IP Multicast: MVPN Configuration Guide, Cisco IOS XE Release 2

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

CONTENTS

Configuring Multicast VPN

The Multicast VPN (MVPN) feature provides the ability to support multicast over a Layer 3 Virtual Private Network (VPN). As enterprises extend the reach of their multicast applications, service providers can accommodate these enterprises over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data to an MPLS VPN network core.

Historically, point-to-point tunnels were the only way to connect through a service provider network. Although such tunneled networks tend to have scalability issues, they represented the only means of passing IP multicast traffic through a VPN.

Because Layer 3 VPNs support only unicast traffic connectivity, deploying in conjunction with a Layer 3 VPN allows service providers to offer both unicast and multicast connectivity to Layer 3 VPN customers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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

Prerequisites for Configuring Multicast VPN

- Before performing the tasks in this module, you should be familiar with the concepts described in the “ IP Multicast Technology Overview ” module.
- The tasks in this module assume that IP multicasting has been enabled and that the Protocol Independent Multicast (PIM) interfaces have been configured using the tasks described in the“ Configuring Basic IP Multicast ” module.

Restrictions for Configuring Multicast VPN

- The update source interface for the Border Gateway Protocol (BGP) peerings must be the same for all BGP peerings configured on the router in order for the default multicast distribution tree (MDT) to be configured properly. If you use a loopback address for BGP peering, then PIM sparse mode must be enabled on the loopback address.
- The **ip mroute-cache** command must be enabled on the loopback interface used as the BGP peering interface in order for distributed multicast switching to function on the platforms that support it. The **no ip mroute-cache** command must not be present on these interfaces.
- MVPN does not support multiple BGP peering update sources.
- Data MDTs are not created for VPN routing and forwarding instance (VRF) PIM dense mode multicast streams because of the flood and prune nature of dense mode multicast flows and the resulting periodic bring-up and tear-down of such data MDTs.
- Multiple BGP update sources are not supported and configuring them can break MVPN reverse path forwarding (RPF) checking. The source IP address of the MVPN tunnels is determined by the highest IP address used for the BGP peering update source. If this IP address is not the IP address used as the BGP peering address with the remote provider edge (PE) router, MVPN will not function properly.
- When a customer configures an ASR 1000 series router as a PE router in an MVPN network, the PE router may be subject to multicast traffic duplication after an RP switchover. All MVPN traffic will recover to normal replication state after a short period.

Information About Configuring Multicast VPN

Multicast VPN Operation

MVPN IP allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This feature supports routing and forwarding of multicast packets for each individual VRF instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone.

A VPN is network connectivity across a shared infrastructure, such as an Internet service provider (ISP). Its function is to provide the same policies and performance as a private network, at a reduced cost of ownership, thus creating many opportunities for cost savings through operations and infrastructure.

An MVPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. The use of an MVPN to interconnect an enterprise network in this way does not change the way that enterprise network is administered, nor does it change general enterprise connectivity.

Benefits of Multicast VPN

- Provides a scalable method to dynamically send information to multiple locations.
- Provides high-speed information delivery.
- Provides connectivity through a shared infrastructure.

Multicast VPN Routing and Forwarding and Multicast Domains

MVPN introduces multicast routing information to the VPN routing and forwarding table. When a provider edge (PE) router receives multicast data or control packets from a customer edge (CE) router, forwarding is performed according to the information in the Multicast VPN routing and forwarding instance (MVRF). MVPN does not use label switching.

A set of MVRFs that can send multicast traffic to each other constitutes a multicast domain. For example, the multicast domain for a customer that wanted to send certain types of multicast traffic to all global employees would consist of all CE routers associated with that enterprise.

Multicast Distribution Trees

MVPN establishes a static default MDT for each multicast domain. The default MDT defines the path used by PE routers to send multicast data and control messages to every other PE router in the multicast domain.

If Source Specific Multicast (SSM) is used as the core multicast routing protocol, then the multicast IP addresses used for the default and data multicast distribution tree (MDT) must be configured within the SSM range on all PE routers.

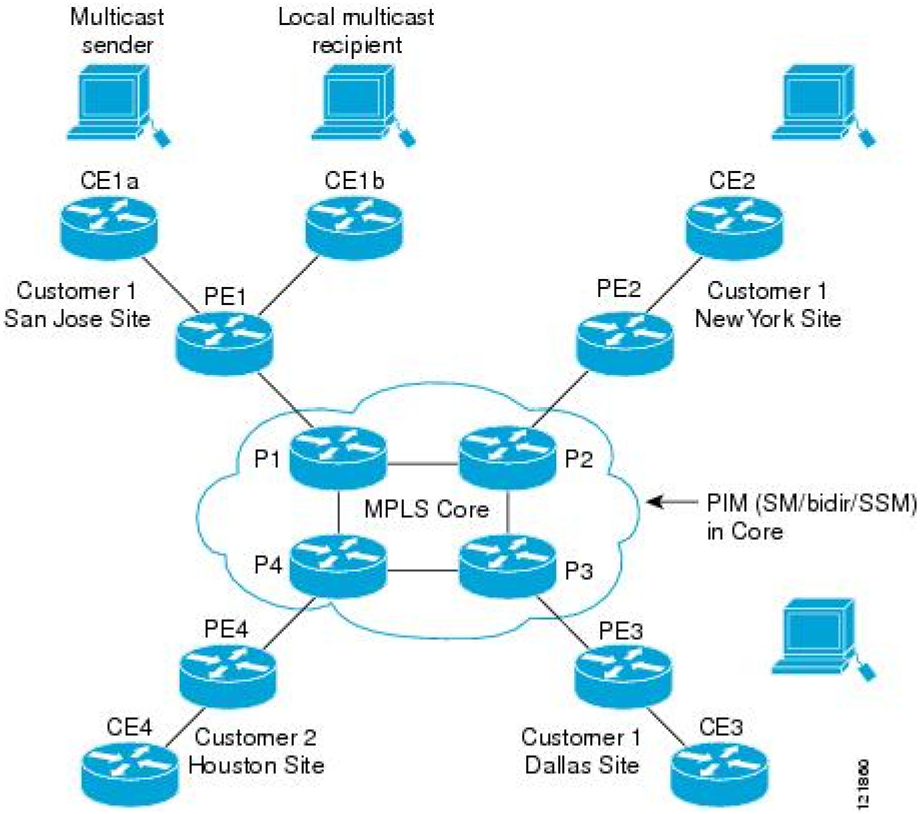
MVPN also supports the dynamic creation of MDTs for high-bandwidth transmission. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-router or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE router creates the data MDT and sends a User Datagram Protocol (UDP) message, which contains information about the data MDT to all routers on the default MDT. The statistics to determine whether a multicast stream has exceeded the data MDT threshold are examined once every second. After a PE router sends the UDP message, it waits 3 more seconds before switching over; 13 seconds is the worst case switchover time and 3 seconds is the best case.

Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries regardless of the value of the individual source data rate.

In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. A one-way multicast presentation is occurring in San Jose. The service provider network supports all three sites associated with this customer, in addition to the Houston site of a different enterprise customer.

The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. PE4 is not part of the default MDT, because it is associated with a different customer. The figure shows that no data flows along the default MDT, because no one outside of San Jose has joined the multicast.

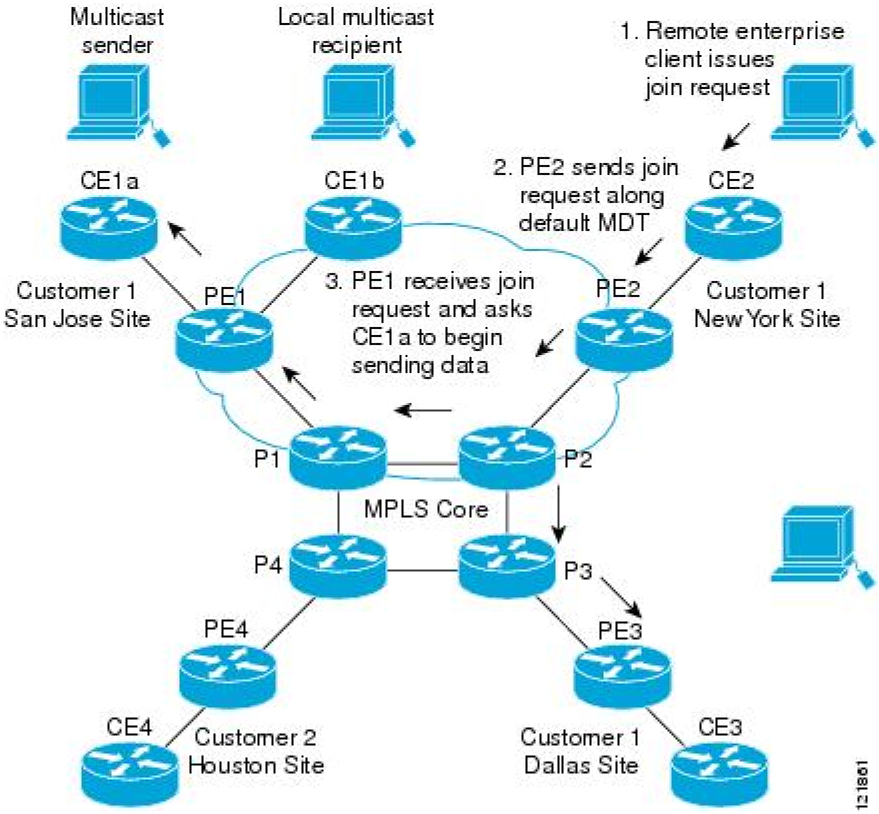
Figure 1 Default Multicast Distribution Tree Overview



An employee in New York joins the multicast session. The PE router associated with the New York site sends a join request that flows across the default MDT for the multicast domain of the customer. PE1, the

PE router associated with the multicast session source, receives the request. The figure depicts that the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 2 Initializing the Data MDT


Multicast sender
Local multicast recipient
1. Remote enterprise client issues join request
CE1a
CE1b
2. PE2 sends join request along default MDT
CE2
Customer 1 San Jose Site
PE1
3. PE1 receives join request and asks CE1a to begin sending data
PE2
Customer 1 New York Site
P1
P2
MPLS Core
P4
P3
PE4
PE3
CE4
Customer 2 Houston Site
Customer 1 Dallas Site
CE3
121861

The CE router (CE1a) begins to send the multicast data to the associated PE router (PE1), which sends the multicast data along the default MDT. Immediately sending the multicast data, PE1 recognizes that the multicast data exceeds the bandwidth threshold for which a data MDT should be created. Therefore, PE1 creates a data MDT, sends a message to all routers using the default MDT that contains information about the data MDT, and, three seconds later, begins sending the multicast data for that particular stream using the data MDT. Only PE2 has interested receivers for this source, so only PE2 will join the data MDT and receive traffic on it.

PE routers maintain a PIM relationship with other PE routers over the default MDT and a PIM relationship with its directly attached PE routers.

Multicast Tunnel Interface

An MVRF, which is created per multicast domain, requires the router to create a tunnel interface from which all MVRF traffic is sourced. A multicast tunnel interface is an interface the MVRF uses to access the multicast domain. It can be thought of as a conduit that connects an MVRF and the global MVRF. One tunnel interface is created per MVRF.

MDT Address Family in BGP for Multicast VPN

The **mdt** keyword has been added to the **address-family ipv4** command to configure an MDT address-family session. MDT address-family sessions are used to pass the source PE address and MDT group address to PIM using Border Gateway Protocol (BGP) MDT Subaddress Family Identifier (SAFI) updates.

