

Configuring Multicast VPN Inter-AS Support

The Multicast VPN Inter-AS Support feature enables Multicast Distribution Trees (MDTs) used for Multicast VPNs (MVPNs) to span multiple autonomous systems. Benefits include increased multicast coverage to customers that require multicast to span multiple service providers in a Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) service with the flexibility to support all options described in RFC 4364. Additionally, the Multicast VPN Inter-AS Support feature can be used to consolidate an existing MVPN service with another MVPN service, such as the case with a company merger or acquisition.

- Finding Feature Information, on page 1
- Prerequisites for Configuring Multicast VPN Inter-AS Support, on page 1
- Restrictions for Configuring Multicast VPN Inter-AS Support, on page 2
- Information About Multicast VPN Inter-AS Support, on page 2
- How to Configure Multicast VPN Inter-AS Support, on page 18
- Configuration Examples for Multicast VPN Inter-AS Support, on page 28
- Additional References, on page 61
- Feature Information for Configuring Multicast VPN Inter-AS Support, on page 62

Finding Feature Information

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

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

Prerequisites for Configuring Multicast VPN Inter-AS Support

- You understand IP multicast concepts and configuration tasks.
- You understand MVPN concepts and configuration tasks.
- You understand Border Gateway Protocol (BGP) concepts and configuration tasks.
- You understand MPLS Layer 3 VPN concepts and configuration tasks.

Restrictions for Configuring Multicast VPN Inter-AS Support

The Multicast VPN Inter-AS Support feature requires that all routers in the core be configured for Protocol Independent Multicast (PIM) Source Specific Multicast (SSM). Protocol Independent Multicast sparse mode (PIM-SM) and bidirectional PIM (bidir-PIM) are not supported.

Information About Multicast VPN Inter-AS Support

MVPN Inter-AS Support Overview

As a general concept, MVPN inter-AS support enables service providers to provide multicast connectivity to VPN sites that span multiple autonomous systems. There are two types of MVPN inter-AS deployment scenarios:

- Single provider inter-AS--A service provider whose internal network consists of multiple autonomous systems.
- Intraprovider inter-AS--Multiple service providers that need to coordinate their networks to provide inter-AS support.

The extensions added to support the Multicast VPN Inter-AS Support feature enable MDTs used for MVPNs to span multiple autonomous systems.

Benefits of MVPN Inter-AS Support

The MVPN Inter-AS Support feature provides the following benefits to service providers:

- Increased multicast coverage to customers that require multicast to span multiple services providers in an MPLS Layer 3 VPN service with the flexibility to support all options described in RFC 4364.
- The ability to consolidate an existing MVPN service with another MVPN service, such as the case with a company merger or acquisition.

MVPN Inter-AS Support Implementation Requirements

The Multicast VPN Inter-AS Support feature was implemented in the software in accordance to the following requirements:

- To achieve parity with unicast inter-AS support, the software must support the following inter-AS options for MVPN (as defined in RFC 4364):
    - Option A--Back-to-back VPN routing and forwarding (VRF) instances at the Autonomous System Border Router (ASBR) provider edge (PE) routers

The Option A model assumes direct connectivity between PE routers of different autonomous systems. The PE routers are attached by multiple physical or logical interfaces, each of which is associated with a given VPN (through a VRF instance). Each PE router, therefore, treats the adjacent PE router like a customer edge (CE) router, and the standard Layer 3 MPLS VPN mechanisms are used for route redistribution with each

autonomous system; that is, the PEs use exterior BGP (eBGP) to distribute unlabeled IPv4 addresses to each other.



Note Option A allows service providers to isolate each autonomous system from the other, which provides better control over routing exchanges and security between the two networks. Option A, however, is considered the least scalable of all the inter-AS connectivity options.

- • Option B--VPNv4 route exchange between ASBRs

In the Option B model, the PE routers use interior BGP (iBGP) to redistribute labeled VPNv4 routes either to an ASBR or to a route reflector of which an ASBR is a client. ASBRs then use multiprotocol eBGP (MP-eBGP) to advertise VPNv4 routes into the local autonomous system.

MP-eBGP provides the functionality to advertise VPNv4 prefix and label information across the service provider boundaries. The advertising ASBR router replaces the two-level label stack (which it uses to reach the originating PE router and VPN destination in the local autonomous system) with a locally allocated label before advertising the VPNv4 route. This replacement is necessary because the next-hop attribute of all routes advertised between the two service providers is reset to the ASBR router's peering address, so the ASBR router becomes the termination point of the label-switched path (LSP) for the advertised routes. To preserve the LSP between ingress and egress PE routers, the ASBR router must allocate a local label that may be used to identify the label stack of the route within the local VPN network. This newly allocated label is set on packets sent toward the prefix from the adjacent service provider.



Note Option B enables service providers to isolate both autonomous systems with the added advantage that it scales to a higher degree than Option A.

- • Option C--Exchange of VPNv4 routes between route reflectors (RRs) using multihop eBGP peering sessions

The Option C model combines MP-eBGP exchange of VPNv4 routes between RRs of different autonomous systems with the next hops for these routes exchanged between corresponding ASBR routers. In the Option C model, VPNv4 routes are neither maintained nor distributed by the ASBRs. ASBRs must maintain labeled IPv4 /32 routes to the PE routers within its autonomous system and use eBGP to distribute these routes to other autonomous systems. ASBRs in any transit autonomous systems will also have to use eBGP to pass along the labeled /32 routes. The result is the creation of a LSP from the ingress PE router to the egress PE router.

Because RRs of different autonomous systems will not be directly connected, multihop functionality is required to allow for the establishment of the MP-eBGP peering sessions. The exchange of next hops is necessary because the RRs do not reset the next-hop attribute of the VPNv4 routes when advertising them to adjacent autonomous systems because they do not want to attract the traffic for the destinations that they advertise. They are not the original endpoint--just a relay station between the source and receiver PEs. The PE router next-hop addresses for the VPNv4 routes, thus, are exchanged between ASBR routers. The exchange of these addresses between autonomous systems can be accomplished by redistributing the PE router /32 addresses between the autonomous systems or by using BGP label distribution.

Note Option C normally is deployed only when each autonomous system belongs to the same overall authority, such as a global Layer 3 MPLS VPN service provider with autonomous systems in different regions of the world. Option B is equally suited for this purpose and is also deployed in networks where autonomy between different regions is desired.

- The Cisco software must support inter-AS MDTs. An inter-AS MDT is an MDT that extends across autonomous system boundaries. In the context of MVPN, because MVPN packets are encapsulated when being forwarded between ASBRs, an inter-AS MDT is needed (for Option B and Option C) to extend the MDT across the boundaries of the autonomous system.

Limitations That Prevented Option B and Option C Support

Prior to the extensions introduced in association with the Multicast VPN Inter-AS Support feature, limitations existed that prevented MVPN inter-AS support for Option B and Option C. These limitations were related to the following areas:

- Supporting reverse path forwarding (RPF) for inter-AS sources (applicable mainly to Option B)
  - When a PE router sends a PIM join (source PE address, MDT group address) for the default MDT, each P router in the path between the source and the destination PE routers must perform an RPF check on the source. Because Interior Gateway Protocol (IGP) routes (which would include the routes to source PE routers in remote autonomous systems) are not leaked across autonomous systems, the P routers in the receiving autonomous system were unable to perform an RPF check.
  - When a PIM join is received in an MVPN, an IP lookup is performed in the VRF to find the next hop toward the destination. This destination must be a PIM neighbor that can be reached through the MDT tunnel interface. However, because ASBRs change the next hop of the originating PE router for a given MDT group, the originating source address would be lost, and the RPF check at the PE router would fail.

Note In typical Option C inter-AS deployments, the limitation related to supporting RPF for MVPN inter-AS support was not applicable because the RRs store all VPNv4 routes.

- Supporting an inter-AS MDT (applicable to Option B and Option C)
  - The default MDT relies on the ability of the PE routers to join the default multicast group. The source of the group is the originating PE router address used for MP-BGP peering. Prior to the extensions introduced in association with the Multicast VPN Inter-AS Support feature, this address could not be reached between autonomous systems because IGP routes could not be distributed across the autonomous systems. The default MDT for inter-AS MVPN, thus, could not be established.

MVPN Inter-AS Support for Option A

The limitations that prevented support for MVPN inter-AS support Options B and C have never applied to Option A for the following reasons:

- For Option A, native IP forwarding is used by the PE routers between autonomous systems; therefore, Option A does not require support for inter-AS MDTs.

- For Option A, the MDT is limited to one autonomous system; therefore, the issues associated with managing MDT group addresses between autonomous systems and RPF for inter-AS sources never applied to Option A.

Note Because Option A requires that one physical or logical interface be configured for each VRF, Option A is considered the least scalable MVPN inter-AS solution.

MVPN Inter-AS Support Solution for Options B and C

The following extensions introduced in association with the MVPN Inter-AS Support feature resolve the MVPN inter-AS protocol limitations related to supporting RPF and inter-AS MDTs in Option B and C deployments:

- BGP connector attribute in MP-BGP--This attribute helps preserve the identity of a PE router originating a VPNv4 prefix. This BGP extension helps solve the challenge of supporting RPF to sources in a remote autonomous system.
- BGP MDT Subaddress Family Identifier (SAFI)--This identifier helps ASBRs RPF to source PEs in a remote autonomous systems. The BGP MDT SAFI also helps ASBRs and receiver PEs insert the RPF Vector needed to build an inter-AS MDT to source PEs in remote autonomous systems.

BGP Connector Attribute

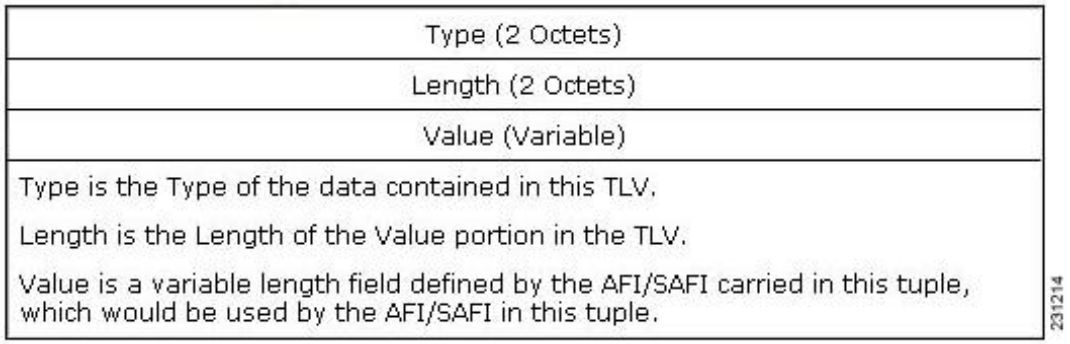
In an adjacent autonomous system, a PE router that wants to join a particular source of the default MDT for a given MVPN must know the originator's address of the source PE router. This presents some challenges for Option B inter-AS deployments because the originator next hop for VPNv4 routes is rewritten at one or more points in the network. To solve this limitation, each VPNv4 route must carry a new attribute (the BGP connector attribute) that defines the route's originator.

The BGP connector attribute is a transitive attribute that stores the PE router that originated a VPNv4 prefix. In a local autonomous system, the BGP connector attribute is the same as the next hop attribute. When advertised to other ASBRs in VPNv4 advertisements (as is the case in Option B), the value of the BGP connector attribute is preserved even after the next hop attribute is rewritten by ASBRs. The BGP connector attribute is a critical component of the MVPN inter-AS solution, helping to enable RPF to sources in remote autonomous systems.

Note The BGP connector attribute also helps ASBRs and receiver PEs insert the RPF Vector needed to build the inter-AS MDT for source PEs in remote autonomous systems. For more information about RPF Vectors, see the PIM RPF Vector, on page 6 section.

The format of the BGP connector attribute is shown in the figure.

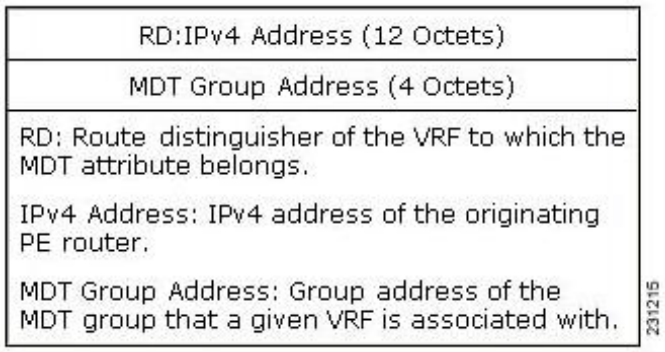
Figure 1: BGP Connector Attribute



BGP MDT SAFI Updates for MVPN Inter-AS Support

The BGP MDT SAFI is specifically designed to carry the address of the source PE router to which a PIM join should be sent for the MDT group contained in the PIM join. The format of the Network Layer Reachability Information (NLRI) carried in this SAFI is {RD:PE-IP-address}. The BGP MDT SAFI is capable of being advertised across autonomous system boundaries. Each MDT group is carried in the MP_REACH attribute using the format shown in the figure.

Figure 2: MDT SAFI Format



When RRs and MP-eBGP peerings are used, the advertisement of the BGP MDT SAFI is independent of the advertisement of VPNv4 routes. BGP MDT SAFI advertisements, however, are processed and filtered like VPNv4 advertisements.

ASBRs store the path advertised in BGP MDT SAFI updates in a separate table. How the BGP MDT SAFI is advertised determines the RPF path to the PE router that originated the advertisement.

PEs also store the BGP MDT SAFI update in a separate table. PE routers use the information contained in the BGP MDT SAFI to determine the ASBR that is the exit router to the source PE router in an adjacent autonomous system.

