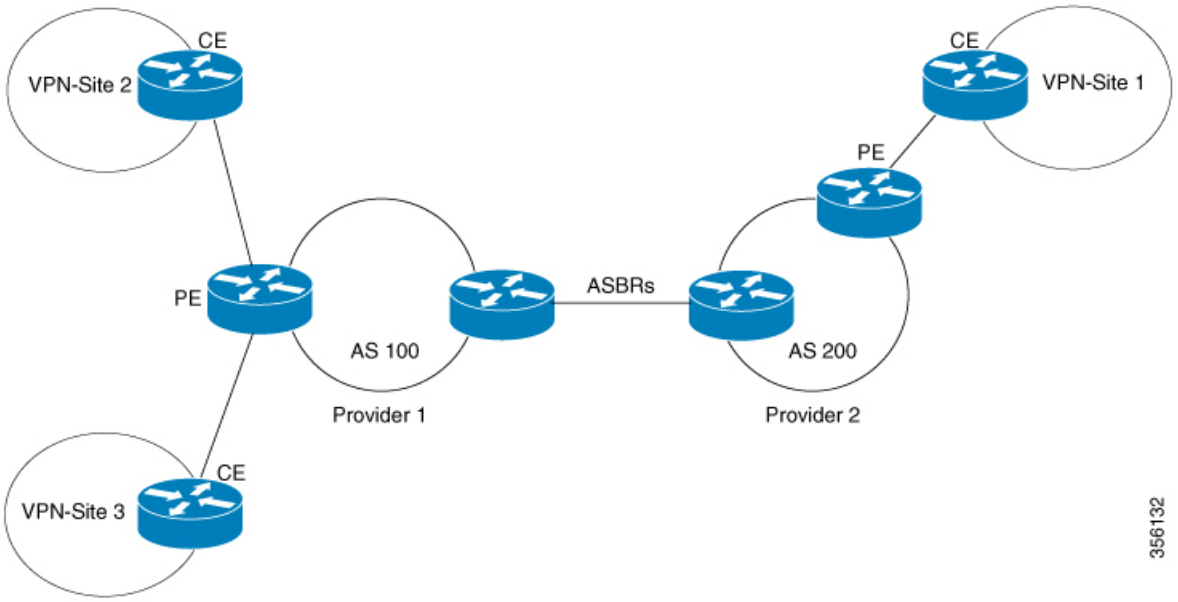
Command	Purpose
ping *ip-address* **source** *interface-type*	Checks the accessibility of devices. Use this command to check the connection between CE1 and CE2 using the loopback interface.
show bgp vpnv4 unicast labels	Displays incoming and outgoing BGP labels.
show mpls forwarding-table	Display the contents of the MPLS Label Forwarding Information Base.
show ip bgp	Displays entries in the BGP routing table.
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** } **route vrf** *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** *interface-type interface-id*	Verifies the route distinguisher (RD) and interface that are configured for the VRF.
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 MPLS VPN InterAS Options

InterAS Option B

Next-Hop-Self Method

Figure 3: Topology for InterAS Option B using Next-Hop-Self Method



Configuration for PE1-P1-ASBR1

PE1	P1	ASBR1
	interface Loopback0 ip address 4.1.1.2 255.255.255.255 ip ospf 1 area 0 interface GigabitEthernet1/0/4 no switchport ip address 10.10.1.2 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp ! interface GigabitEthernet1/0/23 no switchport ip address 10.20.1.1 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp	interface Loopback0 ip address 4.1.1.1 255.255.255.255 ip ospf 1 area 0 interface GigabitEthernet1/0/10 no switchport ip address 10.30.1.1 255.255.255.0 mpls bgp forwarding interface GigabitEthernet1/0/23 no switchport ip address 10.20.1.2 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp router ospf 1 router-id 4.1.1.1 nsr nsf redistribute bgp 200 passive-interface GigabitEthernet1/0/10 passive-interface Tunnel0 network 4.1.1.0 0.0.0.255 area 0 router bgp 200 bgp router-id 4.1.1.1 bgp log-neighbor-changes no bgp default ipv4-unicast no bgp default route-target filter neighbor 4.1.1.3 remote-as 200 neighbor 4.1.1.3 update-source Loopback0 neighbor 10.30.1.2 remote-as 300 ! address-family ipv4 neighbor 10.30.1.2 activate neighbor 10.30.1.2 send-label exit-address-family ! address-family vpnv4 neighbor 4.1.1.3 activate neighbor 4.1.1.3 send-community extended neighbor 4.1.1.3 next-hop-self neighbor 10.30.1.2 activate neighbor 10.30.1.2 send-community extended exit-address-family