BGP Advertisement Methods for Multicast VPN Support

In a single autonomous system, if the default MDT for an MVPN is using PIM sparse mode (PIM-SM) with a rendezvous point (RP), then PIM is able to establish adjacencies over the Multicast Tunnel Interface (MTI) because the source PE and receiver PE discover each other through the RP. In this scenario, the local PE (the source PE) sends register messages to the RP, which then builds a shortest-path tree (SPT) toward the source PE. The remote PE, which acts as a receiver for the MDT multicast group, then sends (*, G) joins toward the RP and joins the distribution tree for that group.

However, if the default MDT group is configured in a PIM Source Specific Multicast (PIM-SSM) environment rather than a PIM-SM environment, the receiver PE needs information about the source PE and the default MDT group. This information is used to send (S, G) joins toward the source PE to build a distribution tree from the source PE (without the need for an RP). The source PE address and default MDT group address are sent using BGP.

The table lists the BGP advertisement methods for sending the source PE address and the default MDT group that are available (by Cisco IOS XE release).

Table 1 BGP Advertisement Methods for MVPN

Cisco IOS XE Release	BGP Advertisement Method
Cisco IOS XE Release 2.5	• Extended communities • BGP address family MDT SAFI

BGP Extended Community

When BGP extended communities are used, the PE loopback (source address) information is sent as a VPNv4 prefix using Route Distinguisher (RD) Type 2 (to distinguish it from unicast VPNv4 prefixes). The MDT group address is carried in a BGP extended community. Using a combination of the embedded source in the VPNv4 address and the group in the extended community, PE routers in the same MVRF instance can establish SSM trees to each other.

Note Prior to the introduction of MDT SAFI support, the BGP extended community attribute was used as an interim solution to advertise the IP address of the source PE and default MDT group before IETF standardization. A BGP extended community attribute in an MVPN environment, however, has certain limitations: it cannot be used in inter-AS scenarios (because the attribute is nontransitive), and it uses RD Type 2 (which is not a supported standard).

BGP MDT SAFI

In Cisco IOS software releases that support the MDT SAFI, the source PE address and the MDT group address are passed to PIM using BGP MDT SAFI updates. The RD type has changed to RD type 0 and BGP determines the best path for the MDT updates before passing the information to PIM.

Note To prevent backwards-compatibility issues, BGP allows the communication of the older style updates with peers that are unable to understand the MDT SAFI address family.

In Cisco IOS software releases that support the MDT SAFI, the MDT SAFI address family needs to be explicitly configured for BGP neighbors using the **address-family ipv4 mdt** command. Neighbors that do not support the MDT SAFI still need to be enabled for the MDT SAFI in the local BGP configuration. Prior to the introduction of the MDT SAFI, additional BGP configuration from the VPNv4 unicast configuration was not needed to support MVPN.

Because the new MDT SAFI does not use BGP route-target extended communities, the regular extended community methods to filter these updates no longer apply. As a result, the **match mdt-group** route-map configuration command has been added to filter on the MDT group address using Access Control Lists (ACLs). These route maps can be applied--inbound or outbound--to the IPv4 MDT address-family neighbor configuration.

Automigration to the MDT SAFI

When a Cisco IOS XE release is migrated to the MDT SAFI, existing VPNv4 neighbors will be automatically configured for the MDT SAFI upon bootup based on the presence of an existing default MDT configuration (that is, pre-MDT SAFI configurations will be automatically converted to an MDT SAFI configuration upon bootup). In addition, when a default MDT configuration exists and a VPNv4 neighbor in BGP is configured, a similar neighbor in the IPv4 MDT address family will be automatically configured.

Note Because there is no VRF configuration on route reflectors (RRs), automigration to the MDT SAFI will not be triggered on RRs. The MDT SAFI configuration, thus, will need to be manually configured on RRs. Having a uniform MDT transmission method will reduce processing time on the routers (because MDT SAFI conversion is not necessary).

Guidelines for Configuring the MDT SAFI

- We recommend that you configure the MDT SAFI on all routers that participate in the MVPN. Even though the benefits of the MDT SAFI are for SSM tree building, the MDT SAFI must also be configured when using MVPN with the default MDT group for PIM-SM. From the multicast point of view, the MDT SAFI is not required for MVPN to work within a PIM-SM core. However, in certain scenarios, the new address family must be configured in order to create the MTI. Without this notification, the MTI would not be created and MVPN would not function (even with PIM-SM).

- For backward-compatible sessions, extended communities must be enabled on all MDT SAFI peers. In a pure MDT SAFI environment there is no need to configure extended communities explicitly for MVPN. However, extended communities will be needed for VPNv4 interior BGP (iBGP) sessions to relay the route-target. In a hybrid (MDT SAFI and pre-MDT SAFI) environment, extended communities must be configured to send the embedded source in the VPNv4 address and the MDT group address to MDT SAFI neighbors.

Guidelines for Upgrading a Network to Support the MDT SAFI

When moving from a pre-MDT SAFI to an MDT SAFI environment, the utmost care should be taken to minimize the impact to the MVPN service. The unicast service will not be affected, other than the outage due to the reload and recovery. To upgrade a network to support the MDT SAFI, we recommend that you perform the following steps:

1. Upgrade the PEs in the MVPN to a Cisco IOS XE release that supports the MDT SAFI. Upon bootup, the PE configurations will be automigrated to the MDT SAFI. For more information about the automigration to the MDT SAFI functionality, see the Guidelines for Upgrading a Network to Support the MDT SAFI, page 8 section.
2. After the PEs have been upgraded, upgrade the RRs and enable the MDT SAFI for all peers providing MVPN service. Enabling or disabling the MDT SAFI will reset the BGP peer relationship for all address families; thus, a loss of routing information may occur.

Note In the case of a multihomed BGP RR scenario, one of the RRs must be upgraded and configured last. The upgraded PEs will use this RR to relay MDT advertisements while the other RRs are being upgraded.

Supported Policy

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

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

How to Configure Multicast VPN

Configuring a Default MDT Group for a VRF

Perform this task to configure a default MDT group for a VRF.

The default MDT group must be the same group configured on all routers that belong to the same VPN. The source IP address will be the address used to source the BGP sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing**
4. **ip multicast-routing vrf** *vrf-name*
5. **ip vrf** *vrf-name*
6. **mdt default** *group-address*

DETAILED STEPS

	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	**ip multicast-routing** **Example:** `Router(config)# ip multicast-routing`	Enables multicast routing.
Step 4	**ip multicast-routing vrf** *vrf-name* **Example:** `Router(config)# ip multicast-routing vrf vrf1`	Supports the MVPN VRF instance.

	Command or Action	Purpose
Step 5	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 6	**mdt default** *group-address* **Example:** `Router(config-vrf)# mdt default 232.0.0.1`	Configures the multicast group address range for data MDT groups for a VRF. • A tunnel interface is created as a result of this command. • By default, the destination address of the tunnel header is the *group-address* argument.

Configuring the MDT Address Family in BGP for Multicast VPN

Perform this task to configure an MDT address family session on PE routers to establish MDT peering sessions for MVPN.

Before MVPN 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
- Network Layer Reachability Information (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. **neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**]
7. **exit**
8. **address-family vpnv4**
9. **neighbor** *neighbor-address* **activate**
10. **neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**]
11. **end**

DETAILED STEPS

	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.
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	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables community and (or) extended community exchange with the specified neighbor.
Step 7	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 8	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.
Step 9	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the VPNv4 address family for this neighbor.
Step 10	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables community and (or) extended community exchange with the specified neighbor.

Step 11	**end**
	Example:
	`Router(config-router-af)# end`
		Exits address family configuration mode and enters privileged EXEC mode.

Configuring the Data Multicast Group

Perform this task to configure a data MDT group.

A data MDT group can include a maximum of 256 multicast groups per VPN per VRF per PE router. Multicast groups used to create the data MDT group are dynamically chosen from a pool of configured IP addresses.

- Before configuring a default MDT group, the VPN must be configured for multicast routing as described in the Configuring a Default MDT Group for a VRF, page 9.
- All access lists needed when using the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the “ Creating an IP Access List and Applying It to an Interface ” module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **mdt data** *group-address-range wildcard-bits* [**threshold** *kb/s*] [**list** *access-list*]
5. **mdt log-reuse**
6. **exit**

DETAILED STEPS

	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	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 4	**mdt data** *group-address-range wildcard-bits* [**threshold** *kb/s*] [**list** *access-list*] **Example:** `Router(config-vrf)# mdt data 239.192.20.32 0.0.0.15 threshold 1`	Specifies a range of addresses to be used in the data MDT pool. • For the *group-address-range* and *wildcard-bits* arguments, specify a a multicast group address range. The range is from 224.0.0.1 to 239.255.255.255. Because the range of addresses used in the data MDT pool are multicast group addresses (Class D addresses), there is no concept of a subnet; therefore, you can use all addresses in the mask (wildcard) range that you specify for the *wildcard-bits* argument. • The threshold is in kb/s. The range is from 1 through 4294967. • Use the optional **list** keyword and *access-list* argument to define the (S, G) MVPN entries to be used in a data MDT pool, which would further limit the creation of a data MDT pool to the particular (S, G) MVPN entries defined in the access list specified for the *access-list*argument

	Command or Action	Purpose
Step 5	**mdt log-reuse** **Example:** `Router(config-vrf)# mdt log-reuse`	(Optional) Enables the recording of data MDT reuse and generates a syslog message when a data MDT has been reused.
Step 6	**exit** **Example:** `Router(config-vrf)# exit`	Returns to global configuration mode.

Configuring Multicast Routes and Information

Perform this task to limit the number of multicast routes that can be added in a router.

- Before configuring a default MDT group, the VPN must be configured for multicast routing as described in the Configuring a Default MDT Group for a VRF, page 9.
- All access lists needed when using the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the “ Creating an IP Access List and Applying It to an Interface ” module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*]
4. **ip multicast mrinfo-filter** *access-list*

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 3	**ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*] **Example:** `Router(config)# ip multicast vrf cisco route-limit 200000 20000`	Sets the mroute limit and the threshold parameters.
Step 4	**ip multicast mrinfo-filter** *access-list* **Example:** `Router(config)# ip multicast mrinfo-filter 4`	Filters the multicast router information request packets for all sources specified in the access list.

Verifying Information for the MDT Default Group

Perform this task to verify information about the MDT default group.

SUMMARY STEPS

1. **enable**
2. **show ip msdp** [**vrf** *vrf-name*] **peer** [*peer-address* | *peer-name*]
3. **show ip msdp** [vrf *vrf-name*] **summary**
4. **show ip pim** [**vrf** *vrf-name*] **mdt bgp**
5. **show ip pim** [**vrf** *vrf-name*] **mdt send**
6. **show ip pim mdt history**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip msdp** [**vrf** *vrf-name*] **peer** [*peer-address* | *peer-name*]

Enter the **show ip msdp peer**command to verify detailed information about MSDP peer 224.135.250.116:

Example:

```
Router# show ip msdp peer 224.135.250.116

MSDP Peer 224.135.250.116 (rtp5-rp1.cisco.com), AS 109 (configured AS)
Description:
 Connection status:
   State: Up, Resets: 9, Connection source: Loopback2 (228.69.199.17)
```

```
    Uptime(Downtime): 1d10h, Messages sent/received: 436765/429062
    Output messages discarded: 0
    Connection and counters cleared 1w2d     ago
  SA Filtering:
    Input (S,G) filter: none, route-map: none
    Input RP filter: none, route-map: none
    Output (S,G) filter: none, route-map: none
    Output RP filter: none, route-map: none
  SA-Requests:
    Input filter: none
    Sending SA-Requests to peer: disabled
  Peer ttl threshold: 0
  SAs learned from this peer: 32, SAs limit: 500
  Input queue size: 0, Output queue size: 0
```

Step 3 **show ip msdp** [vrf *vrf-name*] **summary**

Enter the **show ip msdp summary** command to display MSDP peer status:

Example:

```
Router# show ip msdp summary

MSDP Peer Status Summary
Peer Address     AS    State    Uptime/  Reset SA    Peer Name
                                Downtime Count Count
224.135.250.116  109   Up       1d10h    9     111   rtp5-rp1
*144.228.240.253 1239  Up       14:24:00 5     4010  sl-rp-stk
172.16.253.19    109   Up       12:36:17 5     10    rtp4-rp1
172.16.170.110   109   Up       1d11h    9     12    ams-rp1
```

Step 4 **show ip pim** [**vrf** *vrf-name*] **mdt bgp**

To display information about and to verify information about the BGP advertisement of the RD for the MDT default group, use the **show ip pim mdt bgp** command in privileged EXEC mode:

Example:

```
Router# show ip pim mdt bgp

MDT-default group 232.2.1.4
rid:1.1.1.1 next_hop:1.1.1.1
```

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

To display detailed information about and to verify information regarding the MDT data group, perform the following steps.