PIM RPF Vector

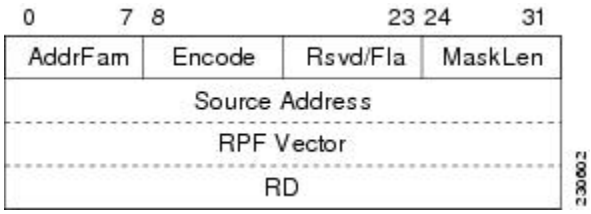
Normally, in an MVPN environment, PIM sends join messages containing the IP address of upstream PE routers that are sources of a given MDT group. To be able to perform RPF checks, however, P routers must have IPv4 reachability to source PE routers in remote autonomous systems. This behavior is not the case with inter-AS Options B and C because the autonomous systems do not exchange any of their IGP routes, including those of their local PE routers. However, P routers do have reachability to the BGP next hop of the BGP MDT

update received with the BGP MDT SAFI updates at the PE routers. Therefore, if the PE routers add the remote PE router IP address (as received within the BGP MDT SAFI) and the BGP next-hop address of this address within the PIM join, the P routers can perform an RPF check on the BGP next-hop address rather than the original PE router address, which, in turn, allows the P router to forward the join toward the ASBR that injected the MDT SAFI updates for a remote autonomous system. This functionality is generally referred to as the *PIM RPF Vector* ; the actual vector that is inserted into PIM joins is referred to as the *RPF Vector* or the *Proxy Vector* . The PIM RPF Vector, therefore, enables P routers to determine the exit ASBR to a source PE router in a remote autonomous system. Having received the join that contains a RPF Vector, an ASBR can then determine that the next-hop address is in fact itself and can perform an RPF check based on the originating PE router address carried in the PIM join.

When configured on PE routers using the **ip multicast rpf proxy vector** command, the RPF Vector is encoded as a part of the source address in PIM join and prune messages. The RPF Vector is the IGP next hop for PIM RPF neighbor in PIM join and prune messages, which is typically the exit ASBR router to a prefix in a remote autonomous system.

The format of this PIM RPF Vector encoding in PIM join and prune messages is shown in the figure.

Figure 3: PIM RPF Vector Encoded in PIM Join and Prune Messages



Note RPF Vectors can be used natively in an IP environment (that is, in a non-VPN environment). For more information about the use of RPF Vectors in a native environment, see RFC 5496, The Reverse Path Forwarding (RPF) Vector TLV .

Originators of an RPF Vector

Whether or not a PE router originates an RPF Vector is determined by configuration; that is, the **ip multicast rpf proxy vector** command must be configured on all PE routers in order for an RPF Vector to be originated. The PE router that originates an RPF Vector always performs an RPF lookup on the source. When a PE router performs an RPF lookup on a source, the PE router learns the origin of an RPF Vector in one of the following ways:

- In an MVPN network environment, the RPF Vector is learned from BGP MDT SAFI updates.
- In a native IP network environment, the RPF Vector is learned from either IP unicast routing (AFI=1, SAFI=1) or IP multicast reverse-path information (AFI=1, SAFI=2).

Note For more information about the use of RPF Vectors in a native environment, see RFC 5496, The Reverse Path Forwarding (RPF) Vector TLV .

Note Routers that understand the RPF Vector format advertise the RPF Vector in PIM hello messages.

Recipients of an RPF Vector

When a router receives a PIM join that contains an RPF Vector, that router stores the RPF Vector so that it can generate a PIM join to the exit ASBR router. P routers, thus, learn the RPF Vector from PIM joins. The RPF Vector is advertised to all P routers in the core. If multiple RPF Vectors are received by a router, the RPF Vector with the lower originator address is used. When the RPF Vector is present, it takes priority; as a result, RPF checks are triggered periodically to readvertise RPF Vectors upstream. If a router receives an RPF Vector that references a local interface (typically an ASBR), the RPF Vector is discarded and a normal RPF lookup is performed.

ASBR Receipt of an RPF Vector

When an ASBR receives an RPF Vector, it typically references a local interface (most likely a loopback interface); in which case, the RPF Vector is discarded and a normal RPF lookup is performed. If the RD type is 2, the ASBR performs an RPF lookup in the BGP MDT table that is built from the BGP MDT SAFI updates; this type of RPF lookup uses both the RD and the source PE address contained in the PIM join.

Interoperability with RPF Vector

A new PIM hello option is introduced along with the PIM RPF Vector extension to determine if the upstream router is capable of parsing the new encoding. An RPF Vector is included in PIM messages only when all PIM neighbors on an RPF interface support it.

MDT Address Family in BGP for Multicast VPN Inter-AS Support

The **mdt** keyword has been added to the **address-family ipv4** command to configure an MDT address-family session.

Supported Policy

The following policy configuration parameters are supported under the BGP MDT SAFI:

- Mandatory attributes and well-known attributes, such as the AS-path, multi-exit discriminator MED, BGP local-preference, and next hop attributes.
- Standard communities, community-lists, and route-maps.

Guidelines for Configuring MDT Address Family Sessions on PE Routers for MVPN Inter-AS Support

When configuring routers for MVPN inter-AS support, follow these guidelines:

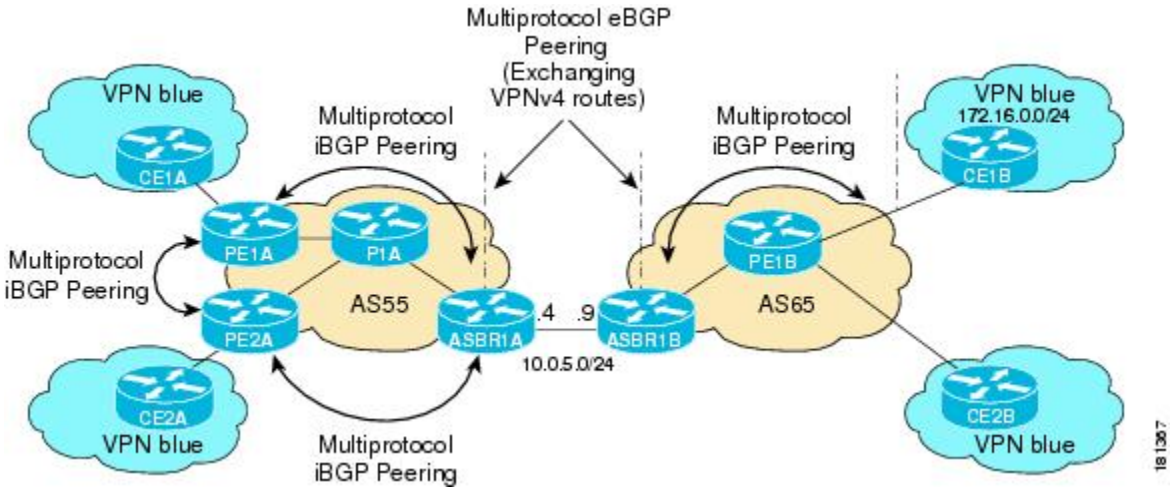
- For MVPN inter-AS Option A, BGP MDT address-family peering sessions are not required between the PE routers because native IP forwarding is used by the PE routers. For option A, BGP MDT peering sessions are only required for intra-AS VPN peering sessions.
- For MVPN inter-AS Option B, BGP MDT address-family peering sessions are only required between the PEs and ASBRs. In the Option B inter-AS case where PE routers use iBGP to redistribute labeled VPNv4 routes to RRs of which ASBRs are clients, then BGP MDT address-family peering sessions are required between the PEs, ASBRs, and RRs.

- For MVPN inter-AS Option C, BGP MDT address-family peering sessions are only required between the PEs and RRs.

MVPN Inter-AS MDT Establishment for Option B

This section describes the sequence of events that leads to the establishment of an inter-AS MDT between the autonomous systems in the sample inter-AS Option B topology illustrated in the following figure. For this topology, assume that all the routers have been configured properly to support all extensions associated with the Multicast VPN Inter-AS Support feature.

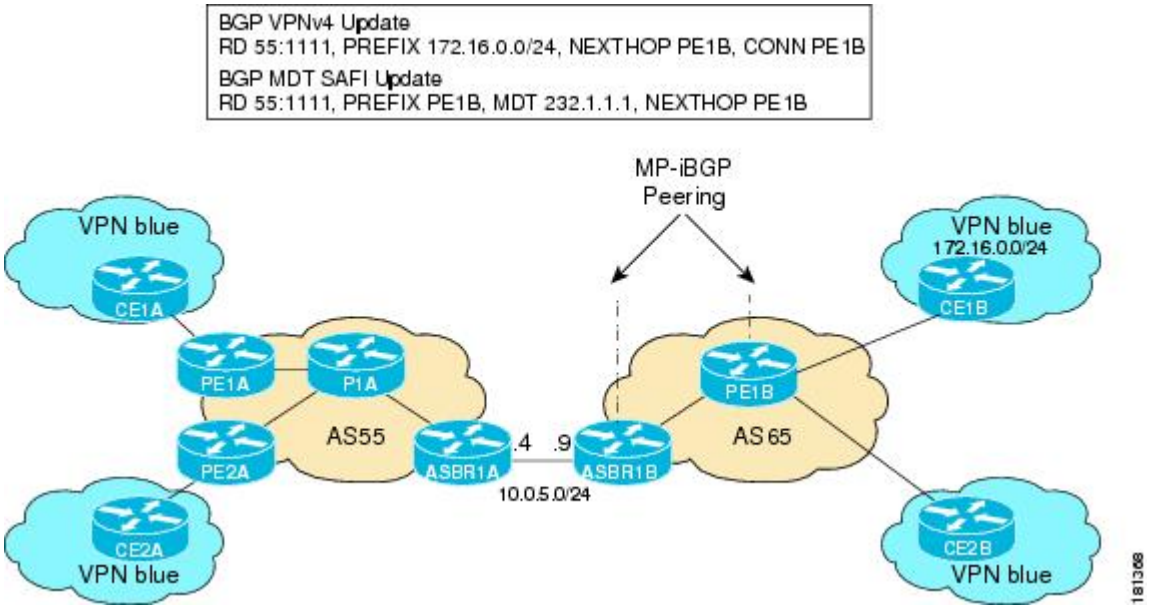
Figure 4: MVPN Inter-AS Support Option B Sample Topology



The following sequence of events occur to establish an MDT default tree rooted at PE1B in this inter-AS MVPN Option B topology:

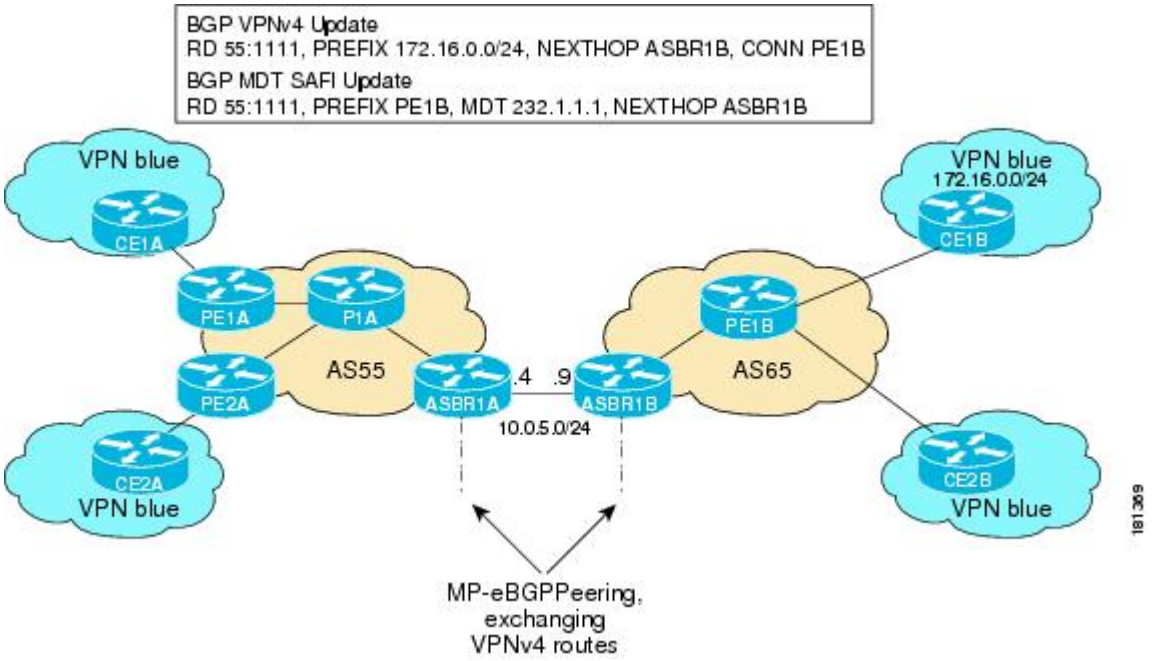
1. As illustrated in the following figure, PE1B advertises the default MDT information for VPN blue using the BGP MDT SAFI with itself (PE1B) as the next hop.

Figure 5: BGP Updates from PE1B to ASBR1B



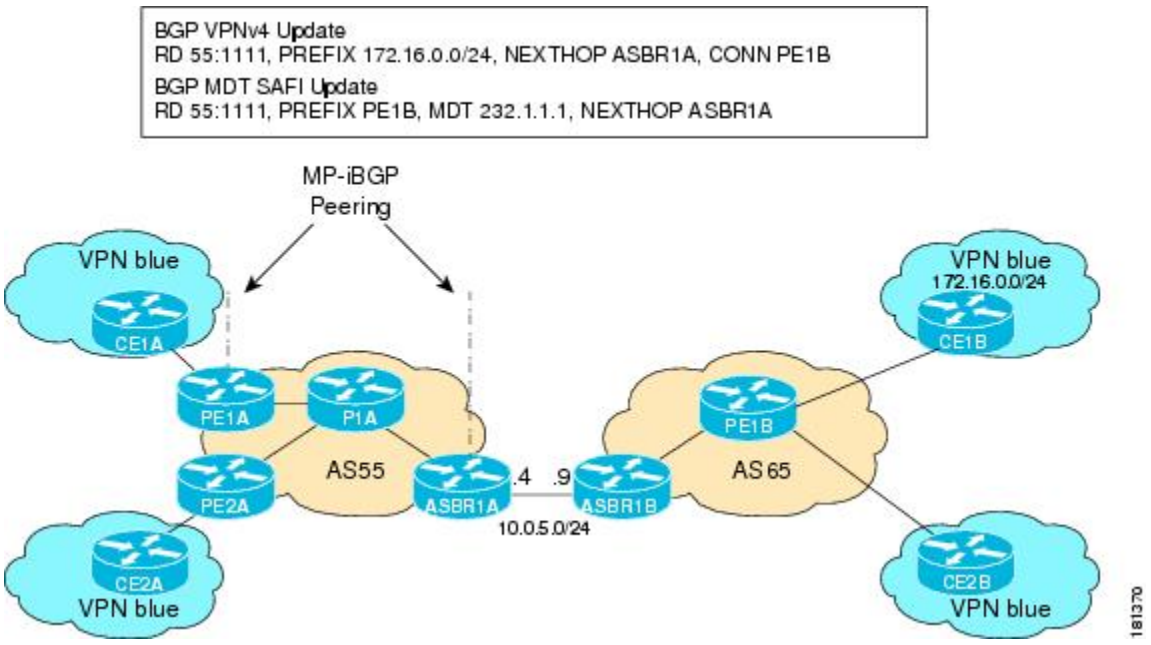
1. As illustrated in the following figure, ASBR1B receives the MDT SAFI information and, in turn, advertises this information to ASBR1A with itself (ASBR1B) as the next hop.