PE1	P1	ASBR1
vrf definition Mgmt-vrf ! address-family ipv4 exit-address-family ! address-family ipv6 exit-address-family ! vrf definition vrf1 rd 200:1 route-target export 200:1 route-target import 200:1 route-target import 300:1 ! address-family ipv4 exit-address-family interface Loopback0 ip address 4.1.1.3 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 vrf forwarding vrf1 ip address 192.1.1.1 255.255.255.255 ip ospf 200 area 0 ! interface GigabitEthernet2/0/4 no switchport ip address 10.10.1.1 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp interface GigabitEthernet2/0/9 description to-IXIA-1:p8 no switchport vrf forwarding vrf1 ip address 192.2.1.1 255.255.255.0 ip ospf 200 area 0 router ospf 200 vrf vrf1 router-id 192.1.1.1 nsr nsf redistribute connected redistribute bgp 200 network 192.1.1.1 0.0.0.0 area 0 network 192.2.1.0 0.0.0.255 area 0 router ospf 1 router-id 4.1.1.3 nsr nsf redistribute connected router bgp 200 bgp router-id 4.1.1.3 bgp log-neighbor-changes neighbor 4.1.1.1 remote-as 200 neighbor 4.1.1.1 update-source Loopback0		

PE1	P1	ASBR1
! address-family vpnv4 neighbor 4.1.1.1 activate neighbor 4.1.1.1 send-community extended exit-address-family ! address-family ipv4 vrf vrf1 redistribute connected redistribute ospf 200 maximum-paths ibgp 2 exit-address-family		

Configuration for ASBR2 – P2 – PE2

Table 1:

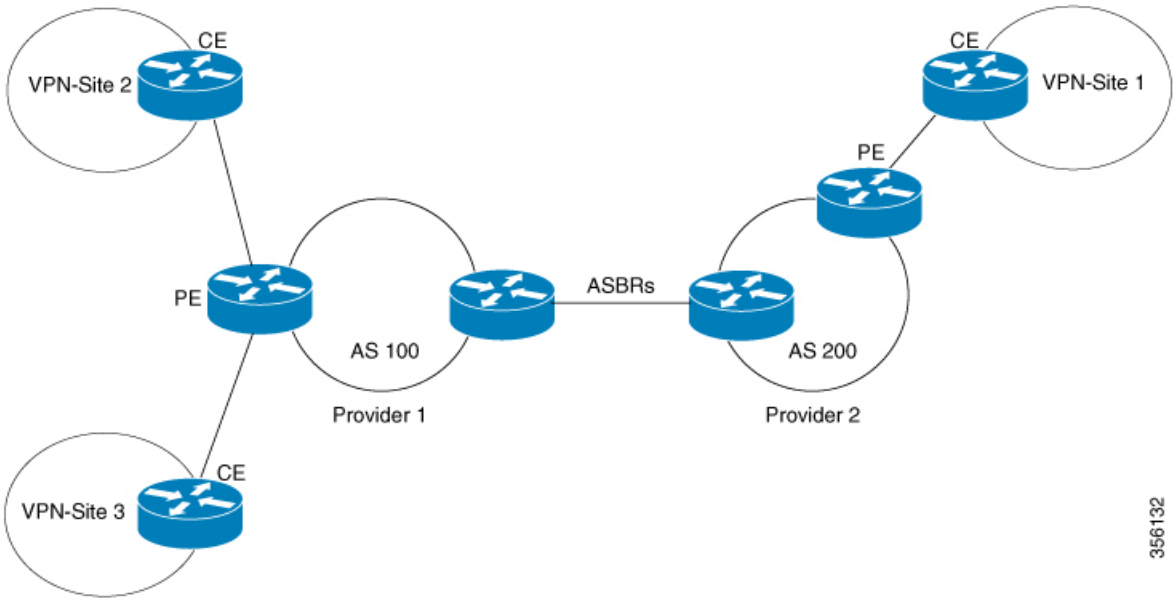
PE2	P2	ASBR2
	interface Loopback0 ip address 5.1.1.2 255.255.255.255 ip ospf 1 area 0 interface GigabitEthernet1/0/1 no switchport ip address 10.50.1.1 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp interface GigabitEthernet2/0/3 no switchport ip address 10.40.1.2 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp	interface Loopback0 ip address 5.1.1.1 255.255.255.255 ip ospf 1 area 0 ! interface GigabitEthernet1/0/37 no switchport ip address 10.30.1.2 255.255.255.0 mpls bgp forwarding interface GigabitEthernet1/0/47 no switchport ip address 10.40.1.1 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp router ospf 1 router-id 5.1.1.1 nsr nsf passive-interface GigabitEthernet1/0/37 passive-interface Tunnel0 network 5.1.1.0 0.0.0.255 area 0 ! router bgp 300 bgp router-id 5.1.1.1 bgp log-neighbor-changes no bgp default ipv4-unicast no bgp default route-target filter neighbor 5.1.1.3 remote-as 300 neighbor 5.1.1.3 update-source Loopback0 neighbor 10.30.1.1 remote-as 200 ! address-family ipv4 neighbor 10.30.1.1 activate neighbor 10.30.1.1 send-label exit-address-family ! address-family vpnv4 neighbor 5.1.1.3 activate neighbor 5.1.1.3 send-community extended neighbor 5.1.1.3 next-hop-self neighbor 10.30.1.1 activate neighbor 10.30.1.1 send-community extended exit-address-family