Enter the **show ip pim mdt send** command to show the MDT advertisements that a specified router has made:

Example:

```
Router# show ip pim mdt send

MDT-data send list for VRF:vpn8
  (source, group)                    MDT-data group      ref_count
  (10.100.8.10, 225.1.8.1)           232.2.8.0           1
  (10.100.8.10, 225.1.8.2)           232.2.8.1           1
  (10.100.8.10, 225.1.8.3)           232.2.8.2           1
  (10.100.8.10, 225.1.8.4)           232.2.8.3           1
  (10.100.8.10, 225.1.8.5)           232.2.8.4           1
  (10.100.8.10, 225.1.8.6)           232.2.8.5           1
  (10.100.8.10, 225.1.8.7)           232.2.8.6           1
  (10.100.8.10, 225.1.8.8)           232.2.8.7           1
  (10.100.8.10, 225.1.8.9)           232.2.8.8           1
  (10.100.8.10, 225.1.8.10)          232.2.8.9           1
```

Step 6 **show ip pim mdt history**

Enter the **show ip pim mdt history** command to display the data MDTs that have been reused during the past configured interval.

Example:

```
Router# show ip pim vrf vrf1 mdt history interval 20

   MDT-data send history for VRF - vrf1 for the past 20 minutes
MDT-data group        Number of reuse
     10.9.9.8           3
     10.9.9.9           2
```

Configuration Examples for Multicast VPN

Configuring MVPN and SSM Example

In the following example, PIM-SSM is configured in the backbone. Therefore, the default and data MDT groups are configured within the SSM range of IP addresses. Inside the VPN, PIM-SM is configured and only Auto-RP announcements are accepted.

```
ip vrf vrf1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 mdt default 232.0.0.1
 mdt data 232.0.1.0 0.0.0.255 threshold 500 list 101
!
ip pim ssm default
ip pim vrf vrf1 accept-rp auto-rp
```

Enabling a VPN for Multicast Routing Example

In the following example, multicast routing is enabled with a VPN routing instance named vrf1:

```
ip multicast-routing vrf1
```

Configuring the MDT Address Family in BGP for Multicast VPN Example

In the following example, an MDT address family session is configured on a PE router to establish MDT peering sessions for MVPN:

```
!
ip vrf test
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.0.0.1
!
ip multicast-routing
ip multicast-routing vrf test
!
router bgp 55
.
.
.
!
 address-family vpnv4
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 send-community both
!
 address-family ipv4 mdt
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 send-community both
!
```

Configuring the Multicast Group Address Range for Data MDT Groups Example

In the following example, the VPN routing instance is assigned a VRF named blue. The MDT default group for a VPN VRF is 239.1.1.1, and the multicast group address range for MDT groups is 239.1.2.0 with wildcard bits of 0.0.0.3:

```
ip vrf blue
 rd 55:1111
 route-target both 55:1111
 mdt default 239.1.1.1
 mdt data 239.1.2.0 0.0.0.3
 end
```

Limiting the Number of Multicast Routes Example

In the following example, the number of multicast routes that can be added in to a multicast routing table is set to 200,000 and the threshold value of the number of mroutes that will cause a warning message to occur is set to 20,000:

```
!
ip multicast-routing distributed
ip multicast-routing vrf cisco distributed
ip multicast cache-headers
ip multicast route-limit 200000 20000
ip multicast vrf cisco route-limit 200000 20000
no mpls traffic-eng auto-bw timers frequency 0
!
```

Additional References

Related Documents

Related Topic	Document Title
Extranet MVPN concepts, tasks, and configuration examples	“ Configuring Multicast VPN Extranet Support ” module
Inter-AS MVPN concepts, tasks, and configuration examples	“ Configuring Multicast VPN Inter-AS Support ” module
MVPN MIB concepts and tasks	“ Multicast VPN MIB ” module
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
CISCO_MVPN_MIB.my	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

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

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

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

Table 2 Feature Information for Configuring Multicast VPN

Feature Name	Releases	Feature Information
Multicast VPN--IP Multicast Support of MPLS VPNs	Cisco IOS XE Release 2.5	The Multicast VPN feature provides the ability to support multicast over a Layer 3 Virtual Private Network (VPN). As enterprises extend the reach of their multicast applications, service providers can accommodate these enterprises over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data to an MPLS VPN network core.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Configuring Multicast VPN Extranet Support

The Multicast VPN Extranet Support feature (sometimes referred to as the MVPN Extranet Support feature) enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers.

This module describes the concepts and the tasks related to configuring Multicast VPN Extranet Support.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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

Prerequisites for Configuring Multicast VPN Extranet Support

- You are familiar with IP multicast concepts and configuration tasks.
- You are familiar with Multicast VPN (MVPN) concepts and configuration tasks.
- You are familiar with Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) concepts and configuration tasks.

Restrictions for Configuring Multicast VPN Extranet Support

- The Multicast VPN Extranet Support feature supports only Protocol Independent Multicast (PIM) sparse mode (PIM-SM) and Source Specific Multicast (SSM) traffic; PIM dense mode (PIM-DM) and bidirectional PIM (bidir-PIM) traffic are not supported.
- When configuring extranet MVPNs in a PIM-SM environment, the source and the rendezvous point (RP) must reside in the same site of the MVPN behind the same provider edge (PE) router.

Information About Multicast VPN Extranet Support

- Overview of MVPN Extranet Support, page 24
- Configuration Guidelines for MVPN Extranet Support, page 26
- Multicast VPN Extranet VRF Select, page 28

Overview of MVPN Extranet Support

An extranet can be viewed as part of a company's intranet that is extended to users outside the company. It has also been described as a "state of mind" in which a VPN is used as a way to do business with other companies and to sell products and content to customers and companies. An extranet is a VPN connecting the corporate site or sites to external business partners or suppliers to securely share part of a business's information or operations among them.

MPLS VPNs inherently provide security, ensuring that users access only appropriate information. MPLS VPN extranet services offer extranet users unicast connectivity without compromising the integrity of their corporate data. The Multicast VPN Extranet Support feature extends this offer to include multicast connectivity to the extranet community of interest.

The Multicast VPN Extranet Support feature enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers. Using this feature, service providers can offer multicast extranet contracts to meet various business partnership requirements, including short-term, annual, and rolling contracts.

- Benefits of MVPN Extranet Support, page 24
- Components of an Extranet MVPN, page 24
- Solution for MVPN Extranet Support, page 25

Benefits of MVPN Extranet Support

The Multicast VPN Extranet Support feature can be used to solve such business problems as:

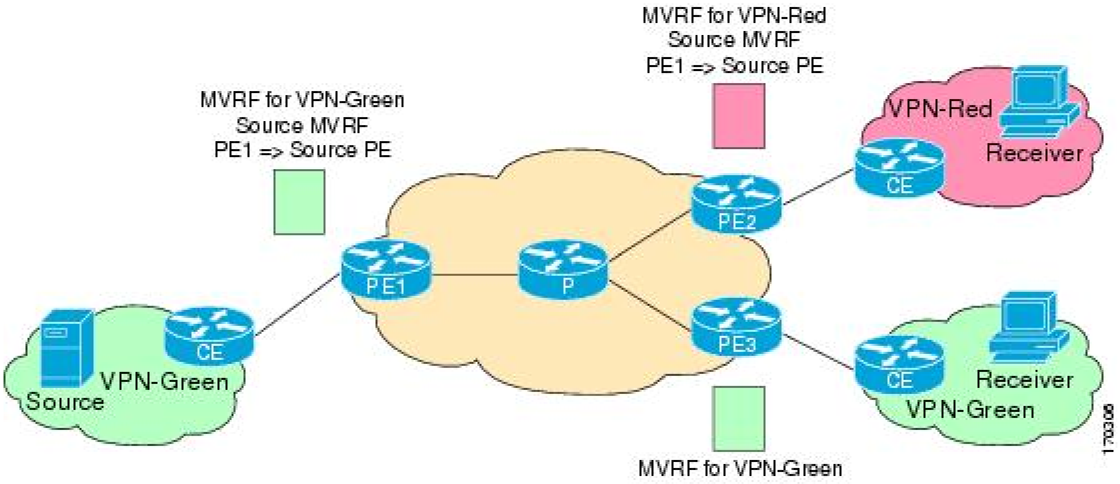
- Efficient content distribution between enterprises
- Efficient content distribution from service providers or content providers to their different enterprise VPN customers

Components of an Extranet MVPN

The figure illustrates the components that constitute an extranet MVPN:

- **MVRF** --Multicast VPN routing and forwarding (VRF) instance. An MVRF is a multicast-enabled VRF. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) router.
- **Source MVRF** --An MVRF that can reach the source through a directly connected customer edge (CE) router.
- **Receiver MVRF** --An MVRF to which receivers are connected through one or more CE devices.
- **Source PE** --A PE router that has a multicast source behind a directly connected CE router.
- **Receiver PE** --A PE router that has one or more interested receivers behind a directly connected CE router.

Figure 3 Components of an Extranet MVPN



Solution for MVPN Extranet Support

For unicast, there is no difference between an intranet or extranet from a routing perspective; that is, when a VRF imports a prefix, that prefix is reachable through a label-switched path (LSP). If the enterprise owns the prefix, the prefix is considered a part of the corporate intranet; otherwise, the prefix is considered a part of an extranet. For multicast, however, the reachability of a prefix (especially through an LSP) is not sufficient to build a multicast distribution tree (MDT).

In order to provide support for extranet MVPN services, the same default MDT group must be configured in the source and receiver MVRF. Prior to the introduction of the Multicast VPN Extranet Support feature, there were challenges that prevented service providers from providing extranet MVPN services:

- The source MVRF may not have been configured with a default MDT group, or it may have been configured with a different MDT group as compared to the receiver MVRF. In the former case there was no way for the source MVRF to forward multicast streams to extranet sites, and in the latter case, there was no way for the separate MVRFs to be linked.
- It was not possible to maintain a forwarding table in cases where the RPF interface and outgoing interfaces belong to different VRFs.

The Multicast VPN Extranet Support feature solves these challenges as follows:

- The receiver and source MVRF multicast route (mroute) entries are linked.
- The Reverse Path Forwarding (RPF) check relies on unicast routing information to determine the interface through which the source is reachable. This interface is used as the RPF interface.

Configuration Guidelines for MVPN Extranet Support

Two configuration options are available to provide extranet MVPN services:

- Option 1--Configure the receiver MVRF on the source PE router.
- Option 2--Configure the source MVRF on the receiver PE router.