Figure 6: BGP Updates from ASBR1B to ASBR1A



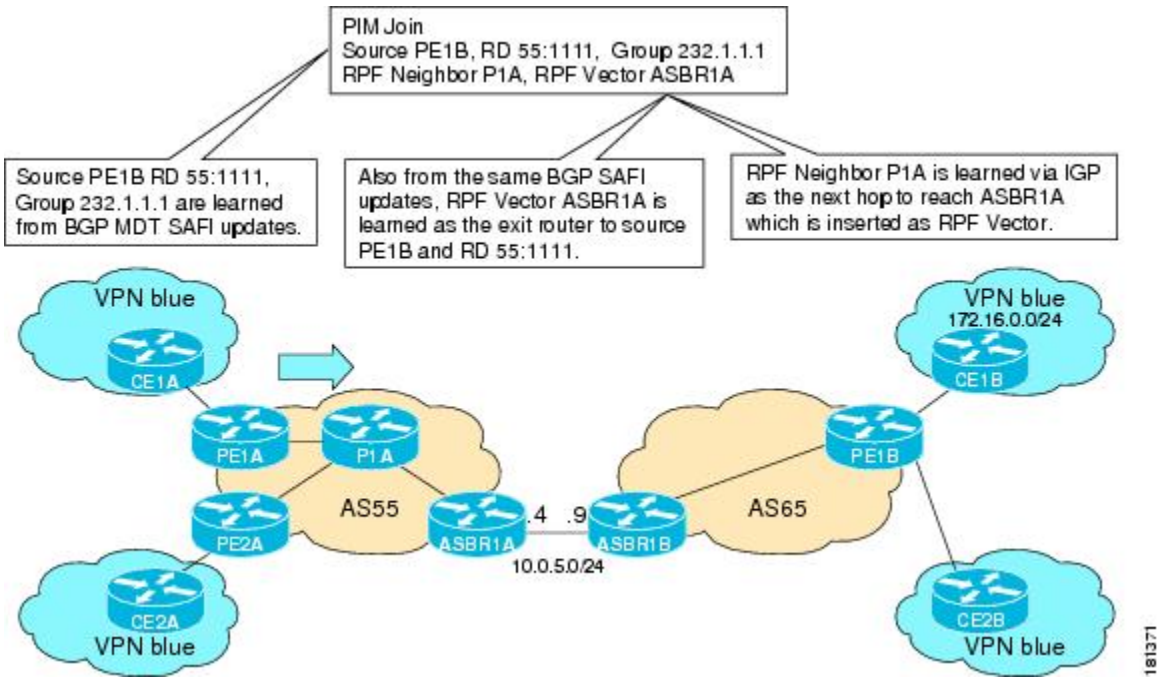
1. As illustrated in the following figure, ASBR1A advertises the MDT SAFI to PE1A with itself (ASBR1A) as the next hop.

Figure 7: BGP Updates from ASBR1A to PE1A



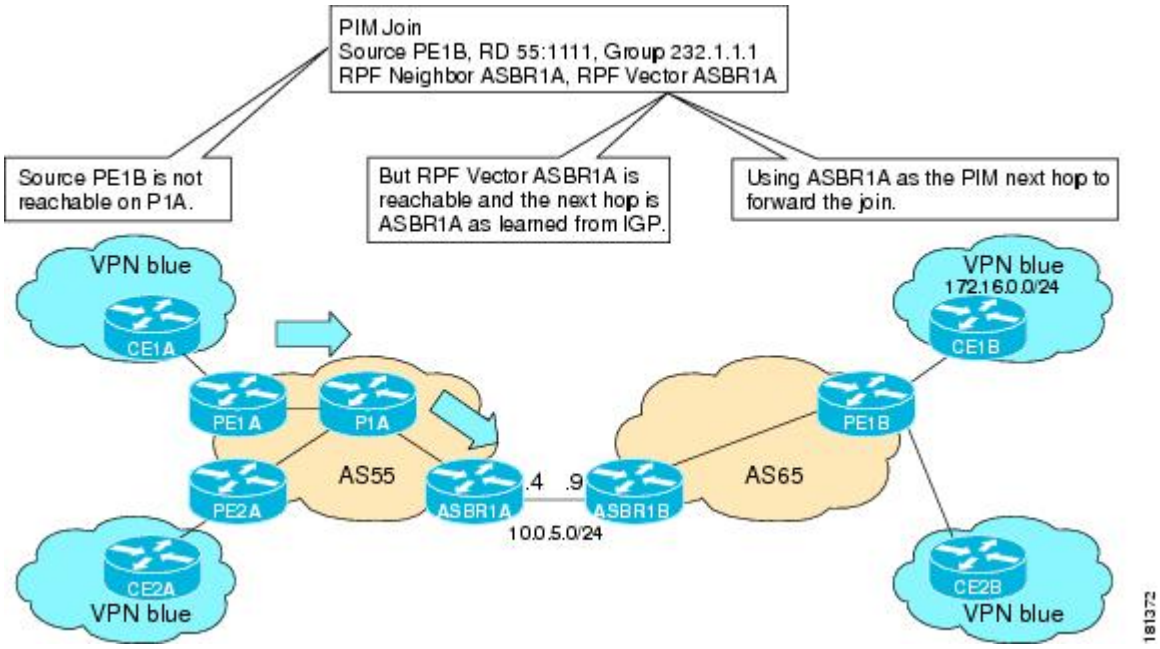
1. As illustrated in the following figure, PE1A learns the source PE router, the RD, and the default MDT group address from BGP MDT SAFI updates. In addition, from the same BGP MDT SAFI updates, PE1A learns that the RPF Vector, ASBR1A, is the exit router to source PE1B RD 55:1111. PE1A learns that P1A is an RPF neighbor through an IGP. PE1A then inserts the RPF Vector into the PIM join and sends the PIM join that is destined for source PE1B to P1A.

Figure 8: SSM Default PIM Join from PE1A to P1A



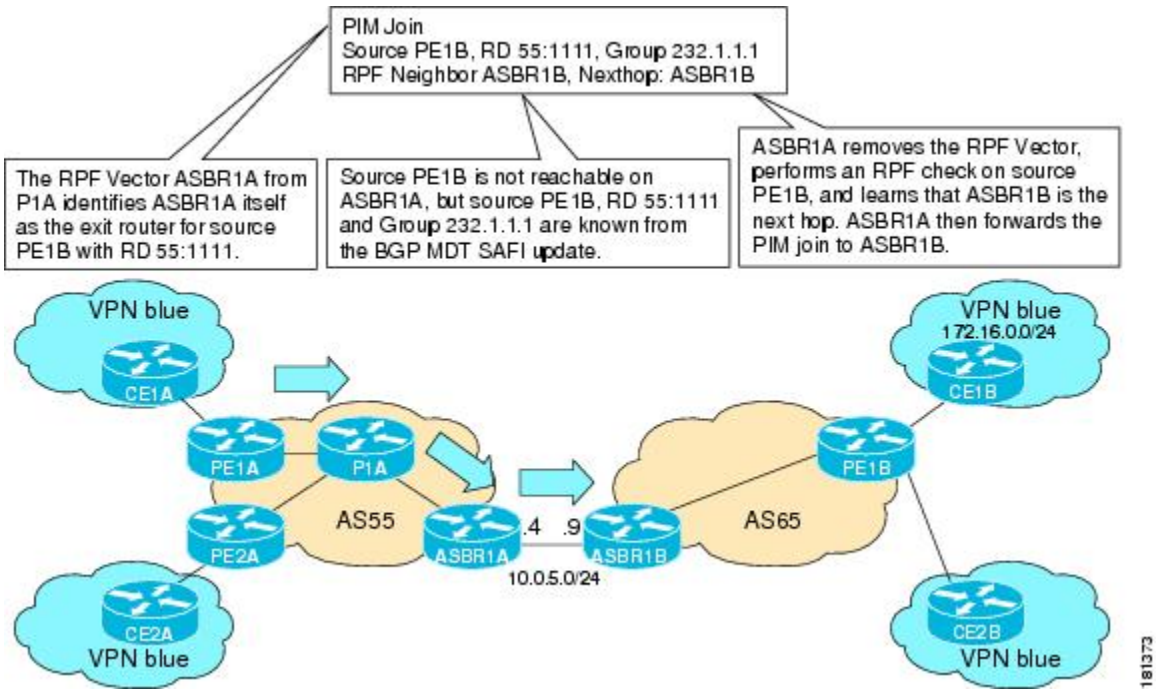
1. As illustrated in the following figure, source PE1B is not reachable on P1A, but the RPF Vector ASBR1A is reachable, and the next hop is ASBR1A, as learned from the IGP running in the core. P1A then forwards the PIM join to ASBR1A.

Figure 9: SSM Default MDT PIM Join from P1A to ASBR1A



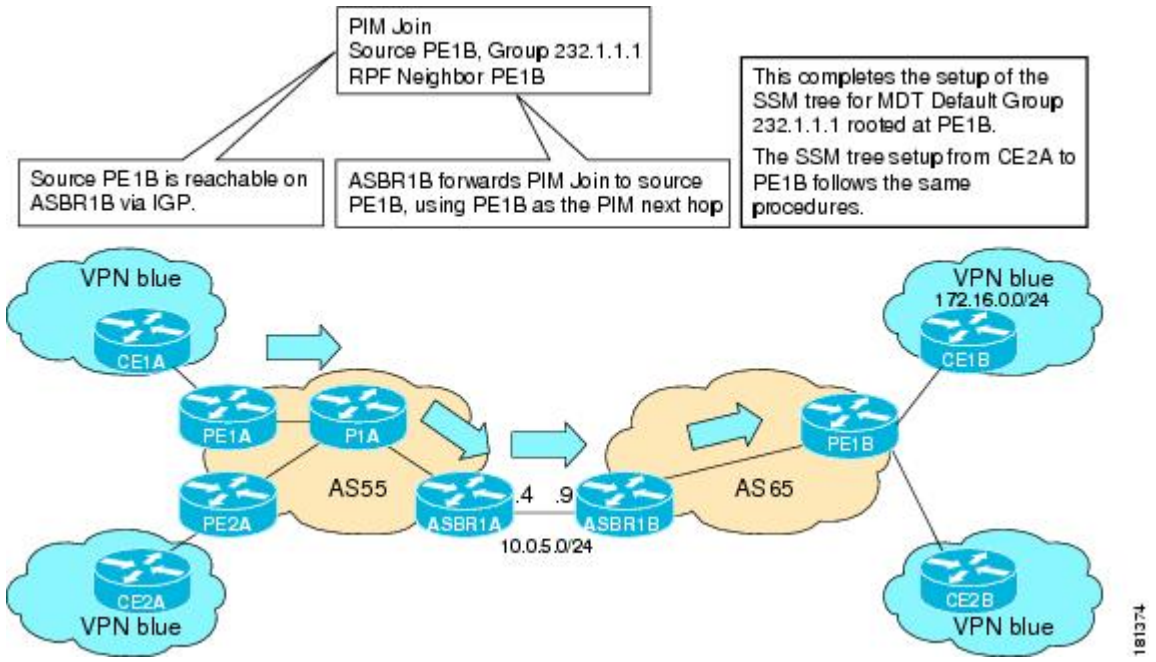
1. As illustrated in the following figure, the RPF Vector, ASBR1A, is contained in the PIM join sent from P1A to ASBR1A. When ASBR1A receives the RPF Vector, it learns that it is the exit router for source PE1B with RD 55:1111. Source PE1B is not reachable on ASBR1A, but source PE1B, RD 55:1111, and group 232.1.1.1 are known from the BGP MDT SAFI updates. The RPF neighbor P1A is learned from the IGP running in the core as the next hop to reach ASBR1A, which is inserted as the RPF Vector. ASBR1A then forwards the PIM join for source PE1B to ASBR1B.

Figure 10: SSM Default MDT PIM Join from ASBR1A to ASBR1B



1. As illustrated in the following figure, source PE1B is reachable on ASBR1B through the IGP running in AS65. ASBR1B forwards the PIM join to source PE1B, using PE1B as the next hop. At this point, the setup of the SSM tree for MDT default group 232.1.1.1 rooted at PE1B is complete. The SSM MDT default group rooted at PE1B, thus, has been established. The SSM trees for the MDT default groups rooted at PE1A and PE2A follow the same procedures.

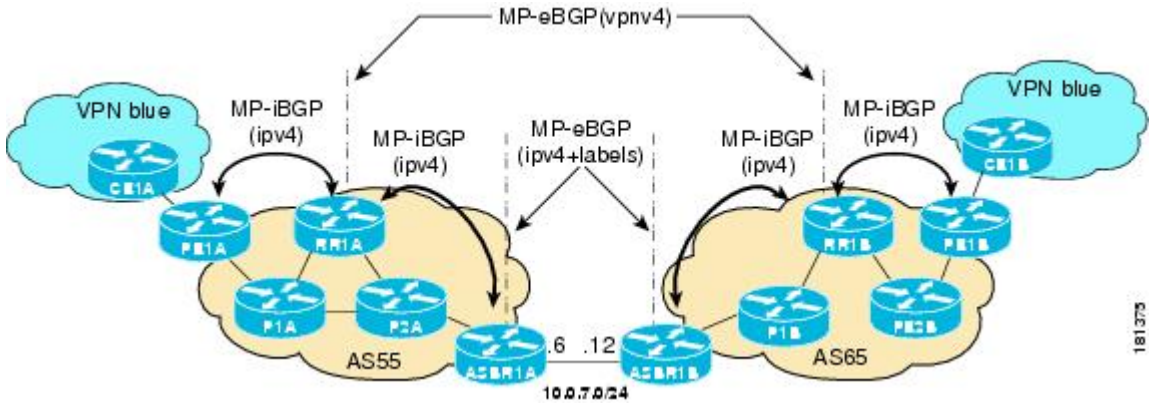
Figure 11: SSM Default MDT PIM Join from ASBR1B to PE1B



MVPN Inter-AS MDT Establishment for Option C

This section describes the sequence of events that leads to the establishment of an inter-AS MDT between the autonomous systems in the sample inter-AS Option C topology illustrated in the following figure. For this topology, assume that all the routers have been configured properly to support all features associated with the Multicast VPN Inter-AS Support feature.