PE2	P2	ASBR2
vrf definition vrf1 rd 300:1 route-target export 300:1 route-target import 300:1 route-target import 200:1 ! address-family ipv4 exit-address-family interface Loopback0 ip address 5.1.1.3 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 vrf forwarding vrf1 ip address 193.1.1.1 255.255.255.255 ip ospf 300 area 0 interface GigabitEthernet1/0/1 no switchport ip address 10.50.1.2 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp ! interface GigabitEthernet1/0/2 no switchport vrf forwarding vrf1 ip address 193.2.1.1 255.255.255.0 ip ospf 300 area 0 router ospf 300 vrf vrf1 router-id 193.1.1.1 nsr nsf redistribute connected redistribute bgp 300 network 193.1.1.1 0.0.0.0 area 0 network 193.2.1.0 0.0.0.255 area 0 ! router ospf 1 router-id 5.1.1.3 nsr nsf redistribute connected router bgp 300 bgp router-id 5.1.1.3 bgp log-neighbor-changes neighbor 5.1.1.1 remote-as 300 neighbor 5.1.1.1 update-source Loopback0 ! address-family ipv4 neighbor 5.1.1.1 activate neighbor 5.1.1.1 send-label exit-address-family ! address-family vpnv4 neighbor 5.1.1.1 activate		

PE2	P2	ASBR2
neighbor 5.1.1.1 send-community extended exit-address-family ! address-family ipv4 vrf vrf1 redistribute connected redistribute ospf 300 maximum-paths ibgp 2 exit-address-family		

IGP Redistribute Connected Subnets Method

Figure 4: Topology for InterAS Option B using Redistribute Connected Subnets Method



Configuration for PE1-P1-ASBR1

PE1	P1	ASBR1
	interface Loopback0 ip address 4.1.1.2 255.255.255.255 ip ospf 1 area 0 interface GigabitEthernet1/0/4 no switchport ip address 10.10.1.2 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp ! interface GigabitEthernet1/0/23 no switchport ip address 10.20.1.1 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp	router ospf 1 router-id 4.1.1.1 nsr nsf redistribute connected passive-interface GigabitEthernet1/0/10 passive-interface Tunnel0 network 4.1.1.0 0.0.0.255 area 0 router bgp 200 bgp router-id 4.1.1.1 bgp log-neighbor-changes no bgp default ipv4-unicast no bgp default route-target filter neighbor 4.1.1.3 remote-as 200 neighbor 4.1.1.3 update-source Loopback0 neighbor 10.30.1.2 remote-as 300 ! address-family vpnv4 neighbor 4.1.1.3 activate neighbor 4.1.1.3 send-community extended neighbor 10.30.1.2 activate neighbor 10.30.1.2 send-community extended exit-address-family mpls ldp router-id Loopback0 force