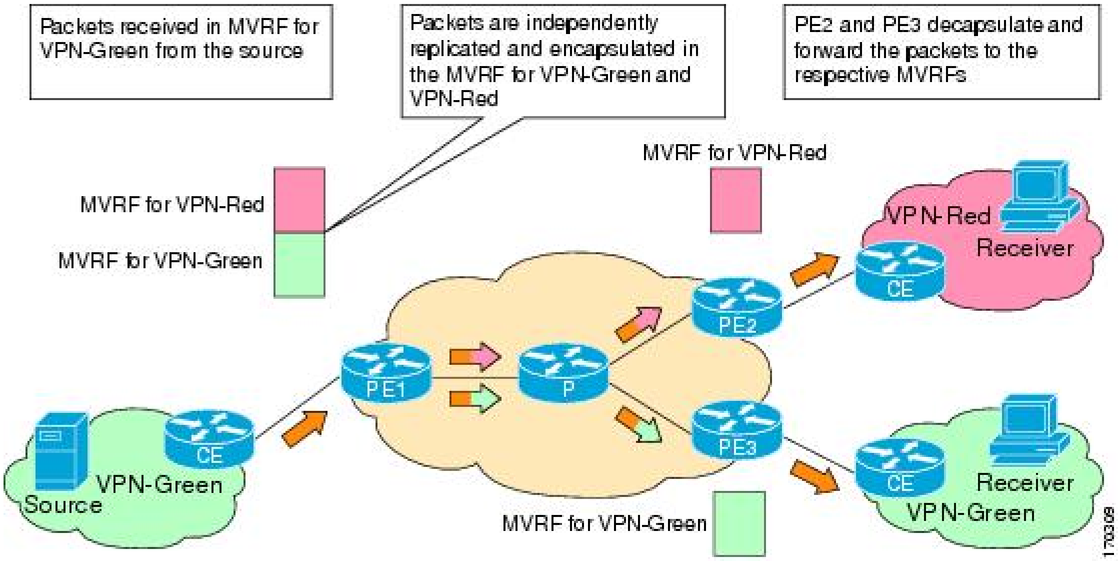
MVPN Extranet Support Configuration Guidelines for Option 1

To provide extranet MVPN services to enterprise VPN customers by configuring the receiver MVRF on the source PE router (Option 1), you would complete the following procedure:

- For each extranet site, you would configure an additional MVRF on the source PE router, that has the same default MDT group as the receiver MVRF, if the MVRF is not configured on the source PE.
- In the receiver MVRF configuration, you would configure the same unicast routing policy on the source and receiver PE routers to import routes from the source MVRF to the receiver MVRF.

The figure illustrates the flow of multicast traffic in an extranet MVPN topology where a receiver MVRF is configured on the source PE router (Option 1). In the topology, an MVRF is configured for VPN-Green and VPN-Red on PE1, the source PE router. A multicast source behind PE1 is sending out a multicast stream to the MVRF for VPN-Green, and there are interested receivers behind PE2 and PE3, the receiver PE routers for VPN-Red and VPN-Green, respectively. After PE1 receives the packets from the source in the MVRF for VPN-Green, it independently replicates and encapsulates the packets in the MVRF for VPN-Green and VPN-Red and forwards the packets. After receiving the packets from this source, PE2 and PE3 decapsulate and forward the packets to the respective MVRFs.

Figure 4 Packet Flow for MVPN Extranet Support Configuration Option 1



Packets received in MVRF for VPN-Green from the source
Packets are independently replicated and encapsulated in the MVRF for VPN-Green and VPN-Red
PE2 and PE3 decapsulate and forward the packets to the respective MVRFs
MVRF for VPN-Red
MVRF for VPN-Green
MVRF for VPN-Red
VPN-Red
Receiver
CE
PE2
PE1
P
PE3
CE
VPN-Green
Source
MVRF for VPN-Green
CE
Receiver
VPN-Green
170309

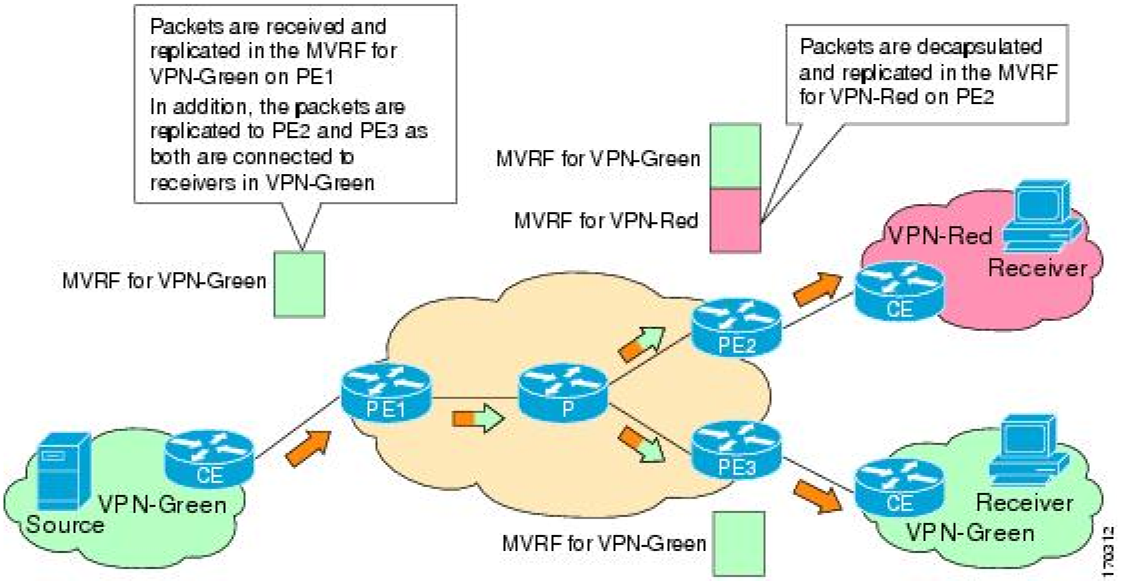
MVPN Extranet Support Configuration Guidelines for Option 2

To provide extranet MVPN services to enterprise VPN customers by configuring a source MVRF on a receiver PE router (Option 2), you would complete the following procedure:

- On a receiver PE router that has one or more interested receivers in a extranet site behind a directly connected CE router, configure an additional MVRF that has the same default MDT group as the site connected to the multicast source, if the MVRF is not configured.
- On the receiver PE router, you would configure the same unicast routing policy to import routes from the source MVRF to the receiver MVRF.

The figure illustrates the flow of multicast traffic in an extranet MVPN topology where the source MVRF is configured on a receiver PE router (Option 2). In the topology, an MVRF is configured for VPN-Green and VPN-Red on PE2, a receiver PE router. A multicast source behind PE1, the source PE router, is sending out a multicast stream to the MVRF for VPN-Green, and there are interested receivers behind PE2, the receiver PE router for VPN-Red, and behind PE3, the receiver PE router for VPN-Green. After PE1 receives the packets from the source in the MVRF for VPN-Green, it replicates and forwards the packets to PE2 and PE3, because both routers are connected to receivers in VPN-Green. The packets that originated from VPN-Green are then replicated on PE2 and forwarded to the interested receivers in VPN-Red and are replicated on PE3 and forwarded to the interested receivers in VPN-Green.

Figure 5 Packet Flow for MVPN Extranet Support Configuration Option 2


Packets are received and replicated in the MVRF for VPN-Green on PE1
In addition, the packets are replicated to PE2 and PE3 as both are connected to receivers in VPN-Green
Packets are decapsulated and replicated in the MVRF for VPN-Red on PE2
MVRF for VPN-Green
MVRF for VPN-Red
MVRF for VPN-Green
VPN-Red
Receiver
CE
PE2
PE1
P
PE3
CE
Source
VPN-Green
CE
Receiver
VPN-Green
MVRF for VPN-Green
170312

RPF for MVPN Extranet Support Using Imported Routes

You must configure either the receiver MVRF on the source PE router (Option 1) or the source MVRF on the receiver PE router (Option 2) for extranet links to be created. Once configured, RPF relies on unicast routing information to determine the interface through which the source is reachable. This interface is used as the RPF interface. No additional configuration is required for RPF resolution. The Multicast VPN Extranet Support feature supports RPF from one VRF to another VRF, from a VRF to the global routing table, and from the global routing table to a VRF.

RPF for MVPN Extranet Support Using Static Mroutes

By default, an extranet MVPN relies on unicast routing policies to determine the RPF interface. When the RPF lookup originates in a receiver MVRF, and it finds that the RPF interface does not lie in the same MVRF, the router uses the information in the Border Gateway Protocol (BGP) imported route to determine the source MVRF. The RPF lookup then continues and resolves in the source MVRF. In cases where the multicast and unicast topologies are incongruent, you can override the default behavior by configuring a static mroute in the receiver MVRF to explicitly specify the source MVRF using the **ip mroute** command with the **fallback-lookup** keyword and **vrf** *vrf-name* keyword and argument.

Static mroutes can also be configured to support RPF for extranet MVPN in the case where the source is present in an MVRF and the receiver is in the global table. In this case, because BGP does not allow VPNv4 routes to be imported into the IPv4 routing table, unicast cannot obtain the source MVRF information needed to resolve the RPF lookup. To enable the RPF lookup to be resolved in this case, a static mroute can be configured to explicitly specify the source MVRF using the **ip mroute** command with the **fallback-lookup** keyword and the **global** keyword.

Multicast VPN Extranet VRF Select

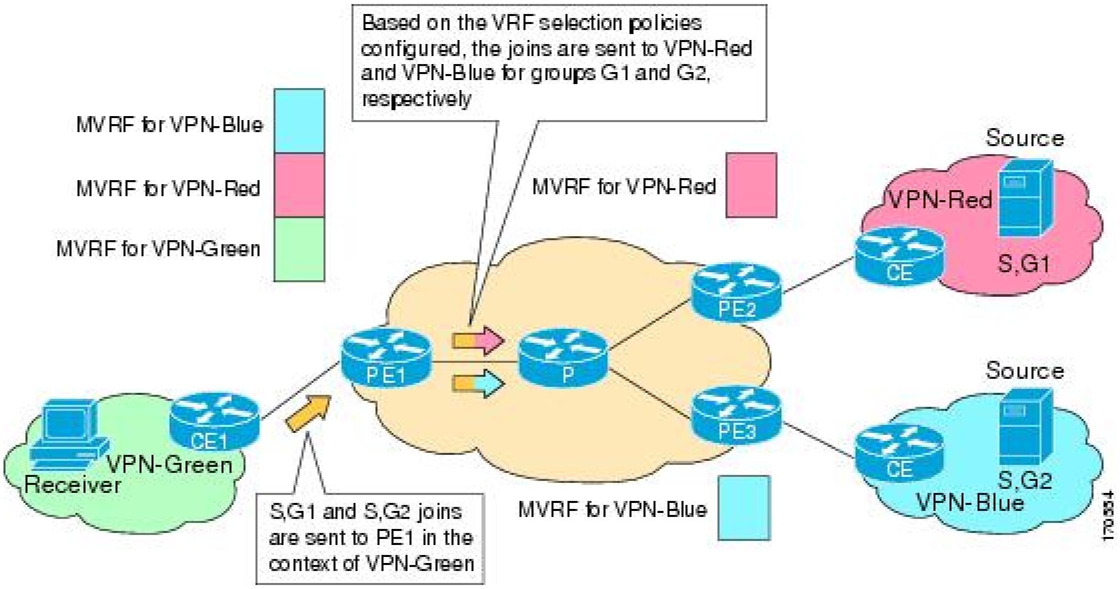
Prior to the introduction of the Multicast VPN Extranet VRF Select feature, RPF lookups for a source address could be performed only in a single VRF, that is, in the VRF where Internet Group Management Protocol (IGMP) or PIM joins are received, in the VRF learned from BGP imported routes, or in the VRF specified in static mroutes (when RPF for an extranet MVPN is configured using static mroutes). In those cases, the source VRF is solely determined by the source address or the way the source address was learned.