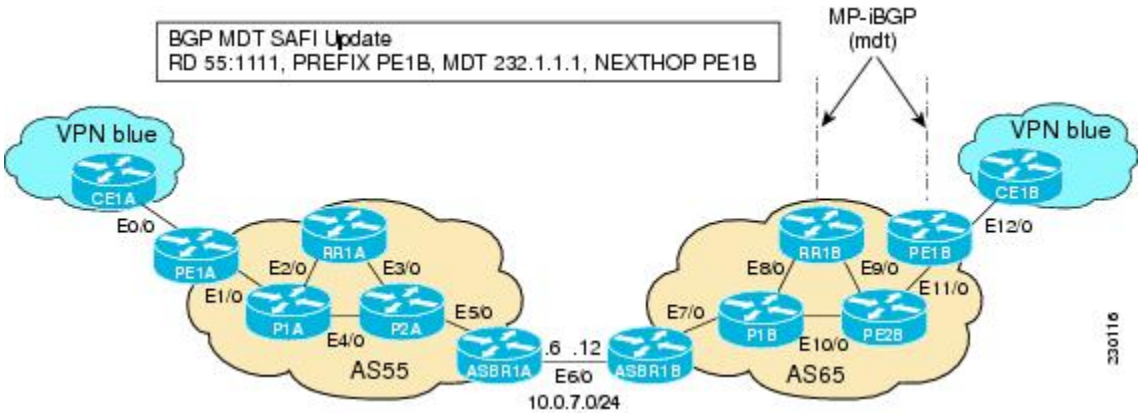
Figure 12: MVPN Inter-AS Support Option C Sample Topology



The following sequence of events occur to establish an MDT default tree rooted at PE1B in this inter-AS MVPN Option C topology:

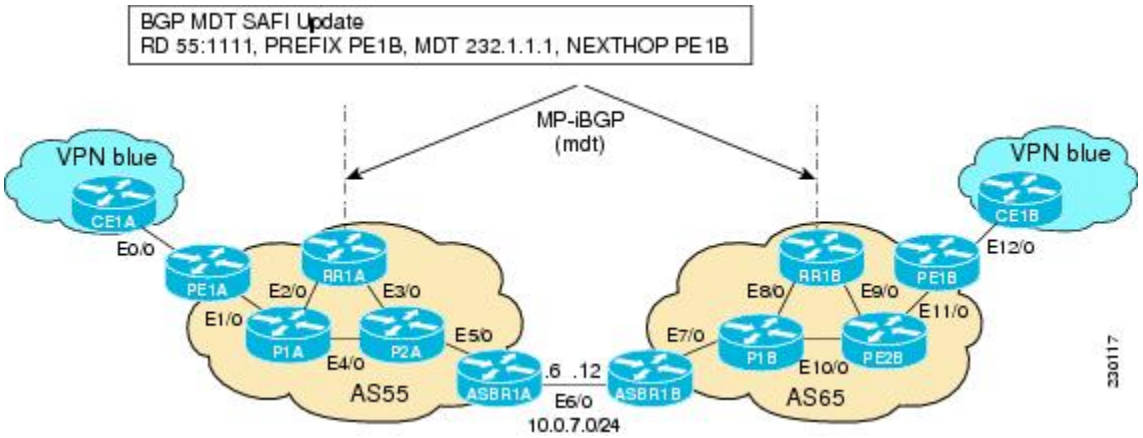
1. As illustrated in the following figure, PE1B advertises the default MDT information for VPN blue to RR1B within the BGP MDT SAFI.

Figure 13: BGP MDT SAFI Update from PE1B to RR1B



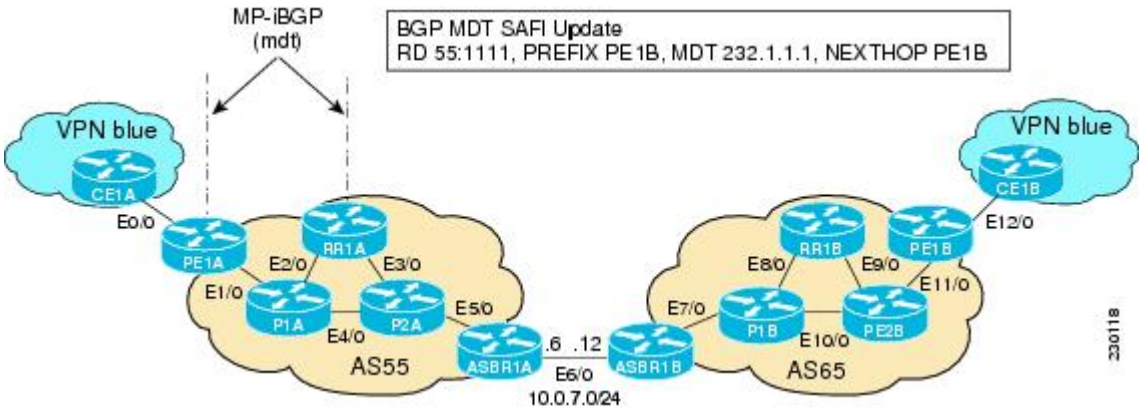
1. As illustrated in the following figure, RR1B receives the MDT SAFI information, and, in turn, advertises this information to RR1A.

Figure 14: BGP MDT SAFI Update from RR1B to RR1A



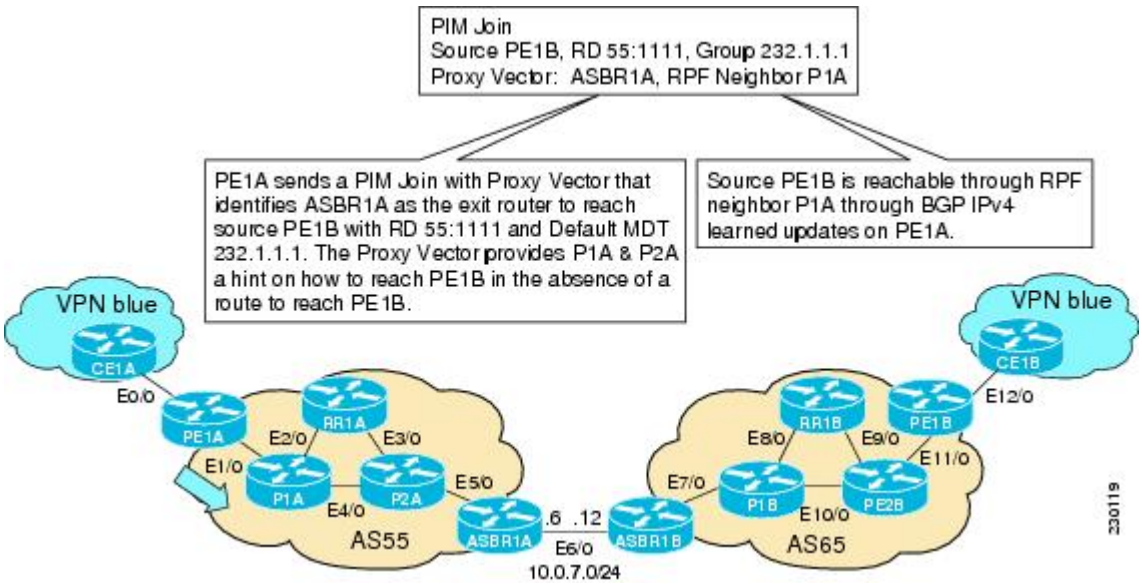
1. As illustrated in the following figure, RR1A receives the MDT SAFI information, and, in turn, advertises this information to PE1A.

Figure 15: BGP MDT SAFI Update from RR1A to PE1A



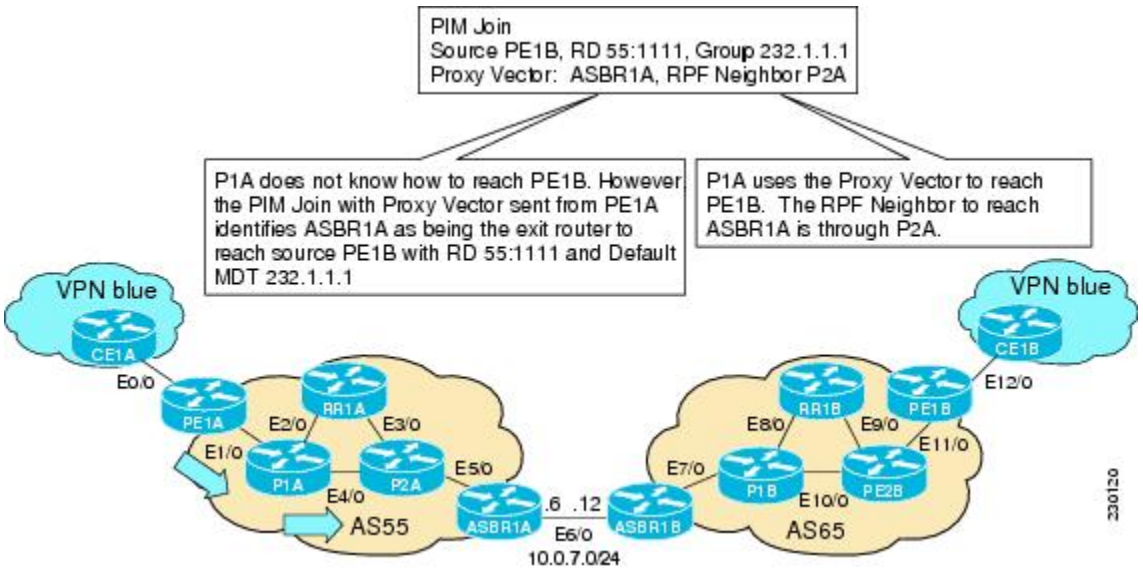
1. As illustrated in the following figure, PE1A sends a PIM join with the Proxy Vector that identifies ASBR1A as the exit router to reach source PE1B with RD 55:1111 and Default MDT 232.1.1.1. The Proxy Vector provides P1A and P2A a hint on how to reach PE1B in the absence of a route to reach PE1B. Source PE1B is reachable through RPF neighbor P1A through BGP IPv4 learned updates on PE1A.

Figure 16: PIM SSM Join for Default MDT with Proxy Vector from PE1A to P1A



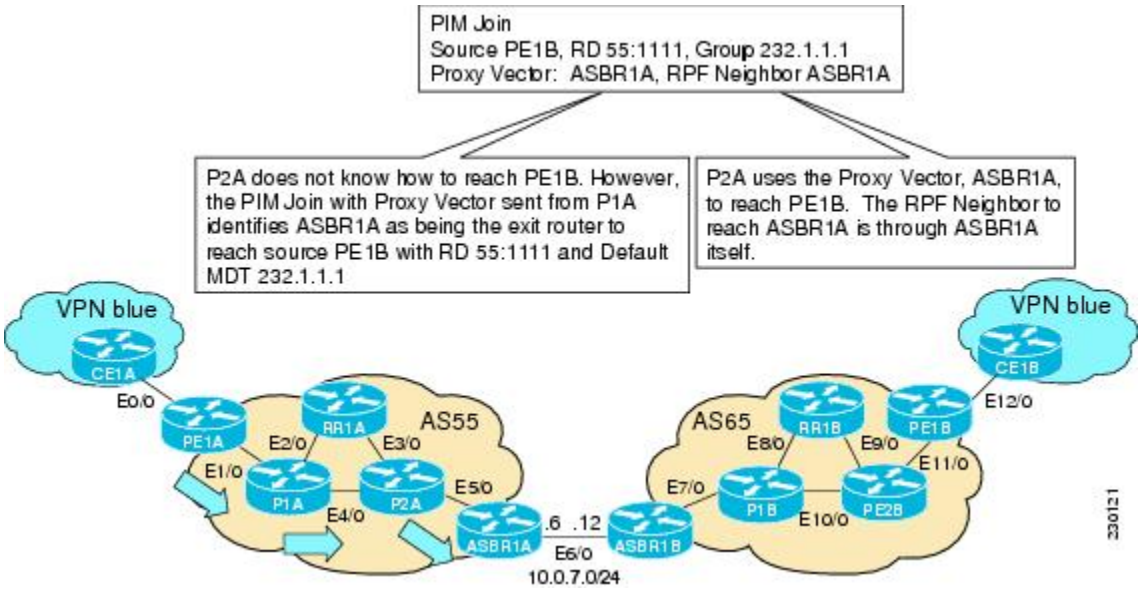
1. As illustrated in the following figure, P1A does not know how to reach PE1B. However, the PIM join with the Proxy Vector sent from PE1A identifies ASBR1A as being the exit router to reach source PE1B with RD 55:1111 and Default MDT 232.1.1.1. P1A uses the Proxy Vector to reach PE1B. The RPF neighbor to reach ASBR1A is through P2A. P1A, thus, forwards the PIM SSM join to P2A.