PE1	P1	ASBR1
vrf definition Mgmt-vrf ! address-family ipv4 exit-address-family ! address-family ipv6 exit-address-family ! vrf definition vrf1 rd 200:1 route-target export 200:1 route-target import 200:1 route-target import 300:1 ! address-family ipv4 exit-address-family interface Loopback0 ip address 4.1.1.3 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 vrf forwarding vrf1 ip address 192.1.1.1 255.255.255.255 ip ospf 200 area 0 ! interface GigabitEthernet2/0/4 no switchport ip address 10.10.1.1 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp interface GigabitEthernet2/0/9 description to-IXIA-1:p8 no switchport vrf forwarding vrf1 ip address 192.2.1.1 255.255.255.0 ip ospf 200 area 0 router ospf 200 vrf vrf1 router-id 192.1.1.1 nsr nsf redistribute connected redistribute bgp 200 network 192.1.1.1 0.0.0.0 area 0 network 192.2.1.0 0.0.0.255 area 0 router ospf 1 router-id 4.1.1.3 nsr nsf redistribute connected router bgp 200 bgp router-id 4.1.1.3 bgp log-neighbor-changes neighbor 4.1.1.1 remote-as 200 neighbor 4.1.1.1 update-source Loopback0		

PE1	P1	ASBR1
! address-family vpnv4 neighbor 4.1.1.1 activate neighbor 4.1.1.1 send-community extended exit-address-family ! address-family ipv4 vrf vrf1 redistribute connected redistribute ospf 200 maximum-paths ibgp 2 exit-address-family		

Configuration for ASBR2 – P2 – PE2

PE2	P2	ASBR2
	interface Loopback0 ip address 5.1.1.2 255.255.255.255 ip ospf 1 area 0 interface GigabitEthernet1/0/1 no switchport ip address 10.50.1.1 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp interface GigabitEthernet2/0/3 no switchport ip address 10.40.1.2 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp	router ospf 1 router-id 5.1.1.1 nsr nsf redistribute connected passive-interface GigabitEthernet1/0/10 passive-interface Tunnel0 network 5.1.1.0 0.0.0.255 area 0 router bgp 300 bgp router-id 5.1.1.1 bgp log-neighbor-changes no bgp default ipv4-unicast no bgp default route-target filter neighbor 5.1.1.3 remote-as 300 neighbor 5.1.1.3 update-source Loopback0 neighbor 10.30.1.1 remote-as 200 ! address-family vpnv4 neighbor 5.1.1.3 activate neighbor 5.1.1.3 send-community extended neighbor 10.30.1.1 activate neighbor 10.30.1.1 send-community extended exit-address-family mpls ldp router-id Loopback0 force

PE2	P2	ASBR2
vrf definition vrf1 rd 300:1 route-target export 300:1 route-target import 300:1 route-target import 200:1 ! address-family ipv4 exit-address-family interface Loopback0 ip address 5.1.1.3 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 vrf forwarding vrf1 ip address 193.1.1.1 255.255.255.255 ip ospf 300 area 0 interface GigabitEthernet1/0/1 no switchport ip address 10.50.1.2 255.255.255.0 ip ospf 1 area 0 mpls ip mpls label protocol ldp ! interface GigabitEthernet1/0/2 no switchport vrf forwarding vrf1 ip address 193.2.1.1 255.255.255.0 ip ospf 300 area 0 router ospf 300 vrf vrf1 router-id 193.1.1.1 nsr nsf redistribute connected redistribute bgp 300 network 193.1.1.1 0.0.0.0 area 0 network 193.2.1.0 0.0.0.255 area 0 ! router ospf 1 router-id 5.1.1.3 nsr nsf redistribute connected router bgp 300 bgp router-id 5.1.1.3 bgp log-neighbor-changes neighbor 5.1.1.1 remote-as 300 neighbor 5.1.1.1 update-source Loopback0 ! address-family ipv4 neighbor 5.1.1.1 activate neighbor 5.1.1.1 send-label exit-address-family ! address-family vpnv4 neighbor 5.1.1.1 activate		