The Multicast VPN Extranet VRF Select feature provides the capability for RPF lookups to be performed to the same source address in different VRFs using the group address as the VRF selector. This feature enhances extranet MVPNs by enabling service providers to distribute content streams coming in from different MVPNs and redistributing them from there.

The Multicast VPN VRF Select feature is configured by creating group-based VRF selection policies. Group-based VRF selection policies are configured using the **ip multicast rpf select** command. The **ip multicast rpf select**command is used to configure RPF lookups originating in a receiver MVRF or in the global routing table to be resolved in a source MVRF or in the global routing table based on group address. Access Control Lists (ACLs) are used to define the groups to be applied to group-based VRF selection policies.

The figure illustrates an extranet MVPN topology with the Multicast VPN VRF Select feature configured. In this topology, (S, G1) and (S, G2) PIM joins originating from VPN-Green, the receiver VRF, are forwarded to PE1, the receiver PE. Based on the group-based VRF selection policies configured, PE1 sends the PIM joins to VPN-Red and VPN-Blue for groups G1 and G2, respectively.

Figure 6 RPF Lookups Using Group-Based VRF Selection Policies


Based on the VRF selection policies configured, the joins are sent to VPN-Red and VPN-Blue for groups G1 and G2, respectively
MVRF for VPN-Blue
MVRF for VPN-Red
MVRF for VPN-Green
MVRF for VPN-Red
Source
VPN-Red
CE
S,G1
PE2
PE1
P
Source
PE3
CE1
VPN-Green
Receiver
CE
S,G2
VPN-Blue
S,G1 and S,G2 joins are sent to PE1 in the context of VPN-Green
MVRF for VPN-Blue
170554

How to Configure Multicast VPN Extranet Support

- Configuring MVPN Extranet Support, page 29
- Configuring RPF for MVPN Extranet Support Using Static Mroutes, page 35
- Configuring Group-Based VRF Selection Policies with MVPN Extranet Support, page 36

Configuring MVPN Extranet Support

Perform this task to configure support for extranet MVPN services. Extranet MVPN services enable service providers to distribute IP multicast content originated from a corporate site to the sites of external business partners or suppliers.

Perform one of the following tasks to provide extranet MVPN capabilities:

- Prerequisites, page 29
- Restrictions, page 30
- Configuring the Receiver MVRF on the Source PE - Option 1, page 30
- Configuring the Source MVRF on the Receiver PE - Option 2, page 32

Prerequisites

- This task assumes that you have configured intranet VPN in the source and receiver VPNs.

Restrictions

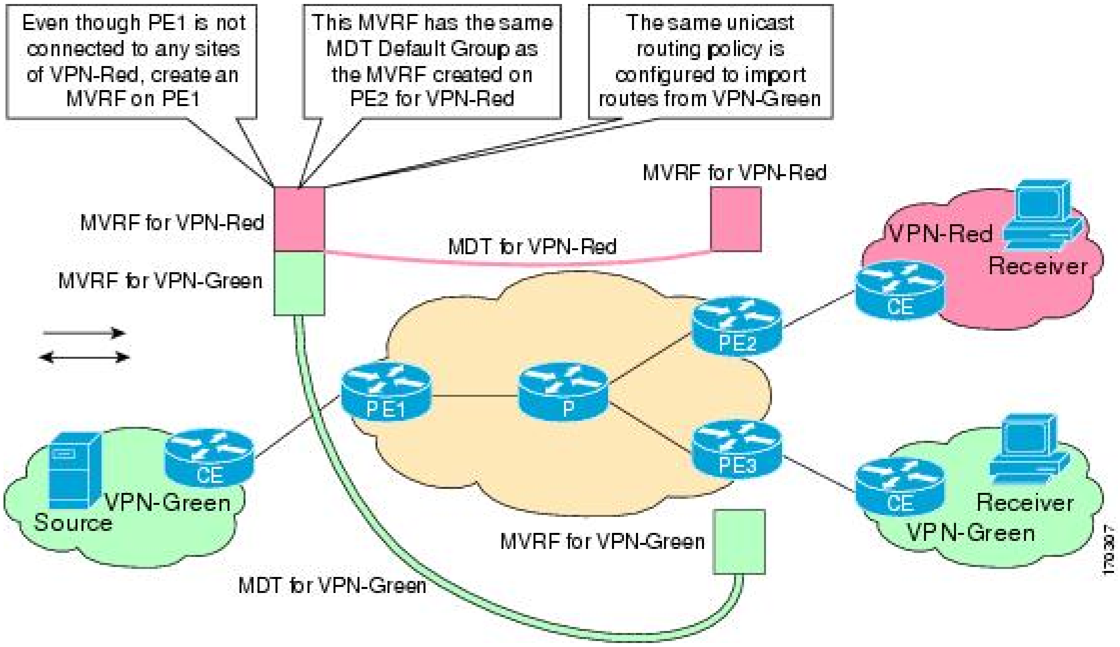
- The Multicast VPN Extranet Support feature supports only PIM-SM and SSM traffic; PIM-DM and bidir-PIM traffic are not supported.
- When extranet MVPNs are configured in a PIM-SM environment, the source and the RP must reside in the same site of the MVPN behind the same PE router.

Configuring the Receiver MVRF on the Source PE - Option 1

Perform this task to provide support for extranet MVPN services by configuring the receiver MVRF on the source PE router (Option 1).

The configuration for this task is done at PE1, the source PE router, in the figure. To provide extranet MVPN services from one enterprise VPN site (for example, VPN-Green) to another enterprise VPN site (for example, VPN-Red) using Option 1, you would configure the receiver MVRF on the source PE router. In the receiver MVRF configuration, the default MDT group must be the same on both the source and receiver PE routers. In addition, you would configure the same unicast routing policy to import routes from the source MVRF (for example, the MVRF for VPN-Green) to the receiver MVRF (for example, the MVRF for VPN-Red).

Figure 7 Topology for MVPN Extranet Support Configuration Option 1


Even though PE1 is not connected to any sites of VPN-Red, create an MVRF on PE1
This MVRF has the same MDT Default Group as the MVRF created on PE2 for VPN-Red
The same unicast routing policy is configured to import routes from VPN-Green
MVRF for VPN-Red
MVRF for VPN-Red
MDT for VPN-Red
MVRF for VPN-Green
VPN-Red
Receiver
CE
PE2
PE1
P
PE3
CE
VPN-Green
Source
CE
Receiver
VPN-Green
MVRF for VPN-Green
MDT for VPN-Green
170307

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target import** *route-target-ext-community*
6. **mdt default** *group-address*
7. **end**
8. **show ip mroute** [**vrf** *vrf-name*] *group-address*

DETAILED STEPS

	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	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf VPN-Red`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 55:2222`	Creates routing and forwarding tables. • Specify the *route-distinguisher* argument to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: ◦ 16-bit autonomous system number: your 32-bit number, for example, 101:3 ◦ 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1

	Command or Action	Purpose
Step 5	**route-target import** *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 55:1111`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities. **Note** For content to be distributed from the source MVRF to the receiver MVRF, you must configure the same unicast routing policy on the source and receiver PE routers to import routes from the source VRF to the receiver VRF.
Step 6	**mdt default** *group-address* **Example:** `Router(config-vrf)# mdt default 232.3.3.3`	Configures the multicast group address range for data MDT groups for a VRF. • A tunnel interface is created as a result of this command. • By default, the destination address of the tunnel header is the *group-address* argument.

Step 7	**end**
	Example:
	`Router(config-vrf)# end`
		Exits VRF configuration mode and returns to privileged EXEC mode.

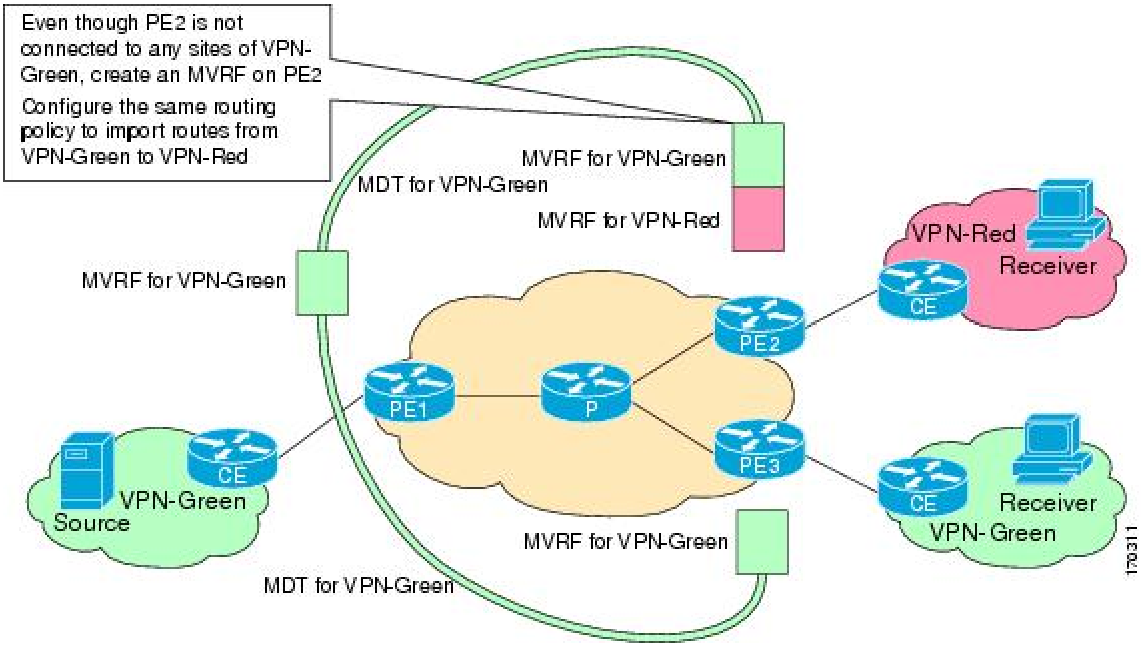
Step 8	**show ip mroute** [**vrf** *vrf-name*] *group-address* **Example:** `Router# show ip mroute 232.3.3.3`	(Optional) Displays the contents of the IP multicast mroute table for a specific group address.

Configuring the Source MVRF on the Receiver PE - Option 2

Perform this task to provide support for extranet MVPN services by configuring the source MVRF on the receiver PE router (Option 2).

The configuration for this task is done at PE2, the receiver PE router, in the figure below. To provide support for extranet MVPN services from one enterprise VPN site (for example, VPN-Green) to another enterprise VPN site (for example, VPN-Red) using Option 2, you would configure the source MVRF on the receiver PE router. The MDT group configuration of the source MVRF must be the same on both the source and receiver PE routers. In addition, you would configure the same unicast routing policy to import routes from the source MVRF (for example, the MVRF for VPN-Green) to the receiver MVRF (for example, the MVRF for VPN-Red).

Figure 8 Topology for MVPN Extranet Support Configuration Option 2



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target import** *route-target-ext-community*
6. **mdt default** *group-address*
7. **end**
8. **show ip mroute** [**vrf** *vrf-name*] *group-address*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf VPN-Red`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 55:1111`	Creates routing and forwarding tables. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: ◦ 16-bit autonomous system (AS) number: your 32-bit number, for example, 101:3 ◦ 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1
Step 5	**route-target import** *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 55:1111`	Creates a route-target extended community for a VRF. • The **import** keyword exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities. **Note** For content to be distributed from the source MVRF to the receiver MVRF, you must configure the same unicast routing policy on the source and receiver PE routers to import routes from the source VRF to the receiver VRF.
Step 6	**mdt default** *group-address* **Example:** `Router(config-vrf)# mdt default 232.1.1.1`	Configures the multicast group address range for data MDT groups for a VRF. • A tunnel interface is created as a result of this command. • By default, the destination address of the tunnel header is the *group-address* argument.