Figure 17: PIM SSM Join for Default MDT with Proxy Vector from P1A to P2A



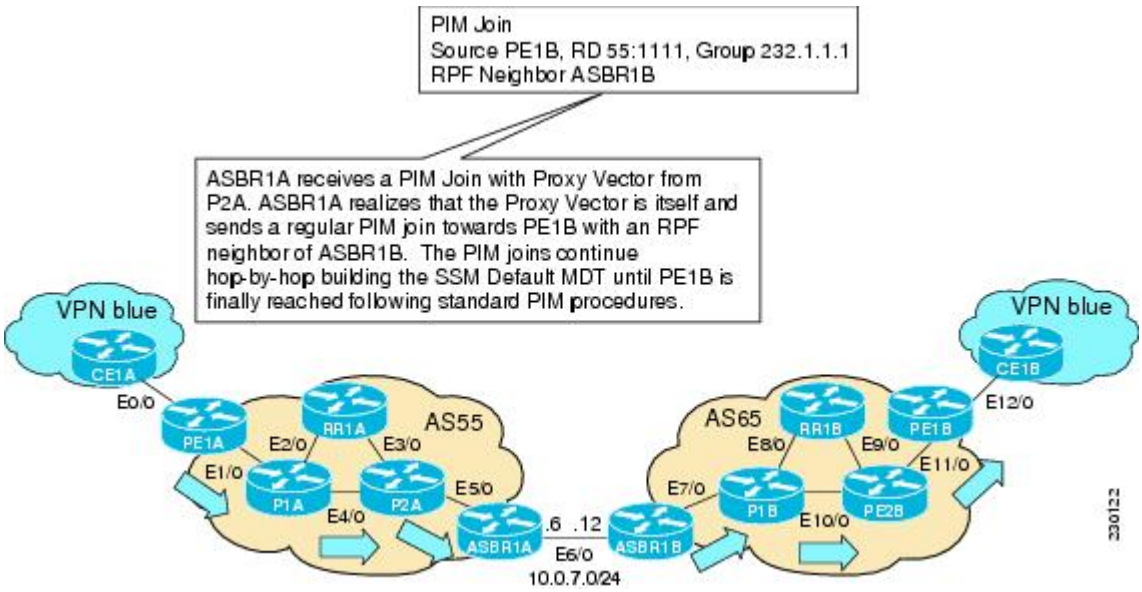
1. As illustrated in the following figure, P2A does not know how to reach PE1B. However, the PIM join with the Proxy Vector sent from P1A identifies ASBR1A as being the exit router to reach source PE1B with RD 55:1111 and Default MDT 232.1.1.1. P2A uses the Proxy Vector, ASBR1A, to reach PE1B. The RPF neighbor to reach ASBR1B is through ASBR1A (that is, itself).

Figure 18: PIM SSM Join for Default MDT with Proxy Vector from P2A to ASBR1A



1. As illustrated in the following figure, ASBR1A receives a PIM join with Proxy Vector from P2A. ASBR1A realizes that the Proxy Vector is itself and sends a regular PIM join towards PE1B with an RPF neighbor of ASBR1B. The PIM joins continue hop-by-hop building the SSM Default MDT until PE1B is finally reached following standard PIM procedures.

Figure 19: PIM SSM Join for Default MDT with Proxy Vector from ASBR1A to PE1B



How to Configure Multicast VPN Inter-AS Support

Configuring the MDT Address Family in BGP for Multicast VPN Inter-AS Support

Perform this task to configure an MDT address family session on PE routers to to establish MDT peering sessions for MVPN. The **mdt** keyword has been added to the **address-family ipv4** command to configure an MDT address-family session.

Supported Policy

The following policy configuration parameters are supported under the BGP MDT SAFI:

- Mandatory attributes and well-known attributes, such as the AS-path, multiexit discriminator (MED), BGP local preference, and next hop attributes.
- Standard communities, community lists, and route maps.

Guidelines for Configuring MDT Address Family Sessions on PE Routers for MVPN Inter-AS Support

When configuring routers for MVPN inter-AS support, follow these guidelines:

- For MVPN inter-AS Option A, BGP MDT address-family peering sessions are not required between the PE routers because native IP forwarding is used by the PE routers. For option A, BGP MDT peering sessions are only required for intra-AS VPN peering sessions.
- For MVPN inter-AS Option B, BGP MDT address-family peering sessions are only required between the PEs and ASBRs. In the Option B inter-AS case where PE routers use iBGP to redistribute labeled VPNv4 routes to RRs of which ASBRs are clients, then BGP MDT address-family peering sessions are required between the PEs, ASBRs, and RRs.

- For MVPN inter-AS Option C, BGP MDT address-family peering sessions are only required between the PEs and RRs.

Before you begin

Before inter-AS VPN peering can be established through an MDT address family, MPLS and Cisco Express Forwarding (CEF) must be configured in the BGP network and multiprotocol BGP on PE routers that provide VPN services to CE routers.



Note The following policy configuration parameters are not supported:

- Route-originator attribute
- NLRI prefix filtering (prefix lists, distribute lists)
- Extended community attributes (route target and site of origin)

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4 mdt**
5. **neighbor** *neighbor-address* **activate**
6. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65535`	Enters router configuration mode and creates a BGP routing process.

	Command or Action	Purpose
Step 4	**address-family ipv4 mdt** **Example:** `Router(config-router)# address-family ipv4 mdt`	Enters address family configuration mode to create an IP MDT address family session.
Step 5	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Displaying Information About IPv4 MDT Sessions in BGP

Perform this optional task to display information about IPv4 MDT sessions in BGP.

SUMMARY STEPS

1. **enable**
2. **show ip bgp ipv4 mdt** {**all** | **rd** | **vrf** *vrf-name*}

DETAILED STEPS

Procedure

Step 1 **enable**

Use this command to enable privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show ip bgp ipv4 mdt** {**all** | **rd** | **vrf** *vrf-name*}

Use this command to display IPv4 MDT sessions in the IPv4 BGP routing table.

The following is sample output from the **show ip bgp ipv4 mdt** command with the **all** keyword:

Example:

```
Router# show ip bgp ipv4 mdt all
```

```
BGP table version is 2, local router ID is 10.1.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 55:1111 (default for vrf blue)
*> 10.5.5.5/32      10.1.0.1                 55             0 23 24 25 54 ?
*  10.9.9.9/32      0.0.0.0                                  0 ?
```

Clearing IPv4 MDT Peering Sessions in BGP

Perform this optional task to reset IPv4 MDT address-family sessions using the **mdt** keyword in one of the various forms of the **clear ip bgp**command. Due to the complexity of some of the keywords available for the **clear ip bgp** command, some of the keywords are documented as separate commands.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **clear ip bgp ipv4 mdt** *as-number* [**in**[**prefix-filter**]] [**out**] [**soft** [**in**[**prefix-filter**] | **out**]]
   -
   - **clear ip bgp ipv4 mdt peer-group** *peer-group-name* [**in** *[* **prefix-filter**]] [**out**] [**soft** *[* **in**[**prefix-filter**] | **out**]]
   -
   - **clear ip bgp ipv4 mdt update-group** [*index-group* | *neighbor-address*]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Do one of the following: • **clear ip bgp ipv4 mdt** *as-number* [**in**[**prefix-filter**]] [**out**] [**soft** [**in**[**prefix-filter**]	**out**]] • • **clear ip bgp ipv4 mdt peer-group** *peer-group-name* [**in** *[* **prefix-filter**]] [**out**] [**soft** *[* **in**[**prefix-filter**]	**out**]] • • **clear ip bgp ipv4 mdt update-group** [*index-group*	*neighbor-address*]	(Optional) Resets IPv4 MDT address-family sessions. • Specifying the **clear ip bgp ipv4 mdt** command with the *as-number*argument resets IPv4 MDT address-family sessions associated with the specified autonomous system. • The example for this form of the **clear ip bgp ipv4 mdt** command shows how to initiate a hard reset for all IPv4 MDT address-family sessions in the autonomous system numbered 65700.

	Command or Action	Purpose

	Example:
	`Router# clear ip bgp ipv4 mdt 65700`
	Example:
	Example:
	`Router# clear ip bgp ipv4 mdt peer-group test soft in`
	Example:
	Example:
	`Router# clear ip bgp ipv4 mdt update-group 3`
		or
		• Specifying the **clear ip bgp ipv4 mdt** command with the **peer-group** keyword resets IPv4 MDT address-family sessions for all members of a BGP peer group.
		• The example for this form of the **clear ip bgp ipv4 mdt** command shows how to initiate a soft reset for inbound MDT address-family sessions with members of the peer group test. Outbound sessions are unaffected.
		or
		• Specifying the **clear ip bgp ipv4 mdt** command with the **update-group** keyword resets IPv4 MDT address-family sessions for all the members of a BGP update group.
		• The example for this form of the **clear ip bgp ipv4 mdt** command shows how to reset for all members of the update group 3.

Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support - Option B

Perform this task to configure PE routers in an Option B deployment to support the extensions necessary (BGP connector attribute, BGP MDT SAFI, and RPF Vector) to send BGP MDT updates to build the default MDT for MVPN inter-AS support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *ip-address* **remote-as** *as-number*
5. **address-family ipv4 mdt**
6. **neighbor** *neighbor-address* **activate**
7. **neighbor** *neighbor-address* **next-hop-self**
8. **exit**
9. **address-family vpnv4**
10. **neighbor** *neighbor-address* **activate**
11. **neighbor** *neighbor-address* **send-community extended**
12. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**address-family ipv4 mdt** **Example:** `Router(config-router)# address-family ipv4 mdt`	Enters address family configuration mode to create an IPv4 MDT address family session.
Step 6	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 7	**neighbor** *neighbor-address* **next-hop-self** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 next-hop-self`	Disables next hop processing of BGP updates on the router.
Step 8	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 9	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.
Step 10	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the VPNv4 address family for this neighbor.
Step 11	**neighbor** *neighbor-address* **send-community extended** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables the standard and extended community attributes to be sent to this neighbor.
Step 12	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support - Option C

Perform this task to configure PE routers in an Option B deployment to support the extensions necessary (BGP connector attribute, BGP MDT SAFI, and RPF Vector) to send BGP MDT updates to build the default MDT for MVPN inter-AS support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *ip-address* **remote-as** *as-number*
5. **address-family ipv4 mdt**
6. **neighbor** *neighbor-address* **activate**
7. **neighbor** *neighbor-address* **send-community extended**
8. **exit**
9. **address-family vpnv4**
10. **neighbor** *neighbor-address* **activate**
11. **neighbor** *neighbor-address* **send-community extended**
12. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**address-family ipv4 mdt** **Example:** `Router(config-router)# address-family ipv4 mdt`	Enters address family configuration mode to create an IPv4 MDT address family session.
Step 6	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 7	**neighbor** *neighbor-address* **send-community extended** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables the standard and extended community attributes to be sent to this neighbor.
Step 8	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 9	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.
Step 10	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the VPNv4 address family for this neighbor.
Step 11	**neighbor** *neighbor-address* **send-community extended** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables the standard and extended community attributes to be sent to this neighbor.
Step 12	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Verifying the Establishment of Inter-AS MDTs in Option B and Option C Deployments

Perform this optional task to verify the establishment of Inter-AS MDTs in Option B and Option C MVPN inter-AS deployments.

Note The steps in this optional task can be performed in any order. All steps in this task are optional.

SUMMARY STEPS

1. **enable**
2. **show ip mroute proxy**
3. **show ip pim** [**vrf** *vrf-name*] **neighbor** [*interface-type interface-number*]
4. **show ip rpf** [**vrf** *vrf-name*] {*route-distinguisher* | *source-address* [*group-address*] [**rd** *route-distinguisher*]} [**metric**]
5. **show ip pim** [**vrf** *vrf-name*] **mdt bgp**

DETAILED STEPS

Procedure

Step 1 **enable**

Use this command to enable privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show ip mroute proxy**

Use this command to display information about RPF Vectors received on a multicast router.

- The information displayed in the output of this command can be used to determine if an RPF Vector proxy is received on a core router.

The following is sample output from the **show ip mroute proxy** command:

Example:

```
Router# show ip mroute proxy

(192.168.0.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.4       10.0.3.1         PIM       00:03:29/00:02:06
  55:1111/192.168.0.4       10.0.3.2         PIM       00:17:47/00:02:06
```

Step 3 **show ip pim** [**vrf** *vrf-name*] **neighbor** [*interface-type interface-number*]

Use this command to display the PIM neighbors discovered by PIMv1 router query messages or PIMv2 hello messages.

The P flag indicates that the neighbor has announced (through PIM hello messages) its capability to handle RPF Vectors in PIM join messages. All software versions that support the PIM RPF Vector feature announce this PIM hello option. An RPF Vector is only included in PIM messages when all PIM neighbors on an RPF interface support it.

The following is sample output from the **show ip pim neighbor** command:

Example:

```
Router# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface               Uptime/Expires    Ver   DR
Address                                                           Prio/Mode
10.0.0.1          GigabitEthernet10/2     00:01:29/00:01:15 v2    1 / S
10.0.0.3          GigabitEthernet10/3     00:01:15/00:01:28 v2    1 / DR S P
```

Step 4 **show ip rpf** [**vrf** *vrf-name*] {*route-distinguisher* | *source-address* [*group-address*] [**rd** *route-distinguisher*]} [**metric**]

Use this command to display information about how IP multicast routing does RPF.

The following is sample output from the **show ip rpf**command:

Example:

```
Router# show ip rpf 10.7.0.7 232.1.1.1 rd 55:1111

RPF information for ? (10.7.0.7)
  RPF interface: GigabitEthernet2/2
  RPF neighbor: ? (10.0.1.3)
  RPF route/mask: 10.5.0.5/32
  RPF type: unicast (UNKNOWN)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
  BGP lookup of 55:1111/10.7.0.7 next_hop: 10.5.0.5
  PROXY vector: 10.5.0.5
```

Step 5 **show ip pim** [**vrf** *vrf-name*] **mdt bgp**

Use this command to display information about the BGP advertisement of RDs for the MDT default group.