PE2	P2	ASBR2
neighbor 5.1.1.1 send-community extended exit-address-family ! address-family ipv4 vrf vrf1 redistribute connected redistribute ospf 300 maximum-paths ibgp 2 exit-address-family		

InterAS OptionAB

The following example displays the topology and the configuration on each device:

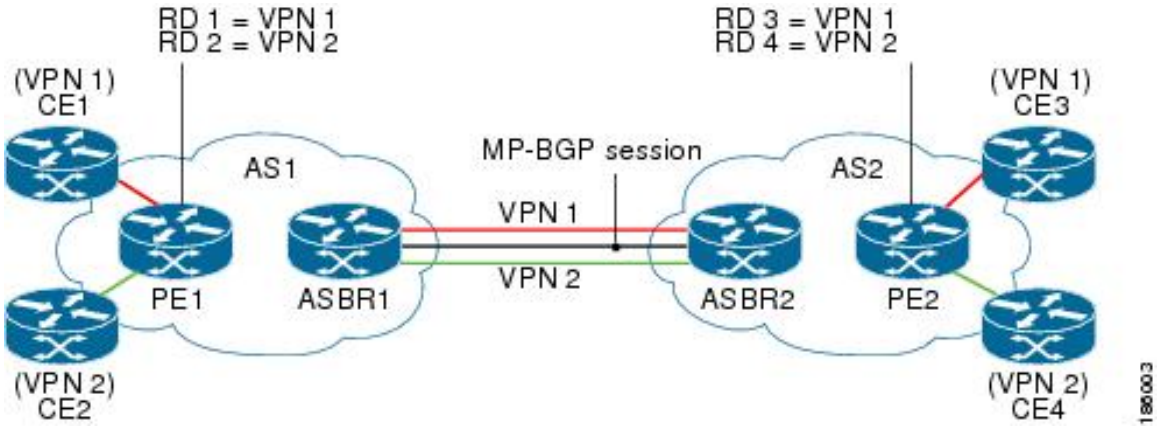


Table 2:

PE1 Config	P1 Config	ASBR1 Config	ASBR2 Config	PE2 Config
	interface Loopback0 ip address 2.2.2.2 255.255.255.255 ! interface TenGigabitEthernet1/1 ip address 10.1.1.2 255.255.255.0 mpls ip ! interface TenGigabitEthernet1/2 no ip address ! interface TenGigabitEthernet1/3 ip address 20.1.1.1 255.255.255.0 mpls ip ! router ospf 1 router-id 2.2.2.2 network 2.2.2.2 0.0.0.0 area 0 network 10.1.1.0 0.0.0.255 area 0 network 20.1.1.0 0.0.0.255 area 0 !			