Step 7	**end**
	Example:
	`Router(config-vrf)# end`
		Exits VRF configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show ip mroute** [**vrf** *vrf-name*] *group-address* **Example:** `Router# show ip mroute 232.1.1.1`	(Optional) Displays the contents of the IP multicast mroute table for a specific group address.

Configuring RPF for MVPN Extranet Support Using Static Mroutes

Perform this task to configure RPF for extranet MVPNs using static mroutes.

You must configure support for extranet MVPN services prior to performing this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip mroute vrf** *vrf-name source-address mask* **fallback-lookup** {**global** | **vrf** *vrf-name*} [*distance*]
4. **end**
5. **show ip mroute** [**vrf** *vrf-name*] *group-address*

DETAILED STEPS

	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	**ip mroute vrf** *vrf-name source-address mask* **fallback-lookup** {**global** | **vrf** *vrf-name*} [*distance*] **Example:** `Router(config)# ip mroute vrf VPN-Red 224.100.0.5 255.255.255.255 fallback-lookup vrf VPN-Green`	Configures the RPF lookup originating in a receiver MVRF to continue and be resolved in a source MVRF or in the global routing table using a static mroute. • The **global** keyword is used to specify that the source MVRF is in the global routing table. • The **vrf** keywordand *vrf-name* argument are used to explicitly specify a VRF as the source MVRF.

	Command or Action	Purpose

Step 4	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Step 5	**show ip mroute** [**vrf** *vrf-name*] *group-address* **Example:** `Router# show ip mroute 224.100.0.5`	(Optional) Displays the contents of the IP multicast mroute table for a specific group address.

Configuring Group-Based VRF Selection Policies with MVPN Extranet Support

Perform this task to configure group-based VRF selection policies with MVPN.

This task enables RPF lookups to be performed to the same source address in different VRFs using the group address as the VRF selector. This feature enhances extranet MVPNs by enabling service providers to distribute content streams coming in from different MVPNs and redistributing them from there.

- This task requires the routers to be running Cisco IOS XE Release 2.5 or a later release.
- You must configure support for extranet MVPN services prior to performing this task.
- ACLs are used to define the groups to be applied to group-based VRF selection policies. This task assumes that you have configured the ACLs to be applied to group-based VRF selection policies.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast** [**vrf** receiver-*vrf-name*] **rpf select** {**global** | **vrf** *source-vrf-name*} **group-list** *access-list*
4. Repeat Step 3 to create additional group-based VRF selection policies.
5. **end**
6. **show ip rpf** [**vrf** *vrf-name*] **select**
7. **show ip rpf** [**vrf** *vrf-name*] *source-address* [*group-address*]

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast** [**vrf** receiver-*vrf-name*] **rpf select** {**global** | **vrf** *source-vrf-name*} **group-list** *access-list* **Example:** `Router(config)# ip multicast vrf VPN-Green rpf select vrf VPN-Red group-list 1`	Configures RPF lookups originating in a receiver MVRF or in the global routing table to be resolved in a source MVRF or in the global routing table based on group address. • The optional **vrf** keyword and *receiver-vrf-name* argument are used to apply a group-based VRF selection policy to RPF lookups originating in the VRF specified for the *receiver-vrf-name* argument. If the optional **vrf** keyword and *receiver-vrf-name* argument are not specified, the group-based VRF selection policy applies to RPF lookups originating in the global table. • The **global** keyword is used to specify that the RPF lookup for groups matching the access list specified for the **group-list** keyword and *access-list* argument be performed in the global routing table. • The **vrf** keyword and *source-vrf-name* argument are used to specify that the RPF lookups for groups matching the access list specified with the **group-list** keyword and *access-list* argument be performed in the VRF specified for the *vrf-name* argument. • The **group-list** keyword and *access-list* argument are used to specify the access list to be applied to the group-based VRF selection policy.
Step 4	Repeat Step 3 to create additional group-based VRF selection policies.	--

Step 5	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Step 6	**show ip rpf** [**vrf** *vrf-name*] **select** **Example:** `Router# show ip rpf select`	Displays group-to-VRF mapping information. • Use the optional **vrf** keyword and *vrf-name* argument to display the group-to-VRF mappings for the VRF instance specified for the *vrf-name* argument.

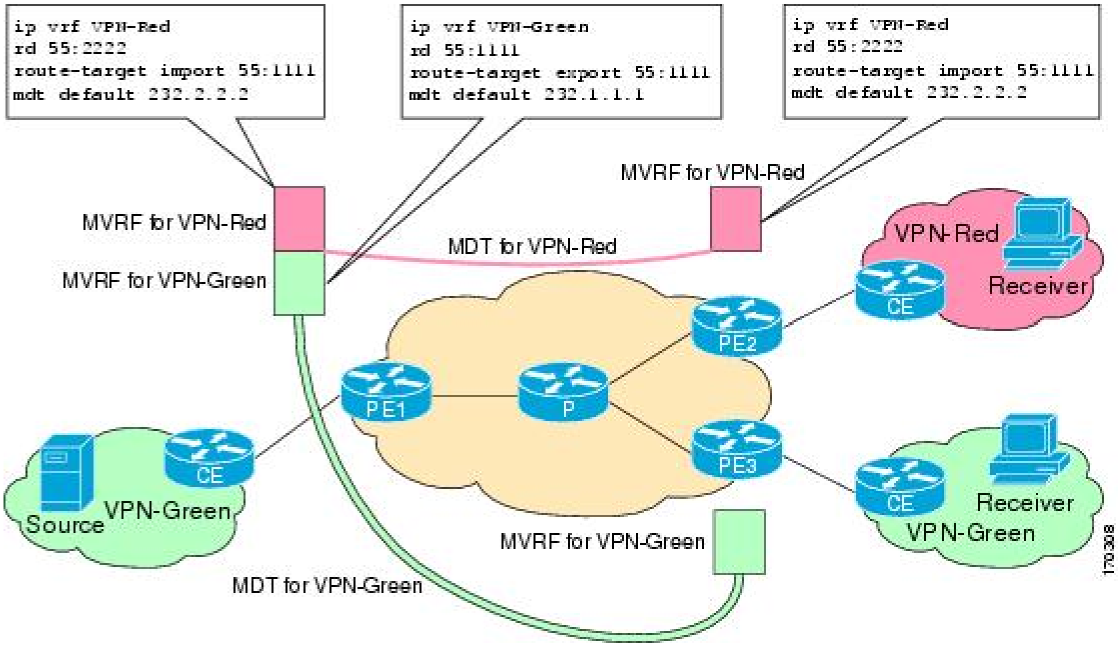
	Command or Action	Purpose
Step 7	**show ip rpf** [**vrf** *vrf-name*] *source-address* [*group-address*] **Example:** `Router# show ip rpf 172.16.10.13`	Displays information about how IP multicast routing does RPF. • Use this command after configuring group-based VRF selection policies to confirm that RPF lookups are being performed based on the group address and to display the VRF where the RPF lookup is being performed. • Use the optional **vrf** keyword and *vrf-name* argument to display how IP multicast routing does RPF in the VRF specified for the *vrf-name*argument.

Configuration Examples for Multicast VPN Extranet Support

Example Configuring the Receiver VRF on the Source PE Router - Option 1

The following example shows the configurations for PE1, the source PE router, and PE2, the receiver PE router, in the figure. In this example, extranet MVPN services are supported between VPN-Green and VPN-Red by configuring the receiver MVRF for VPN-Red on PE1, the source PE router. The MVRF configuration for VPN-Red is configured to import routes from the MVRF for VPN-Green to the MVRF for VPN-Red.

Figure 9 Topology for MVPN Extranet Support Option 1 Configuration Example



```
ip vrf VPN-Red
rd 55:2222
route-target import 55:1111
mdt default 232.2.2.2
```

```
ip vrf VPN-Green
rd 55:1111
route-target export 55:1111
mdt default 232.1.1.1
```

```
ip vrf VPN-Red
rd 55:2222
route-target import 55:1111
mdt default 232.2.2.2
```

PE1 Configuration

```
ip cef
!
ip vrf VPN-Green
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip vrf VPN-Red
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 route-target import 55:1111
 mdt default 232.3.3.3
!
ip multicast-routing
ip multicast-routing vrf VPN-Green
ip multicast-routing vrf VPN-Red
!
interface Loopback0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.0.2 remote-as 55
 neighbor 10.2.0.2 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 10.2.0.2 activate
 neighbor 10.2.0.2 send-community extended
 !
 address-family vpnv4
 neighbor 10.2.0.2 activate
 neighbor 10.2.0.2 send-community extended
 !
```

PE2 Configuration

```
!
ip vrf VPN-Red
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 route-target import 55:1111
 mdt default 232.3.3.3
!
ip multicast-routing
ip multicast-routing vrf VPN-Red
!
interface Loopback0
 ip address 10.2.0.2 255.255.255.0
 ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.0.1 remote-as 55
 neighbor 10.1.0.1 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 send-community extended
 !
 address-family vpnv4
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 send-community extended
 !
```

States in the Global Table on PE1 and PE2 for the MDT Default Group 232.3.3.3

The following are sample outputs from the **show ip mroute** command on PE1 and PE2. The sample outputs show the global table for the MDT default group 232.3.3.3 on PE1 and PE2.

```
PE1# show ip mroute 232.3.3.3
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
(10.1.0.1, 232.3.3.3), 00:46:27/00:03:27, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet0/0, Forward/Sparse-Dense, 00:45:17/00:02:44
(10.2.0.2, 232.3.3.3), 00:45:17/00:02:57, flags: sTIZ
  Incoming interface: GigabitEthernet0/0, RPF nbr 224.0.1.4
  Outgoing interface list:
    MVRF VPN-Red, Forward/Sparse-Dense, 00:45:17/00:01:09
PE2# show ip mroute 232.3.3.3
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
(10.1.0.1, 232.3.3.3), 00:45:08/00:02:37, flags: sTIZ
  Incoming interface: GigabitEthernet1/0, RPF nbr 224.0.2.4
  Outgoing interface list:
    MVRF VPN-Red, Forward/Sparse-Dense, 00:45:08/00:01:27
(10.2.0.2, 232.3.3.3), 00:46:19/00:03:07, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse-Dense, 00:45:08/00:02:49
```

States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The output indicates that extranet receivers in VPN-Red are receiving content from a source in VPN-Green that is sending to multicast group 228.8.8.8. The “E” flag in the output indicates that a (*, G) or (S, G) entry in the VRF routing table is a source VRF entry and has extranet receiver MVRF mroute entries linked to it.

```
PE1# show ip mroute vrf VPN-Green 228.8.8.8
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, E - Extranet,
       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
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, flags: SE
  Incoming interface: GigabitEthernet3/0, RPF nbr 10.1.1.5
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:00:05/00:02:54, flags: TE
  Incoming interface: GigabitEthernet3/0, RPF nbr 10.1.1.5
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
  (10.1.1.200, 228.8.8.8), 00:00:05/stopped, OIF count: 1, flags:
```

States in the VRF Table for VPN-Red on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Red on PE1 when receivers join the multicast group 228.8.8.8. The “using vrf VPN-Green” field indicates that VPN-Red is using unicast routing information from VPN-Green to determine the RPF interface through which the source is reachable.

```
PE1# show ip mroute vrf VPN-Red 228.8.8.8
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, E - Extranet,
       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
(*, 228.8.8.8), 00:01:45/stopped, RP 10.100.0.5, flags: S
  Incoming interface: GigabitEthernet3/0, RPF nbr 10.1.1.5, using vrf VPN-Green
  Outgoing interface list:
    Tunnel2, Forward/Sparse-Dense, 00:01:45/00:02:49
(10.1.1.200, 228.8.8.8), 00:00:12/00:03:27, flags:
  Incoming interface: GigabitEthernet3/0, RPF nbr 10.1.1.5, using vrf VPN-Green
  Outgoing interface list:
    Tunnel2, Forward/Sparse-Dense, 00:00:12/00:03:18
```

States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE2. The sample output shows the VRF table for VPN-Red on PE2 when receivers join the multicast group 228.8.8.8.