The following is sample output from the **show ip pim mdt bgp** command:

Example:

```
Router# show ip pim mdt bgp
MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:192.168.0.2                        192.168.0.2       192.168.0.2
   55:1111:192.168.0.8                        192.168.0.4       192.168.0.4
```

Configuration Examples for Multicast VPN Inter-AS Support

Configuring an IPv4 MDT Address-Family Session for Multicast VPN Inter-AS Support Example

The following examples shows how to configure a router to support an IPv4 MDT address-family session with the BGP neighbor at 10.1.1.2:

```
router bgp 1
 address-family ipv4 mdt
 neighbor 10.1.1.2 activate
```

Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support

The following example shows how to configure a PE router to support the extensions necessary (BGP connector attribute, BGP MDT SAFI, and RPF Vector) to send BGP MDT updates to build the default MDT for MVPN inter-AS support in an Option B deployment. Only the relevant configuration is shown in this example.

```
!
ip multicast-routing
ip multicast-routing vrf blue
```

```
ip multicast vrf blue rpf proxy rd vector
!
.
.
.
router bgp 55
.
.
.
 !
 address-family ipv4 mdt
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 next-hop-self
  neighbor 192.168.0.4 activate
  neighbor 192.168.0.4 next-hop-self
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 send-community extended
  neighbor 192.168.0.4 activate
  neighbor 192.168.0.4 send-community extended
 exit-address-family
 !
.
.
.
!
ip pim ssm default
!
```

Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support

The following example shows how to configure a PE router to support the extensions necessary (BGP connector attribute, BGP MDT SAFI, and RPF Vector) to send BGP MDT updates to build the default MDT for MVPN inter-AS support in an Option B deployment. Only the relevant configuration is shown in this example.

```
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast rpf proxy vector
!
.
.
.
!
router bgp 65
.
.
.
 !
 address-family ipv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 mdt
```

```
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 exit-address-family
 !
.
.
.
!
ip pim ssm default
!
```

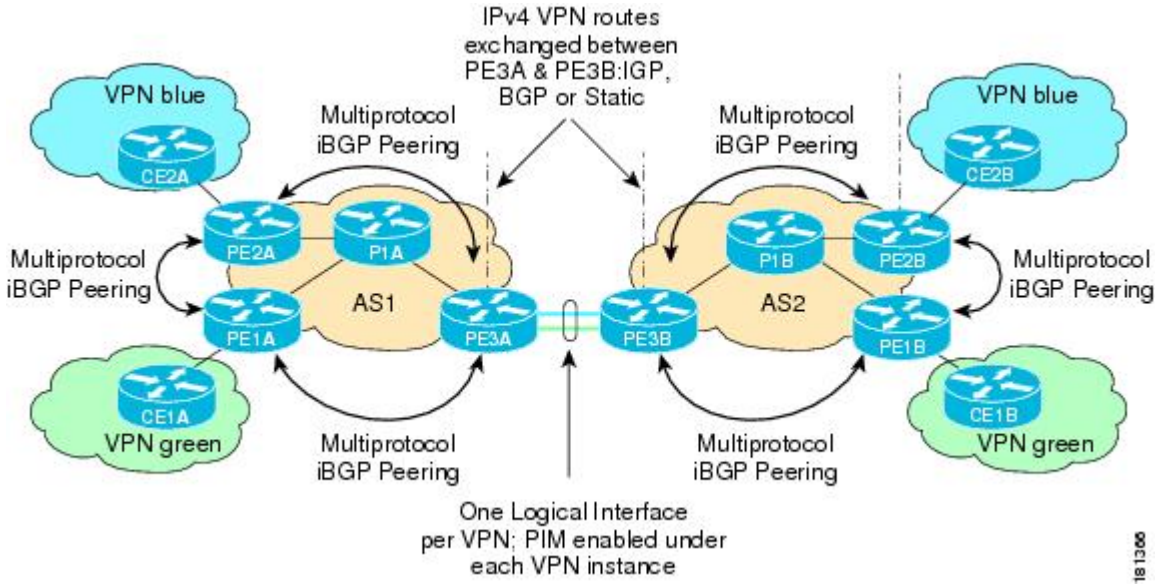
Configuring Back-to-Back ASBR PEs - Option A Example

The following example shows how to configure support for MVPN inter-AS support Option A. This configuration example is based on the sample inter-AS network Option A topology illustrated in the figure.

In this configuration example, PE3A in AS1 is attached directly to PE3B in AS2. The two PE routers are attached by physical interfaces, one physical interface for each of the VPNs (VPN blue and VPN green) whose routes need to be passed from AS1 to AS2, and vice versa. Each PE will treat the other as if it were a CE router; that is, the PEs associate each interface with a VRF and use eBGP to distribute unlabeled IPv4 addresses to each other. Intermediate System-to-Intermediate System (IS-IS) is being used for the BGP peerings in both autonomous systems, and Routing Information Protocol (RIP) is being used on the PE routers that face the CE routers to dynamically learn the routes from the VRFs and advertise them as VPNv4 routes to the remote PE routers. RIP is also being used between the ASBRs to set up the eBGP peerings between PE3A and PE3B.

Note For Option A, any IGP can be used to exchange the IPv4 routes for the loopback interfaces.

Figure 20: Topology for MVPN Inter-AS Support Option A Configuration Example



The table provides information about the topology used for the Option A configuration example presented in this section.

Table 1: Topology Information for MVPN Inter-AS Option A Configuration Example

PE Router	VPN	RD	AS Number	Loopback0 Interface	Default MDT (PIM-SSM)
PE1A	green	55:1111	1	10.1.1.1/32	232.1.1.1
PE2A	blue	55:1111	1	10.1.1.2/32	232.1.1.1
PE3A	blue/green	55:1111	1	10.1.1.3/32	232.1.1.1
PE1B	green	55:2222	2	10.2.2.1/32	232.2.2.2
PE2B	blue	55:2222	2	10.2.2.2/32	232.2.2.2
PE3B	blue/green	55:2222	2	10.2.2.3/32	232.2.2.2

PE1A

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE1A
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf green
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.2.2.2
!
ip multicast-routing
ip multicast-routing vrf green
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface GigabitEthernet0/0
 ip vrf forwarding green
 ip address 172.25.11.1 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-mode
 tag-switching ip
```

```
!
interface GigabitEthernet1/0
 ip address 172.30.41.1 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.1111.00
!
router rip
 version 2
 !
 address-family ipv4 vrf green
 version 2
 network 172.25.0.0
 no auto-summary
 exit-address-family
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback0
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.3 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf green
 redistribute rip metric 1
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
ip pim vrf green send-rp-announce GigabitEthernet0/0 scope 32
ip pim vrf green send-rp-discovery 0/0 scope 32
ip pim vrf green register-rate-limit 2
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
```

```
no cns aaa enable
end
```

PE2A

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE2A
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.1.1.2 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface GigabitEthernet0/0
 ip vrf forwarding blue
 ip address 172.17.12.2 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-mode
 tag-switching ip
!
interface GigabitEthernet1/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface GigabitEthernet2/0
 ip address 172.19.142.2 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.2222.00
!
router rip
 version 2
 !
```

```
 address-family ipv4 vrf blue
 version 2
 network 172.17.0.0
 no auto-summary
 exit-address-family
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback0
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.3 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute rip metric 1
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
ip pim vrf blue send-rp-announce GigabitEthernet0/0 scope 32
ip pim vrf blue send-rp-discovery GigabitEthernet0/0 scope 32
ip pim vrf blue ssm default
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE3A

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE3A
!
boot-start-marker
```

```
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip vrf green
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.2.2.2
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast-routing vrf green
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.1.1.3 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface GigabitEthernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface GigabitEthernet1/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface GigabitEthernet2/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface GigabitEthernet3/0
 ip address 192.168.143.3 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
interface GigabitEthernet4/0
 ip vrf forwarding blue
 ip address 172.20.34.3 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-dense-mode
 tag-switching ip
!
interface GigabitEthernet5/0
 ip vrf forwarding green
 ip address 172.23.35.3 255.255.255.0
 no ip directed-broadcast
```

```
 ip pim sparse-dense-mode
 tag-switching ip
!
router eigrp 1
 !
 address-family ipv4 vrf blue
 network 172.20.0.0
 no auto-summary
 exit-address-family
!
router isis
 net 49.0000.0000.3333.00
!
router rip
 version 2
 !
 address-family ipv4 vrf green
 version 2
 redistribute bgp 1 metric 2
 network 172.23.0.0
 no auto-summary
 exit-address-family
 !
 address-family ipv4 vrf blue
 version 2
 redistribute bgp 1 metric 1
 network 172.20.0.0
 no auto-summary
 exit-address-family
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback0
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.2 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 bgp redistribute-internal
 exit-address-family
 !
 address-family ipv4 vrf green
 no synchronization
 bgp redistribute-internal
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute rip
 no synchronization
 bgp redistribute-internal
 exit-address-family
!
ip classless
```

```
!
ip pim ssm default
ip pim vrf blue ssm default
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE3B

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE3B
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip vrf green
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.2.2.2
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast-routing vrf green
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.2.2.3 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface GigabitEthernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
```

```
!
interface GigabitEthernet1/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface GigabitEthernet2/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface GigabitEthernet3/0
 ip address 172.16.43.3 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
interface GigabitEthernet4/0
 ip vrf forwarding blue
 ip address 172.20.34.4 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-dense-mode
 tag-switching ip
!
interface GigabitEthernet5/0
 ip vrf forwarding green
 ip address 172.23.35.4 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-dense-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.3333.00
!
router rip
 version 2
 !
 address-family ipv4 vrf green
 version 2
 network 172.23.0.0
 no auto-summary
 exit-address-family
 !
 address-family ipv4 vrf blue
 version 2
 network 172.20.0.0
 no auto-summary
 exit-address-family
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 redistribute rip metric 1
 neighbor 10.2.2.1 remote-as 2
 neighbor 10.2.2.1 update-source Loopback0
 neighbor 10.2.2.2 remote-as 2
 neighbor 10.2.2.2 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.2.2.1 activate
 neighbor 10.2.2.2 activate
 exit-address-family
```

```
 !
 address-family vpnv4
 neighbor 10.2.2.1 activate
 neighbor 10.2.2.1 send-community extended
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf green
 redistribute rip
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute rip
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
ip pim vrf blue ssm default
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE2B

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE2B
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
mpls label protocol ldp
!
```

```
!
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface GigabitEthernet0/0
 ip vrf forwarding blue
 ip address 172.18.22.2 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-mode
 tag-switching ip
!
interface GigabitEthernet1/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface GigabitEthernet2/0
 ip address 172.19.42.2 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.2222.00
!
router rip
 !
 address-family ipv4 vrf blue
 network 172.18.0.0
 no auto-summary
 exit-address-family
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.2.1 remote-as 2
 neighbor 10.2.2.1 update-source Loopback0
 neighbor 10.2.2.3 remote-as 2
 neighbor 10.2.2.3 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.2.2.1 activate
 neighbor 10.2.2.3 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.2.2.1 activate
 neighbor 10.2.2.1 send-community extended
 neighbor 10.2.2.3 activate
 neighbor 10.2.2.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 no synchronization
 exit-address-family
!
ip classless
!
```

```
ip pim ssm default
ip pim vrf blue ssm default
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE1B

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE1B
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf green
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.2.2.2
!
ip multicast-routing
ip multicast-routing vrf green
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.2.2.1 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface GigabitEthernet0/0
 ip vrf forwarding green
 ip address 172.25.111.1 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-mode
 tag-switching ip
!
interface GigabitEthernet1/0
 ip address 172.30.141.1 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
```

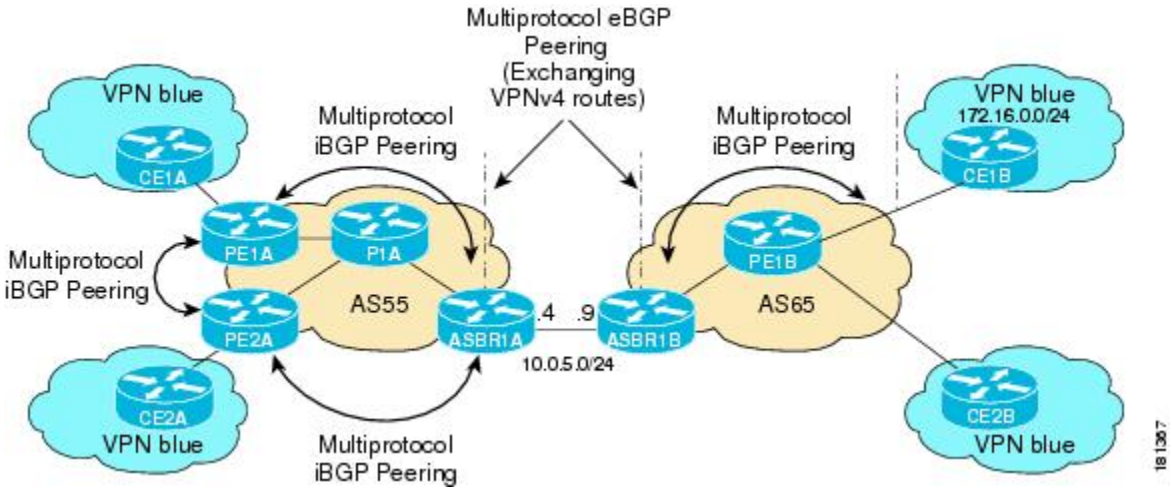
```
 tag-switching ip
!
router isis
 net 49.0000.0000.1111.00
!
router rip
 version 2
 !
 address-family ipv4 vrf green
 version 2
 network 172.25.0.0
 no auto-summary
 exit-address-family
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.2.2 remote-as 2
 neighbor 10.2.2.2 update-source Loopback0
 neighbor 10.2.2.3 remote-as 2
 neighbor 10.2.2.3 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.3 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community extended
 neighbor 10.2.2.3 activate
 neighbor 10.2.2.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf green
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

Configuring the Exchange of VPNv4 Routes Directly Between ASBRs - Option B Example

The following example shows how to configure MVPN inter-AS support in an Option B deployment. This configuration is based on the sample inter-AS topology illustrated in the figure.

In this configuration example, PE1A and PE2A are configured to use iBGP to redistribute labeled VPNv4 routes to each other and to ASBR1A, and PE1B is configured to redistribute labeled VPNv4 routes to ASBR1B. ASBR1A and ASBR1B are configured to use eBGP to exchange those labeled VPNv4 routes to each other.

Figure 21: Topology for MVPN Inter-AS Support Option B Configuration Example



The table provides information about the topology used for this particular Option B configuration example.

Table 2: Topology Information for MVPN Inter-AS Support Option B Configuration Example

PE or ASBR Router	AS Number	Loopback0 Interfaces	Default MDT (PIM-SSM)
PE1A	55	192.168.0.1/32	232.1.1.1
PE2A	55	192.168.0.2/32	232.1.1.1
ASBR1A	55	192.168.0.4/32	232.1.1.1
ASBR1B	65	192.168.0.9/32	232.1.1.1
PE1B	65	192.168.0.8/32	232.1.1.1

PE1A

```
!
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast vrf blue rpf proxy rd vector
!
.
.
.
!
interface GigabitEthernet0/0
 ip vrf forwarding blue
 ip pim sparse-mode
```