PE1 Config	P1 Config	ASBR1 Config	ASBR2 Config	PE2 Config
ip vrf cust-1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip vrf cust-2 rd 100:2 route-target export 100:2 route-target import 100:2 ! interface Loopback0 ip address 1.1.1.1 255.255.255.255 ! interface Loopback1 ip address 11.11.11.11 255.255.255.255 ! interface Loopback2 ip address 12.12.12.12 255.255.255.255 ! ! interface HundredGigE1/0/1/1 no switchport ip address 10.1.1.1 255.255.255.0 mpls ip ! ! interface HundredGigE1/0/1/4 no switchport no ip address ! interface HundredGigE1/0/1/4.100 encapsulation dot1Q 100 ip vrf forwarding cust-1 ip address 11.1.1.1 255.255.255.0 ! interface HundredGigE1/0/1/4.101 encapsulation		ip vrf cust-1 rd 100:10001 route-target export 100:1 route-target import 100:1 route-target import 200:1 inter-as-hybrid next-hop 160.1.1.2 ! ip vrf cust-2 rd 100:20001 route-target export 100:2 route-target import 100:2 route-target import 200:2 inter-as-hybrid next-hop 170.1.1.2 ! ! interface Loopback0 ip address 3.3.3.3 255.255.255.255 ! ! interface TwentyFiveGigE1/0/3 no switchport ip address 20.1.1.2 255.255.255.0 mpls ip ! ! interface TwentyFiveGigE1/0/10.10 encapsulation dot1Q 10 ip address 150.1.1.1 255.255.255.0 mpls bgp forwarding ! interface TwentyFiveGigE1/0/10.20 encapsulation dot1Q 20 ip vrf forwarding cust-1 ip address 160.1.1.1 255.255.255.0 ! interface	ip vrf cust-1 rd 200:10001 route-target export 200:1 route-target import 200:1 route-target import 100:1 inter-as-hybrid next-hop 160.1.1.1 ! ip vrf cust-2 rd 200:20001 route-target export 200:2 route-target import 200:2 route-target import 100:2 inter-as-hybrid next-hop 170.1.1.1 ! ! interface Loopback0 ip address 4.4.4.4 255.255.255.255 ! ! interface TwentyFiveGigE1/0/2 no switchport ip address 30.1.1.1 255.255.255.0 mpls ip ! ! interface TwentyFiveGigE1/0/10.10 encapsulation dot1Q 10 ip address 150.1.1.2 255.255.255.0 mpls bgp forwarding ! interface TwentyFiveGigE1/0/10.20 encapsulation dot1Q 20 ip vrf forwarding cust-1 ip address 160.1.1.2 255.255.255.0	ip vrf cust-1 rd 200:1 route-target export 200:1 route-target import 200:1 route-target import 100:1 ! ip vrf cust-2 rd 200:2 route-target export 200:2 route-target import 200:2 route-target import 100:2 ! interface Loopback0 ip address 5.5.5.5 255.255.255.255 ! interface Loopback1 ip address 55.55.55.55 255.255.255.255 ! interface Loopback2 ip address 56.56.56.56 255.255.255.255 ! ! interface HundredGigE1/0/1/1.200 encapsulation dot1Q 200 ip vrf forwarding cust-1 ip address 55.1.1.1 255.255.255.0 ! interface HundredGigE1/0/1/1.201 encapsulation dot1Q 201 ip vrf forwarding cust-2 ip address 56.1.1.1 255.255.255.0 ! interface HundredGigE1/0/1/3 no switchport ip address