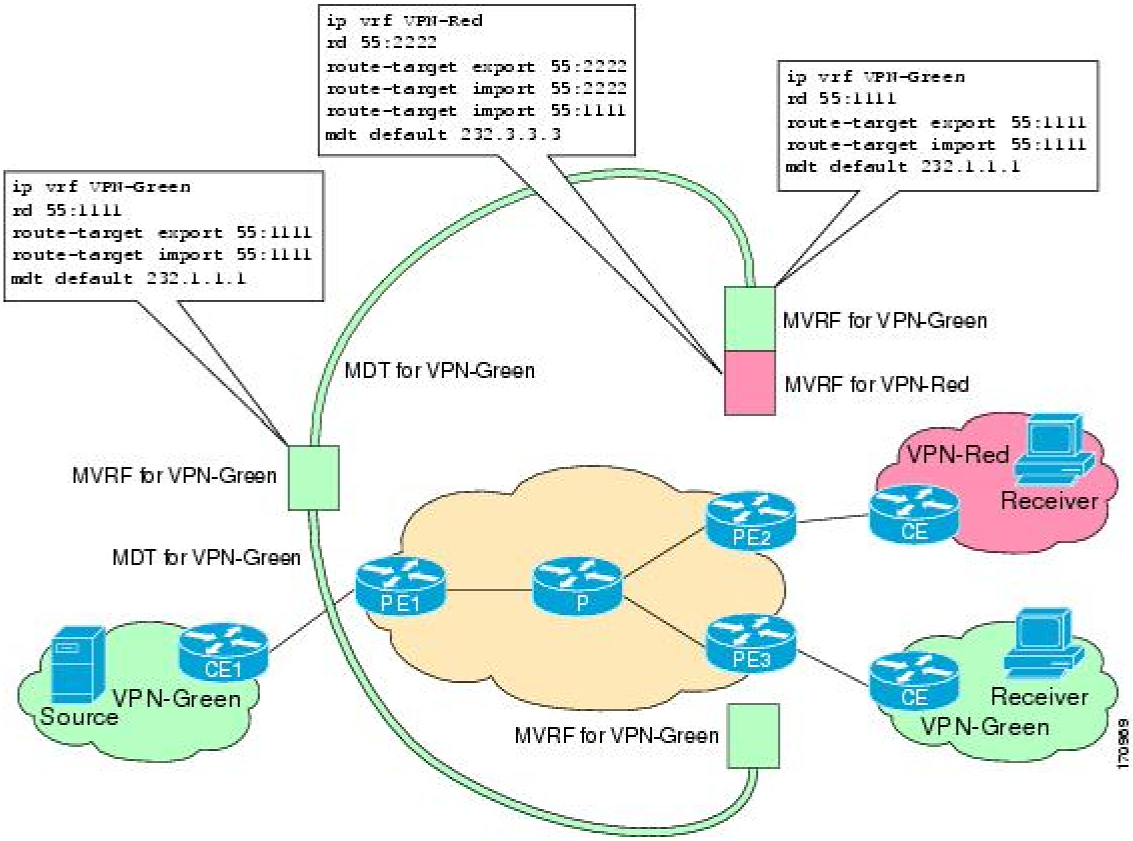
```
PE2# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - 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
(*, 228.8.8.8), 00:00:28/stopped, RP 10.100.0.5, flags: S
```

```
  Incoming interface: Tunnel1, RPF nbr 10.1.0.1
  Outgoing interface list:
    GigabitEthernet9/0, Forward/Sparse-Dense, 00:00:28/00:03:02
(10.1.1.200, 228.8.8.8), 00:00:00/00:03:29, flags:
  Incoming interface: Tunnel1, RPF nbr 10.1.0.1
  Outgoing interface list:
    GigabitEthernet9/0, Forward/Sparse-Dense, 00:00:00/00:03:29
```

Example Configuring the Source VRF on the Receiver PE - Option 2

The following configuration example is based on the extranet MVPN topology illustrated in the figure. This example shows the configurations for PE2, the receiver PE router, and PE1, the source PE router. In this example, extranet MVPN services are supported between VPN-Green and VPN-Red by configuring the source MVRF for VPN-Green on PE2. The same unicast routing policy is configured to import routes from VPN-Green to VPN-Red.

Figure 10 Topology for MVPN Extranet Support Option 2 Configuration Example



```
ip vrf VPN-Red
rd 55:2222
route-target export 55:2222
route-target import 55:2222
route-target import 55:1111
mdt default 232.3.3.3
```

```
ip vrf VPN-Green
rd 55:1111
route-target export 55:1111
route-target import 55:1111
mdt default 232.1.1.1
```

```
ip vrf VPN-Green
rd 55:1111
route-target export 55:1111
route-target import 55:1111
mdt default 232.1.1.1
```

PE2 Configuration

```
ip cef
!
ip vrf VPN-Red
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 route-target import 55:1111
 mdt default 232.3.3.3
!
```

```
ip vrf VPN-Green
 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 VPN-Red
ip multicast-routing vrf VPN-Green
!
interface Loopback0
 ip address 10.2.0.2 255.255.255.0
 ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.0.1 remote-as 55
 neighbor 10.1.0.1 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 send-community extended
 !
 address-family vpnv4
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 send-community extended
 !
```

PE1 Configuration

```
ip cef
!
ip vrf VPN-Green
 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 VPN-Green
!
interface Loopback0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.0.2 remote-as 55
 neighbor 10.2.0.2 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 10.2.0.2 activate
 neighbor 10.2.0.2 send-community extended
 !
 address-family vpnv4
 neighbor 10.2.0.2 activate
 neighbor 10.2.0.2 send-community extended
 !
```

States in the Global Table on PE1 and PE2 for the MDT Default Group 232.1.1.1

The following are sample outputs from the **show ip mroute** command on PE1 and PE2. The sample output shows the global table for the MDT default group 232.1.1.1 on PE1 and PE2.

```
PE1# 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, E - Extranet,
       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
(10.2.0.2, 232.1.1.1), 00:01:19/00:02:42, flags: sTIZ
  Incoming interface: GigabitEthernet0/0, RPF nbr 10.0.1.4
  Outgoing interface list:
    MVRF VPN-Green, Forward/Sparse-Dense, 00:01:19/00:02:07
(10.1.0.1, 232.1.1.1), 00:02:19/00:03:11, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet0/0, Forward/Sparse-Dense, 00:02:00/00:02:36
PE2# 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, E - Extranet,
       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
(10.1.0.1, 232.1.1.1), 00:02:04/00:02:38, flags: sTIZ
  Incoming interface: GigabitEthernet1/0, RPF nbr 10.0.2.4
  Outgoing interface list:
    MVRF VPN-Green, Forward/Sparse-Dense, 00:02:04/00:02:09
(10.2.0.2, 232.1.1.1), 00:02:04/00:03:09, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse-Dense, 00:01:22/00:03:09
```

States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8.

```
PE1# show ip mroute vrf VPN-Green 228.8.8.8
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, E - Extranet,
       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
(*, 228.8.8.8), 00:01:43/00:02:52, RP 10.100.0.5, flags: S
  Incoming interface: GigabitEthernet3/0, RPF nbr 10.1.1.5
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:01:43/00:02:52
```

```
(10.1.1.200, 228.8.8.8), 00:01:15/00:03:26, flags: T
  Incoming interface: GigabitEthernet3/0, RPF nbr 10.1.1.5
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:01:15/00:03:19
```

States in the VRF Table for VPN-Green on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE2. The output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The output indicates that extranet receivers in VPN-Red are receiving content from the source in VPN-Green that is sending to multicast group 228.8.8.8. The “E” flag indicates that a (*, G) or (S, G) entry in the VRF routing table is a source VRF entry and has extranet receiver MVRF mroute entries linked to it.

```
PE2# show ip mroute vrf VPN-Green 228.8.8.8
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, E - Extranet,
       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
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, flags: SE
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:01:31/00:02:59, flags: TE
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
  (10.1.1.200, 228.8.8.8), 00:01:31/00:03:29, OIF count: 1, flags:
```

States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE2. The sample output shows the state of the VRF table for VPN-Red on PE2 when receivers join the multicast group 228.8.8.8. The “using vrf VPN-Green” field indicates that VPN-Red is using unicast routing information from VPN-Green to determine the RPF interface through which the source is reachable.

```
PE2# show ip mroute vrf VPN-Red 228.8.8.8

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, E - Extranet,
       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
(*, 228.8.8.8), 00:02:00/stopped, RP 10.100.0.5, flags: S
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green
  Outgoing interface list:
    GigabitEthernet9/0, Forward/Sparse-Dense, 00:02:00/00:02:34
(10.1.1.200, 228.8.8.8), 00:01:32/00:03:28, flags:
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green
  Outgoing interface list:
    GigabitEthernet9/0, Forward/Sparse-Dense, 00:01:32/00:03:01
```

Example Configuring RPF for MVPN Extranet Support Using Static Mroutes

The following example shows how to configure the RPF lookup originating in VPN-Red to be resolved in VPN-Green using the static mroute 192.168.1.1:

```
ip mroute vrf VPN-Red 192.168.1.1 255.255.255.255 fallback-lookup vrf VPN-Green
```

Example Configuring Group-Based VRF Selection Policies with MVPN Extranet Support

The following example shows how to use group-based VRF selection policies to configure RPF lookups originating in VPN-Green to be performed in VPN-Red for group addresses that match ACL 1 and to be performed in VPN-Blue for group addresses that match ACL 2.

```
ip multicast vrf VPN-Green rpf select vrf VPN-Red group-list 1
ip multicast vrf VPN-Green rpf select vrf VPN-Blue group-list 2
!
.
.
.
!
access-list 1 permit 239.0.0.0 0.255.255.255
access-list 2 permit 238.0.0.0 0.255.255.255
!
```

Additional References

Related Documents

Related Topic	Document Title
Basic IP multicast concepts, configuration tasks, and examples	“ Configuring Basic IP Multicast ” module
IP multicast overview	“ IP Multicast Technology Overview ” module
Multicast VPN concepts, configuration tasks, and examples	“ Configuring Multicast VPN ” module
MPLS VPN concepts, configuration tasks, and examples	“ MPLS Virtual Private Networks ” module
IP Access List concepts, configuration tasks, and examples	“ IP Access List Overview ” module
IP multicast commands: complete command syntax, command mode, command history, command defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Related Topic	Document Title
MPLS commands: complete command syntax, command mode, command history, command defaults, usage guidelines, and examples	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

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

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

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 Extranet Support

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

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

Table 3 Feature Information for Configuring Multicast VPN Extranet Support

Feature Name	Releases	Feature Information
Multicast VPN Extranet Support	Cisco IOS XE Release 2.5	The Multicast VPN Extranet Support feature enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers. The following commands were introduced or modified by this feature: **ip mroute**, **show ip mroute**.
Multicast VPN Extranet VRF Select	Cisco IOS XE Release 2.5	The Multicast VPN Extranet VRF Select feature provides the capability for RPF lookups to be performed to the same source address in different VRFs using the group address as the VRF selector. This feature enhances extranet MVPNs by enabling service providers to distribute content streams coming in from different MVPNs and redistributing them from there. The following commands were introduced or modified by this feature: **ip multicast rpf select**, **show ip rpf**, **show ip rpf select**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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