```
!
.
.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 192.168.0.2 remote-as 55
 neighbor 192.168.0.2 update-source Loopback0
 neighbor 192.168.0.4 remote-as 55
 neighbor 192.168.0.4 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 next-hop-self
  neighbor 192.168.0.4 activate
  neighbor 192.168.0.4 next-hop-self
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 send-community extended
  neighbor 192.168.0.4 activate
  neighbor 192.168.0.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
  redistribute connected
  redistribute static
  redistribute rip metric 50
  no auto-summary
  no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

PE2A

```
!
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast vrf blue rpf proxy rd vector
!
.
.
.
!
interface GigabitEthernet0/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
```

```
.
.
!
router bgp 55
 neighbor 192.168.0.1 remote-as 55
 neighbor 192.168.0.1 update-source Loopback0
 neighbor 192.168.0.4 remote-as 55
 neighbor 192.168.0.4 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 192.168.0.1 activate
 neighbor 192.168.0.1 next-hop-self
 neighbor 192.168.0.4 activate
 neighbor 192.168.0.4 next-hop-self
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.0.1 activate
 neighbor 192.168.0.1 send-community extended
 neighbor 192.168.0.4 activate
 neighbor 192.168.0.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute connected
 redistribute static
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

ASBR1A

```
!
ip multicast-routing
ip multicast-routing vrf blue
!
.
.
.
!
!
interface GigabitEthernet0/0
 ip pim sparse-mode
!
.
.
.
!
router bgp 55
 bgp log-neighbor-changes
 neighbor 10.0.5.9 remote-as 65
 neighbor 192.168.0.1 remote-as 55
 neighbor 192.168.0.1 update-source Loopback0
 neighbor 192.168.0.2 remote-as 55
 neighbor 192.168.0.2 update-source Loopback0
 !
 address-family ipv4 mdt
```

```
 neighbor 10.0.5.9 activate
 neighbor 192.168.0.1 activate
 neighbor 192.168.0.1 next-hop-self
 neighbor 192.168.0.2 activate
 neighbor 192.168.0.2 next-hop-self
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.0.5.9 activate
 neighbor 10.0.5.9 send-community extended
 neighbor 192.168.0.1 activate
 neighbor 192.168.0.1 send-community extended
 neighbor 192.168.0.1 next-hop-self
 neighbor 192.168.0.2 activate
 neighbor 192.168.0.2 send-community extended
 neighbor 192.168.0.2 next-hop-self
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

ASBR1B

```
!
ip multicast-routing
ip multicast-routing vrf blue
!
.
.
.
!
interface GigabitEthernet0/0
 ip pim sparse-mode
!
.
.
.
!
router bgp 65
 bgp log-neighbor-changes
 neighbor 10.0.5.4 remote-as 55
 neighbor 192.168.0.8 remote-as 65
 neighbor 192.168.0.8 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 10.0.5.4 activate
 neighbor 192.168.0.8 activate
 neighbor 192.168.0.8 next-hop-self
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.0.5.4 activate
 neighbor 10.0.5.4 send-community extended
 neighbor 192.168.0.8 activate
 neighbor 192.168.0.8 send-community extended
 neighbor 192.168.0.8 next-hop-self
 exit-address-family
!
.
```

```
.
.
!
ip pim ssm default
!
```

PE1B

```
!
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast vrf blue rpf proxy rd vector
!
.
.
.
!
interface GigabitEthernet0/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 65
 neighbor 192.168.0.9 remote-as 65
 neighbor 192.168.0.9 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 192.168.0.9 activate
 neighbor 192.168.0.9 next-hop-self
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.0.9 activate
 neighbor 192.168.0.9 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute connected
 redistribute static
 redistribute rip metric 50
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

The following is sample output from the **show ip pim mdt bgp** command for PE1A, PE2A, and PE1B. The sample output displays information about the BGP advertisement of RDs for the MDT default group 232.1.1.1. The output displays the MDT default groups advertised, the RDs and source addresses of sources sending to the MDT default groups, the BGP router ID of the advertising routers, and the BGP next hop address contained in the advertisements.

```
PE1A# show ip pim mdt bgp
MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:192.168.0.2                        192.168.0.2       192.168.0.2
   55:1111:192.168.0.8                        192.168.0.4       192.168.0.4
PE2A# show ip pim mdt bgp
MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:192.168.0.1                        192.168.0.1       192.168.0.1
   55:1111:192.168.0.8                        192.168.0.4       192.168.0.4
PE1B# show ip pim mdt bgp
MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:192.168.0.1                        192.168.0.9       192.168.0.9
   55:1111:192.168.0.2                        192.168.0.9       192.168.0.9
```

The following are sample outputs from the **show ip mroute proxy** command for PE1A, PE2A, and PE1B. The output displays information about the RPF Vectors learned by each PE router in this configuration example. The RPF Vector is the exit address of the ASBR router through which PIM messages are sent to reach inter-AS sources. The “Proxy” field displays the RPF Vectors learned by the PE routers. Each RPF Vector listed under the “Proxy” field is prepended by the RD associated with the RPF Vector. Because the PE routers are the assigners of the RPF Vector (that is, the PE routers insert the RPF Vector into PIM joins), 0.0.0.0 is the address displayed under the “Assigner” field in all the sample outputs. Finally, because PE routers learn the RPF Vector from BGP MDT SAFI updates, BGP MDT is displayed as the origin under the “Origin” field in all the outputs.

```
PE1A# show ip mroute proxy
(192.168.0.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.4       0.0.0.0          BGP MDT   00:13:07/stopped
PE2A# show ip mroute proxy
(192.168.0.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.4       0.0.0.0          BGP MDT   00:14:28/stopped
PE1B# show ip mroute proxy
(192.168.0.1, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.9       0.0.0.0          BGP MDT   00:35:19/stopped
(192.168.0.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.9       0.0.0.0          BGP MDT   00:35:49/stopped
```

The following is sample output from the **show ip mroute proxy** command from P1A. Because P routers learn the RPF Vector from the PIM joins sent from PE routers, the IP addresses of PE1A (10.0.3.1) and PE2A (10.0.3.2) are displayed under the “Assigner” field in the output for P1A. Because P1A learns the RPF Vector from encodings in the PIM join message, PIM is displayed as the origin under the “Origin” field.

```
P1A# show ip mroute proxy

(192.168.0.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.4       10.0.3.1         PIM       00:03:29/00:02:06
  55:1111/192.168.0.4       10.0.3.2         PIM       00:17:47/00:02:06
```

The following is sample output from the **show ip mroute proxy** command for ASBR1A and ASBR1B. If a router receives an RPF Vector that references a local interface (which occurs in an Option B deployment when an ASBR receives a RPF Vector owned by a local interface), the router discards the RPF Vector and performs a normal RPF lookup using information that the router learned from BGP MDT SAFI updates. In the output

for all ASBR routers, under the “Proxy” field, the word “local” is displayed instead of the RPF Vector because ASBR1A and ASBR1B are using local interfaces to perform RPF lookups for PIM joins with RPF Vectors that reference one of their local interfaces. The “Assigner” field displays the RPF address that sent the PIM join to the ASBR. Because the ASBRs learn the RPF Vectors from the PIM joins (the RPF Vectors that are subsequently discarded), PIM is displayed as the origin under the “Origin” field.

```
ASBR1A# show ip mroute proxy
(192.168.0.1, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.5.9         PIM       00:18:19/00:02:46
(192.168.0.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.5.9         PIM       00:18:50/00:02:24
(192.168.0.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.4.3         PIM       00:18:49/00:02:19
ASBR1B# show ip mroute proxy
(192.168.0.1, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.7.8         PIM       00:37:39/00:02:44
(192.168.0.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.7.8         PIM       00:38:10/00:02:19
(192.168.0.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.5.4         PIM       00:38:09/00:02:19
```

The following is sample output from the **show ip mroute** command for PE1A, PE2A, P1A, ASBR1A, ASBR1B, and PE1B. The sample outputs show the global table for the MDT default group 232.1.1.1. The output from this command confirms that all three PE routers (PE1A, PE2A, and PE1B) have joined the default MDT.

PE1A

```
PE1A# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(192.168.0.8, 232.1.1.1), 00:13:11/00:02:41, flags: sTIZV
  Incoming interface: GigabitEthernet2/0, RPF nbr 10.0.3.3, vector 192.168.0.4
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:13:11/00:00:00
(192.168.0.2, 232.1.1.1), 00:13:12/00:02:41, flags: sTIZ
  Incoming interface: GigabitEthernet2/0, RPF nbr 10.0.3.2
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:13:12/00:00:00
(192.168.0.1, 232.1.1.1), 00:13:12/00:03:27, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet2/0, Forward/Sparse-Dense, 00:13:11/00:02:50
```

PE2A

```
PE2A# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(192.168.0.8, 232.1.1.1), 00:17:05/00:02:46, flags: sTIZV
  Incoming interface: GigabitEthernet2/0, RPF nbr 10.0.3.3, vector 192.168.0.4
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:17:05/00:00:00
(192.168.0.1, 232.1.1.1), 00:17:05/00:02:46, flags: sTIZ
  Incoming interface: GigabitEthernet2/0, RPF nbr 10.0.3.1
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:17:05/00:00:00
(192.168.0.2, 232.1.1.1), 00:17:06/00:03:15, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet2/0, Forward/Sparse-Dense, 00:17:06/00:03:08
```

P1A

```
P1A# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(192.168.0.1, 232.1.1.1), 00:17:43/00:03:08, flags: sT
  Incoming interface: GigabitEthernet2/0, RPF nbr 10.0.3.1
  Outgoing interface list:
    GigabitEthernet3/0, Forward/Sparse-Dense, 00:17:43/00:02:51
(192.168.0.8, 232.1.1.1), 00:18:12/00:03:15, flags: sTV
  Incoming interface: GigabitEthernet3/0, RPF nbr 10.0.4.4, vector 192.168.0.4
  Outgoing interface list:
    GigabitEthernet2/0, Forward/Sparse-Dense, 00:18:12/00:03:15
(192.168.0.2, 232.1.1.1), 00:18:13/00:03:18, flags: sT
  Incoming interface: GigabitEthernet2/0, RPF nbr 10.0.3.2
  Outgoing interface list:
    GigabitEthernet3/0, Forward/Sparse-Dense, 00:18:13/00:03:18
```

ASBR1A

```
ASBR1A# show ip mroute 232.1.1.1
IP Multicast Routing Table
```

```
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.8, 232.1.1.1), 00:20:13/00:03:16, flags: sT
  Incoming interface: GigabitEthernet6/0, RPF nbr 10.0.7.12
  Outgoing interface list:
    GigabitEthernet5/0, Forward/Sparse-Dense, 00:20:13/00:02:46
(10.254.254.2, 232.1.1.1), 00:20:13/00:03:16, flags: sT
  Incoming interface: GigabitEthernet5/0, RPF nbr 10.0.6.5
  Outgoing interface list:
    GigabitEthernet6/0, Forward/Sparse-Dense, 00:20:13/00:02:39
```

ASBR1B

```
ASBR1B# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(192.168.0.1, 232.1.1.1), 00:37:43/00:03:16, flags: sTV
  Incoming interface: GigabitEthernet4/0, RPF nbr 10.0.5.4, vector 10.0.5.4
  Outgoing interface list:
    GigabitEthernet6/0, Forward/Sparse-Dense, 00:37:43/00:03:10
(192.168.0.8, 232.1.1.1), 00:38:14/00:03:16, flags: sT
  Incoming interface: GigabitEthernet6/0, RPF nbr 10.0.7.8
  Outgoing interface list:
    GigabitEthernet4/0, Forward/Sparse-Dense, 00:38:14/00:02:45
(192.168.0.2, 232.1.1.1), 00:38:14/00:03:16, flags: sTV
  Incoming interface: GigabitEthernet4/0, RPF nbr 10.0.5.4, vector 10.0.5.4
  Outgoing interface list:
    GigabitEthernet6/0, Forward/Sparse-Dense, 00:38:14/00:02:45
```

PE1B

```
PE1B# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
```

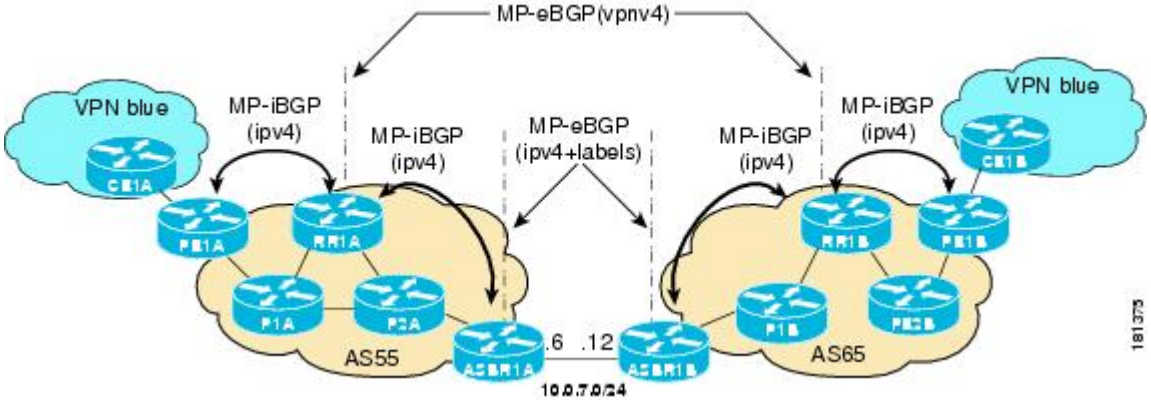
```
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(192.168.0.1, 232.1.1.1), 00:35:23/00:02:40, flags: sTIZV
  Incoming interface: GigabitEthernet6/0, RPF nbr 10.0.7.9, vector 192.168.0.9
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:35:23/00:00:00
(192.168.0.2, 232.1.1.1), 00:35:53/00:02:40, flags: sTIZV
  Incoming interface: GigabitEthernet6/0, RPF nbr 10.0.7.9, vector 192.168.0.9
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:35:53/00:00:00
(192.168.0.8, 232.1.1.1), 00:35:53/00:03:10, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet6/0, Forward/Sparse-Dense, 00:35:53/00:02:35
```

Configuring the Exchange of VPNv4 Routes Between RRs Using Multihop MP-EBGP

The following example shows how to configure support for MVPN inter-AS option C. This configuration is based on the sample inter-AS topology illustrated in the figure.

In the configuration example, MP-eBGP is used to exchange VPNv4 routes between RRs of different autonomous systems with the next hops for these routes exchanged between corresponding ASBR routers. Because the RRs in the two autonomous systems are not directly connected, multihop functionality is required to allow them to establish MP-eBGP peering sessions. The PE router next-hop addresses for the VPNv4 routes are exchanged between ASBR routers. In this configuration example, the exchange of these addresses between autonomous systems is established using IPv4 BGP label distribution, which enables the ASBRs to distribute IPv4 routes with MPLS labels.

Figure 22: Topology for MVPN Inter-AS Support Option C Configuration Example



The table provides information about the topology used for this inter-AS MVPN Option C configuration example.

Table 3: Topology Information for MVPN Inter-AS Support Option C Configuration Example

PE, RR, or ASBR Router	AS Number	Loopback0 Interfaces	Default MDT (PIM-SSM)
PE1A	55	10.254.254.2/32	232.1.1.1
RR1A	55	10.252.252.4/32	232.1.1.1

PE, RR, or ASBR Router	AS Number	Loopback0 Interfaces	Default MDT (PIM-SSM)
ASBR1A	55	10.254.254.6/32	232.1.1.1
PE1B	65	10.254.254.8/32	232.1.1.1
RR1B	65	10.252.252.10/32	232.1.1.1
ASBR1B	65	10.254.254.12/32	232.1.1.1

PE1A