PE1 Config	P1 Config	ASBR1 Config	ASBR2 Config	PE2 Config
dot1Q 101 ip vrf forwarding cust-2 ip address 12.1.1.1 255.255.255.0 ! ! router ospf 2 vrf cust-1 router-id 11.11.11.11 network 11.1.1.0 0.0.0.255 area 0 network 11.11.11.11 0.0.0.0 area 0 ! router ospf 3 vrf cust-2 router-id 12.12.12.12 network 12.1.1.0 0.0.0.255 area 0 network 12.12.12.12 0.0.0.0 area 0 ! router ospf 1 router-id 1.1.1.1 network 1.1.1.1 0.0.0.0 area 0 network 10.1.1.0 0.0.0.255 area 0 ! router bgp 100 bgp router-id 1.1.1.1 bgp log-neighbor- changes neighbor 3.3.3.3 remote-as 100 neighbor 3.3.3.3 update- source Loopback0 ! address-family vpnv4 neighbor 3.3.3.3 activate neighbor 3.3.3.3 send- community extended exit-address-family ! address-family ipv4 vrf cust-1 redistribute connected		TwentyFiveGigE1/0/10.30 encapsulation dot1Q 30 ip vrf forwarding cust-2 ip address 170.1.1.1 255.255.255.0 ! router ospf 1 router-id 3.3.3.3 network 3.3.3.3 0.0.0.0 area 0 network 20.1.1.0 0.0.0.255 area 0 ! router bgp 100 bgp router-id 3.3.3.3 bgp log-neighbor- changes neighbor 1.1.1.1 remote- as 100 neighbor 150.1.1.2 remote-as 200 ! address-family ipv4 redistribute connected neighbor 1.1.1.1 activate neighbor 150.1.1.2 activate exit-address-family ! address-family vpnv4 neighbor 1.1.1.1 activate neighbor 1.1.1.1 send- community both neighbor 150.1.1.2 activate neighbor 150.1.1.2 send- community both neighbor 150.1.1.2 inter- as-hybrid exit-address-family ! address-family ipv4 vrf cust-1 redistribute connected exit-address-family	! interface TwentyFiveGigE1/0/10.30 encapsulation dot1Q 30 ip vrf forwarding cust-2 ip address 170.1.1.2 255.255.255.0 ! router ospf 1 router-id 4.4.4.4 network 4.4.4.4 0.0.0.0 area 0 network 30.1.1.0 0.0.0.255 area 0 ! router bgp 200 bgp router-id 4.4.4.4 bgp log-neighbor- changes neighbor 5.5.5.5 remote- as 200 neighbor 150.1.1.1 remote-as 100 ! address-family ipv4 neighbor 5.5.5.5 activate neighbor 150.1.1.1 activate exit-address-family ! address-family vpnv4 neighbor 5.5.5.5 activate neighbor 5.5.5.5 send-community both neighbor 150.1.1.1 activate neighbor 150.1.1.1 send-community both neighbor 150.1.1.1 inter-as-hybrid ' exit-address-family ! address-family ipv4 vrf cust-1 redistribute	30.1.1.2 255.255.255.0 mpls ip ! router ospf 2 vrf cust-1 router-id 55.55.55.55 network 55.1.1.0 0.0.0.255 area 0 network 55.55.55.55 0.0.0.0 area 0 ! router ospf 3 vrf cust-2 router-id 56.56.56.56 network 56.1.1.0 0.0.0.255 area 0 network 56.56.56.56 0.0.0.0 area 0 ! router ospf 1 router-id 5.5.5.5 network 5.5.5.5 0.0.0.0 area 0 network 30.1.1.0 0.0.0.255 area 0 ! router bgp 200 bgp router-id 5.5.5.5 bgp log-neighbor-changes neighbor 4.4.4.4 remote-as 200 neighbor 4.4.4.4 update-source Loopback0 ! address-family vpnv4 neighbor 4.4.4.4 activate neighbor 4.4.4.4 send-community extended exit-address-family ! address-family ipv4 vrf cust-1 redistribute connected redistribute ospf 2 maximum-paths ibgp

PE1 Config	P1 Config	ASBR1 Config	ASBR2 Config	PE2 Config
redistribute ospf 2 maximum-paths ibgp 2 exit-address-family ! address-family ipv4 vrf cust-2 redistribute connected redistribute ospf 3 maximum-paths ibgp 2 exit-address-family		! address-family ipv4 vrf cust-2 redistribute connected exit-address-family !	connected exit-address-family ! address-family ipv4 vrf cust-2 redistribute connected exit-address-family !	2 exit-address-family ! address-family ipv4 vrf cust-2 redistribute connected redistribute ospf 3 maximum-paths ibgp 2 exit-address-family !

Additional References for MPLS VPN InterAS Options

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the MPLS Commands section of the*Command Reference (Catalyst 9400 Series Switches)*

Feature History for MPLS VPN InterAS Options

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	MPLS VPN InterAS Option B	InterAS Options use iBGP and eBGP peering to allow VPNs in different AS to communicate with each other. In an interAS option B network, ASBR ports are connected by one or more interfaces that are enabled to receive MPLS traffic.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.1.1	MPLS VPN InterAS Option A	MPLS VPN InterAS Option A is the simplest to configure of the available InterAS Options. This option provides back to back virtual routing and forwarding (VRF) connectivity. Here, MPLS VPN providers exchange routes across VRF interfaces.
Cisco IOS XE Amsterdam 17.3.1	MPLS VPN InterAS Option AB	MPLS VPN InterAS Option AB enables different autonomous systems to interconnect by using a single Multiprotocol Border Gateway Protocol (MP-BGP) session, which is enabled globally on the router.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.