```
!
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast rpf proxy vector
!
.
.
.
!
interface GigabitEthernet0/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 55
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.252.252.4 remote-as 55
 neighbor 10.252.252.4 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 mdt
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 send-community extended
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute connected
 redistribute static
 redistribute rip metric 50
 no synchronization
```

```
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

RR1A

```
!
router bgp 55
 neighbor 10.252.252.10 remote-as 65
 neighbor 10.252.252.10 ebgp-multihop 255
 neighbor 10.252.252.10 update-source Loopback0
 neighbor 10.254.254.2 remote-as 55
 neighbor 10.254.254.2 update-source Loopback0
 neighbor 10.254.254.6 remote-as 55
 neighbor 10.254.254.6 update-source Loopback0
 !
 address-family ipv4
 no neighbor 10.252.252.10 activate
 neighbor 10.254.254.2 activate
 neighbor 10.254.254.2 route-reflector-client
 neighbor 10.254.254.2 send-label
 neighbor 10.254.254.6 activate
 neighbor 10.254.254.6 route-reflector-client
 neighbor 10.254.254.6 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 mdt
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 next-hop-unchanged
 neighbor 10.254.254.2 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 neighbor 10.252.252.10 next-hop-unchanged
 neighbor 10.254.254.2 activate
 neighbor 10.254.254.2 send-community extended
 neighbor 10.254.254.2 route-reflector-client
 exit-address-family
!
```

ASBR1A

```
!
ip multicast-routing
ip multicast-routing vrf blue
!
.
.
.
!
interface GigabitEthernet7/0
 ip vrf forwarding blue
```

```
 ip pim sparse-mode
!
.
.
.
!
router bgp 55
 neighbor 10.0.7.12 remote-as 65
 neighbor 10.252.252.4 remote-as 55
 neighbor 10.252.252.4 update-source Loopback0
 !
 address-family ipv4
 redistribute isis level-2 route-map inter-as
 neighbor 10.0.7.12 activate
 neighbor 10.0.7.12 route-map IN in
 neighbor 10.0.7.12 route-map OUT out
 neighbor 10.0.7.12 send-label
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 next-hop-self
 neighbor 10.252.252.4 send-label
 no auto-summary
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

ASBR1B

```
!
ip multicast-routing
ip multicast-routing vrf blue
!
.
.
.
!
interface GigabitEthernet6/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 65
 neighbor 10.0.7.6 remote-as 55
 neighbor 10.252.252.10 remote-as 65
 neighbor 10.252.252.10 update-source Loopback0
 !
 address-family ipv4
 redistribute isis level-2 route-map inter-as
 neighbor 10.0.7.6 activate
 neighbor 10.0.7.6 route-map IN in
 neighbor 10.0.7.6 route-map OUT out
 neighbor 10.0.7.6 send-label
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 next-hop-self
 neighbor 10.252.252.10 send-label
```

```
 no auto-summary
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

RR1B

```
!
router bgp 65
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.252.252.4 remote-as 55
 neighbor 10.252.252.4 ebgp-multihop 255
 neighbor 10.252.252.4 update-source Loopback0
 neighbor 10.254.254.8 remote-as 65
 neighbor 10.254.254.8 update-source Loopback0
 neighbor 10.254.254.12 remote-as 65
 neighbor 10.254.254.12 update-source Loopback0
 !
 address-family ipv4
 no neighbor 10.252.252.4 activate
 neighbor 10.254.254.8 activate
 neighbor 10.254.254.8 route-reflector-client
 neighbor 10.254.254.8 send-label
 neighbor 10.254.254.12 activate
 neighbor 10.254.254.12 route-reflector-client
 neighbor 10.254.254.12 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 mdt
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 next-hop-unchanged
 neighbor 10.254.254.8 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 send-community extended
 neighbor 10.252.252.4 next-hop-unchanged
 neighbor 10.254.254.8 activate
 neighbor 10.254.254.8 send-community extended
 neighbor 10.254.254.8 route-reflector-client
 exit-address-family
!
```

PE1B

```
!
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
!
```

```
ip multicast-routing
ip multicast-routing vrf blue
ip multicast rpf proxy vector
!
.
.
.
!
interface GigabitEthernet12/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 65
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.252.252.10 remote-as 65
 neighbor 10.252.252.10 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 mdt
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute connected
 redistribute static
 redistribute rip metric 50
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

The following is sample output from the **show ip pim mdt bgp** command for PE1A and PE2A. The sample output displays information about the BGP advertisement of RDs for MDT default groups. The output displays the MDT default groups advertised, the RDs and source addresses of sources sending to the MDT default groups, the BGP router ID of the advertising routers, and the BGP next hop address contained in the advertisements.

```
PE1A# show ip pim mdt bgp
MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:10.254.254.8                       10.252.252.4      10.254.254.8
```

```
PE1B# show ip pim mdt bgp
MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:10.254.254.2                       10.252.252.10     10.254.254.2
```

The following is sample output from the **show ip mroute proxy** command from P1A, P2A, P1B, and P2B. Because P routers learn the RPF Vector from encodings in the PIM join message, PIM is displayed as the origin under the “Origin” field.

```
P1A# show ip mroute proxy
(10.254.254.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  10.254.254.6              10.0.2.2         PIM       00:15:37/00:02:57
P2A# show ip mroute proxy
(10.254.254.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  10.254.254.6              10.0.4.3         PIM       00:20:41/00:02:46
P1B# show ip mroute proxy
(10.254.254.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  10.254.254.12             10.0.10.9        PIM       00:29:38/00:02:16
P2B# show ip mroute proxy
(10.254.254.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  10.254.254.12             10.0.12.8        PIM       00:29:58/00:02:09
```

The following is sample output from the **show ip mroute** command for PE1A, P1A, P2A, ASBR1A, ASBR1B, P1B, P2B, and PE1B. The sample outputs show the global table for the MDT default group 232.1.1.1. The output from this command confirms that all three PE routers (PE1A, PE2A, and PE1B) have joined the default MDT.

PE1A

```
PE1A# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.8, 232.1.1.1), 00:12:27/00:02:43, flags: sTIZv
  Incoming interface: GigabitEthernet1/0, RPF nbr 10.0.2.3, vector 10.254.254.6
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:12:27/00:00:00
(10.254.254.2, 232.1.1.1), 00:14:40/00:03:12, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse-Dense, 00:12:27/00:03:06
```

P1A

```
P1A# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
```

```
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.2, 232.1.1.1), 00:15:40/00:03:25, flags: sT
  Incoming interface: GigabitEthernet1/0, RPF nbr 10.0.2.2
  Outgoing interface list:
    GigabitEthernet4/0, Forward/Sparse-Dense, 00:15:40/00:03:24
(10.254.254.8, 232.1.1.1), 00:15:40/00:03:25, flags: sTv
  Incoming interface: GigabitEthernet4/0, RPF nbr 10.0.4.5, vector 10.254.254.6
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse-Dense, 00:15:40/00:03:25
```

P2A

```
P2A# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.2, 232.1.1.1), 00:20:43/00:03:15, flags: sT
  Incoming interface: GigabitEthernet4/0, RPF nbr 10.0.4.3
  Outgoing interface list:
    GigabitEthernet5/0, Forward/Sparse-Dense, 00:20:43/00:03:15
(10.254.254.8, 232.1.1.1), 00:20:43/00:03:15, flags: sTv
  Incoming interface: GigabitEthernet5/0, RPF nbr 10.0.6.6, vector 10.254.254.6
  Outgoing interface list:
    GigabitEthernet4/0, Forward/Sparse-Dense, 00:20:43/00:03:14
```

ASBR1A

```
ASBR1A# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.8, 232.1.1.1), 00:20:13/00:03:16, flags: sT
  Incoming interface: GigabitEthernet6/0, RPF nbr 10.0.7.12
  Outgoing interface list:
```

```
    GigabitEthernet5/0, Forward/Sparse-Dense, 00:20:13/00:02:46
(10.254.254.2, 232.1.1.1), 00:20:13/00:03:16, flags: sT
  Incoming interface: GigabitEthernet5/0, RPF nbr 10.0.6.5
  Outgoing interface list:
    GigabitEthernet6/0, Forward/Sparse-Dense, 00:20:13/00:02:39
```

ASBR1B

```
ASBR1B# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.8, 232.1.1.1), 00:25:46/00:03:13, flags: sT
  Incoming interface: GigabitEthernet7/0, RPF nbr 10.0.8.11
  Outgoing interface list:
    GigabitEthernet6/0, Forward/Sparse-Dense, 00:25:46/00:03:04
(10.254.254.2, 232.1.1.1), 00:25:46/00:03:13, flags: sT
  Incoming interface: GigabitEthernet6/0, RPF nbr 10.0.7.6
  Outgoing interface list:
    GigabitEthernet7/0, Forward/Sparse-Dense, 00:25:46/00:03:07
```

P1B

```
P1B# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.8, 232.1.1.1), 00:29:41/00:03:17, flags: sT
  Incoming interface: GigabitEthernet10/0, RPF nbr 10.0.10.9
  Outgoing interface list:
    GigabitEthernet7/0, Forward/Sparse-Dense, 00:29:41/00:02:56
(10.254.254.2, 232.1.1.1), 00:29:41/00:03:17, flags: sTv
  Incoming interface: GigabitEthernet7/0, RPF nbr 10.0.8.12, vector 10.254.254.12
  Outgoing interface list:
    GigabitEthernet10/0, Forward/Sparse-Dense, 00:29:41/00:02:44
```

P2B

```
P2B# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
```

```
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.8, 232.1.1.1), 00:30:01/00:03:25, flags: sT
  Incoming interface: GigabitEthernet11/0, RPF nbr 10.0.12.8
  Outgoing interface list:
    GigabitEthernet10/0, Forward/Sparse-Dense, 00:30:01/00:02:30
(10.254.254.2, 232.1.1.1), 00:30:01/00:03:25, flags: sTv
  Incoming interface: GigabitEthernet10/0, RPF nbr 10.0.10.11, vector 10.254.254.12
  Outgoing interface list:
    GigabitEthernet11/0, Forward/Sparse-Dense, 00:30:01/00:02:36
```

PE1B

```
PE1B# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.254.254.2, 232.1.1.1), 00:31:22/00:02:55, flags: sTIZv
  Incoming interface: GigabitEthernet11/0, RPF nbr 10.0.12.9, vector 10.254.254.12
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:31:22/00:00:00
(10.254.254.8, 232.1.1.1), 00:33:35/00:03:25, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet11/0, Forward/Sparse-Dense, 00:31:22/00:03:22
```

Additional References

Related Documents

Related Topic	Document Title
Multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
draft-ietf-pim-rpf-vector	*The RPF Vector TLV*

Standard	Title
draft-ietf-l3vpn-rfc2547bis[1]	*BGP/MPLS IP VPNs*

[1] The Internet draft standard draft-ietf-l3vpn-rfc2547bis is generally referred to as RFC 2547bis.

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4364[2]	BGP/MPLS IP Virtual Private Networks (VPN)
RFC 5496	The Reverse Path Forwarding (RPF) Vector TLV

[2] RFC 4364 is the latest RFC standard and obsoletes RFC 2547 (and the later RFC2547bis Internet draft standard).

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring Multicast VPN Inter-AS Support

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

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

Table 4: Feature Information for Configuring Multicast VPN Inter-AS Support

Feature Name	Releases	Feature Information
BGP Multicast Inter-AS VPN	12.0(29)S 12.2(33)SRA 12.2(31)SB2 12.2(33)SXH 12.4(20)T 15.0(1)S	The BGP Multicast Inter-AS VPN feature introduces the IPv4 MDT SAFI in BGP. The MDT SAFI is a transitive multicast capable connector attribute that is defined as an IPv4 address family in BGP. The MDT SAFI is designed to support inter-AS VPN peering sessions. The following commands were introduced or modified by this feature: **address-family ipv4** (BGP), **clear bgp ipv4 mdt**, **show ip bgp ipv4**.
Multicast VPN Inter-AS Support	12.0(30)S 12.2(33)SRA 12.2(31)SB2 12.2(33)SXH 12.4(20)T 15.0(1)S	The Multicast VPN Inter-AS support feature enables MDTs used for MVPNs to span multiple autonomous systems. Benefits include increased multicast coverage to customers that require multicast to span multiple service providers in an MPLS Layer 3 VPN service with the flexibility to support all options described in RFC 4364. Additionally, the Multicast VPN Inter-AS Support feature may be used to consolidate an existing MVPN service with another MVPN service, such as the case with a company merger or acquisition. The following commands were introduced or modified by this feature: **ip multicast rpf proxy vector**, and **show ip mroute**, **show ip pim neighbor**, **show ip rpf**.