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

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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

Prerequisites for 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, page 54
- MVPN Inter-AS Support for Option A, page 54

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

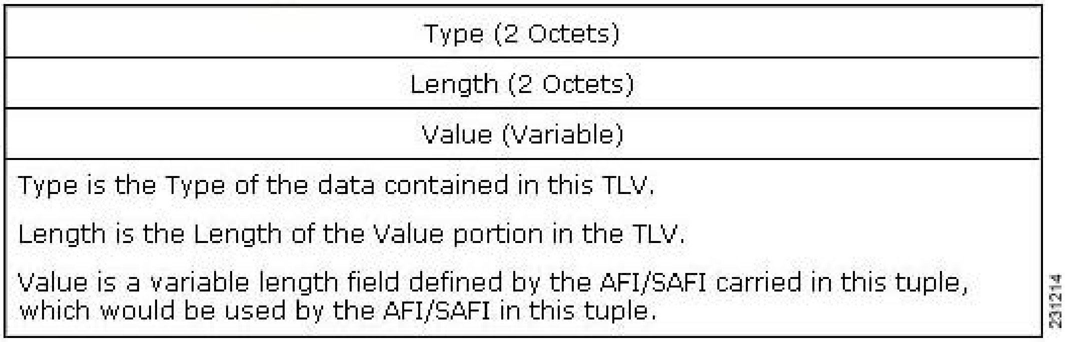
In an adjacent autonomous system, a PE router that wants to join a particular source of the default MDT for a given MVPN must know the originator's address of the source PE router. This presents some challenges for Option B inter-AS deployments because the originator next hop for VPNv4 routes is rewritten 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, page 57 section.

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

Figure 11 ***BGP Connector Attribute***



Type (2 Octets)

Length (2 Octets)

Value (Variable)

Type is the Type of the data contained in this TLV.

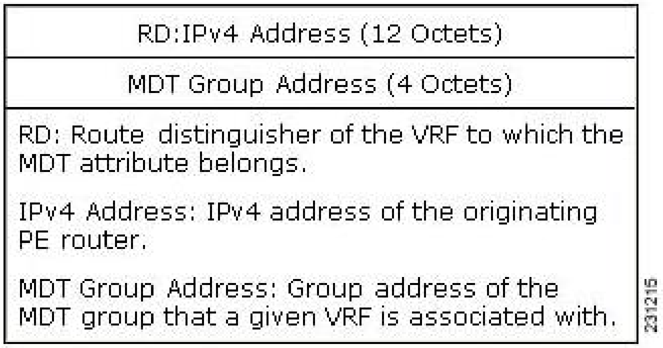
Length is the Length of the Value portion in the TLV.

Value is a variable length field defined by the AFI/SAFI carried in this tuple, which would be used by the AFI/SAFI in this tuple.

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 12 ***MDT SAFI Format***



RD:IPv4 Address (12 Octets)

MDT Group Address (4 Octets)

RD: Route distinguisher of the VRF to which the MDT attribute belongs.

IPv4 Address: IPv4 address of the originating PE router.

MDT Group Address: Group address of the MDT group that a given VRF is associated with.

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.

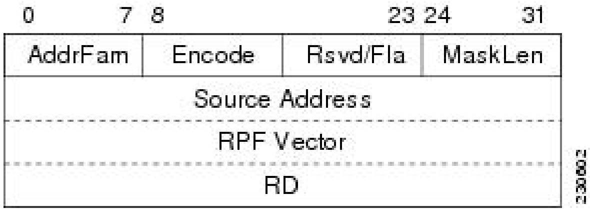
PIM RPF Vector

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 13 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, page 57
- Recipients of an RPF Vector, page 58
- ASBR Receipt of an RPF Vector, page 58
- Interoperability with RPF Vector, page 58

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, page 58
- Guidelines for Configuring MDT Address Family Sessions on PE Routers for MVPN Inter-AS Support, page 59

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

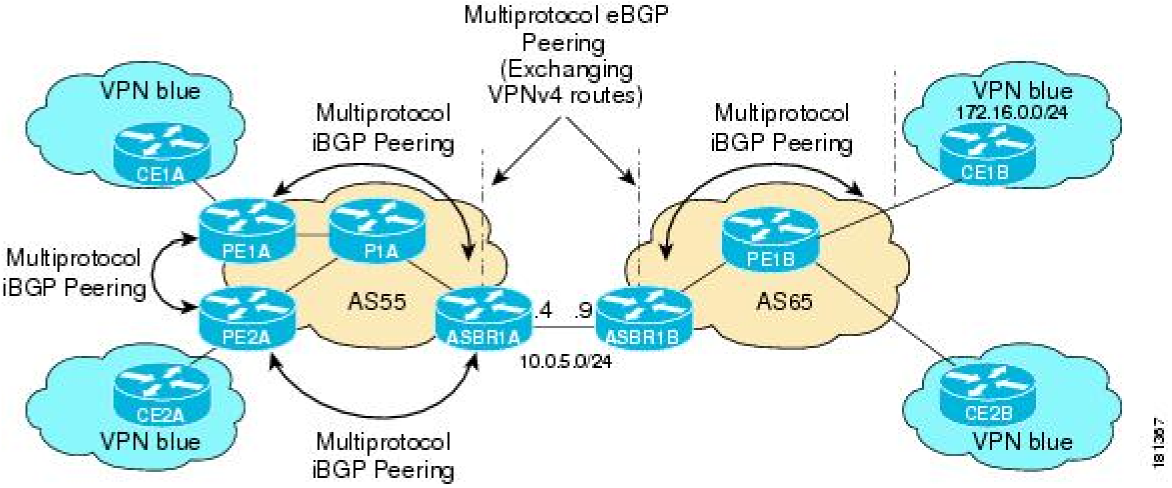
When configuring routers for MVPN inter-AS support, follow these guidelines:

- 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.

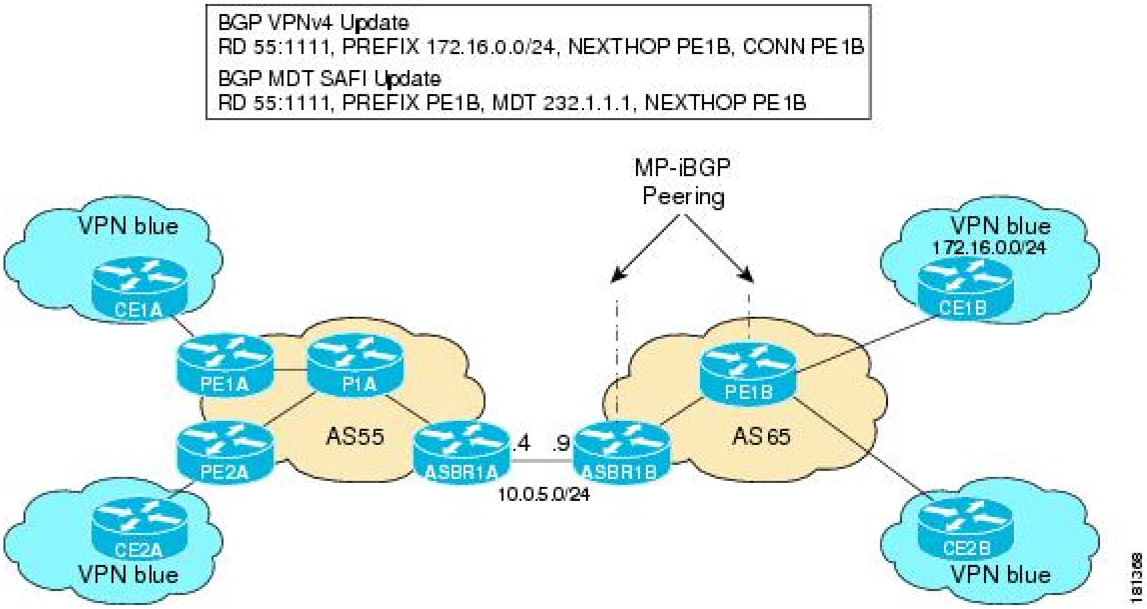
Figure 14 MVPN Inter-AS Support Option B Sample Topology


Multiprotocol eBGP Peering (Exchanging VPNv4 routes)
VPN blue
CE1A
Multiprotocol iBGP Peering
Multiprotocol iBGP Peering
PE1A
P1A
Multiprotocol iBGP Peering
PE2A
AS55
ASBR1A
.4
.9
ASBR1B
10.0.5.0/24
CE2A
VPN blue
Multiprotocol iBGP Peering
VPN blue
172.16.0.0/24
CE1B
PE1B
AS65
CE2B
VPN blue
181367

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.

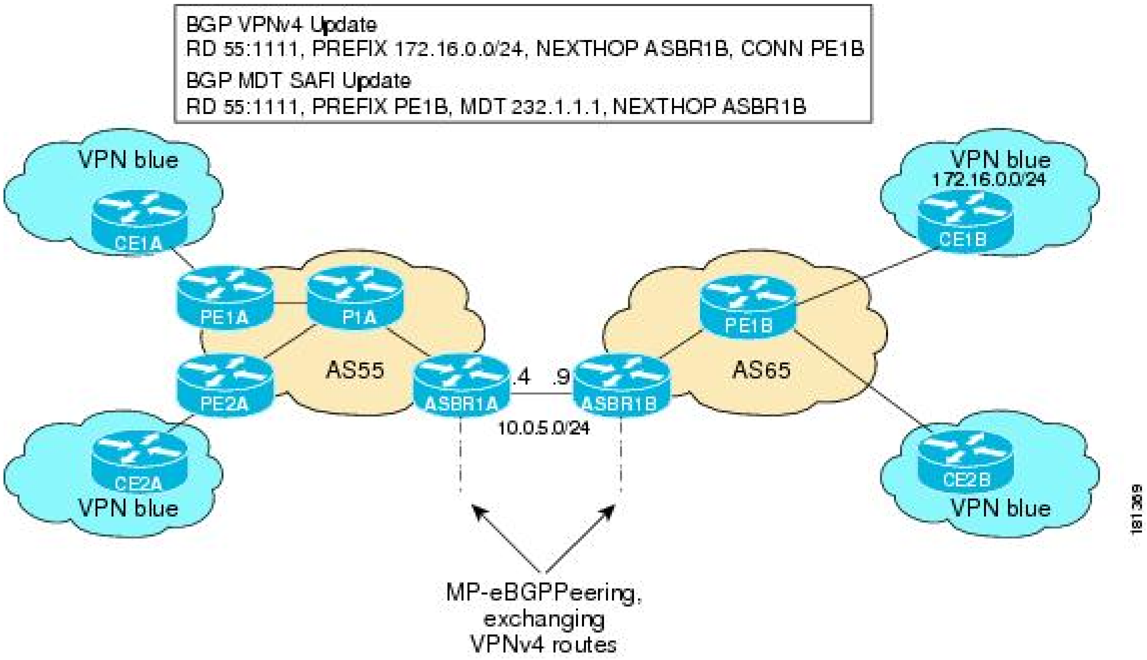
Figure 15 ***BGP Updates from PE1B to ASBR1B***



BGP VPNv4 Update
RD 55:1111, PREFIX 172.16.0.0/24, NEXTHOP PE1B, CONN PE1B
BGP MDT SAFI Update
RD 55:1111, PREFIX PE1B, MDT 232.1.1.1, NEXTHOP PE1B

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.

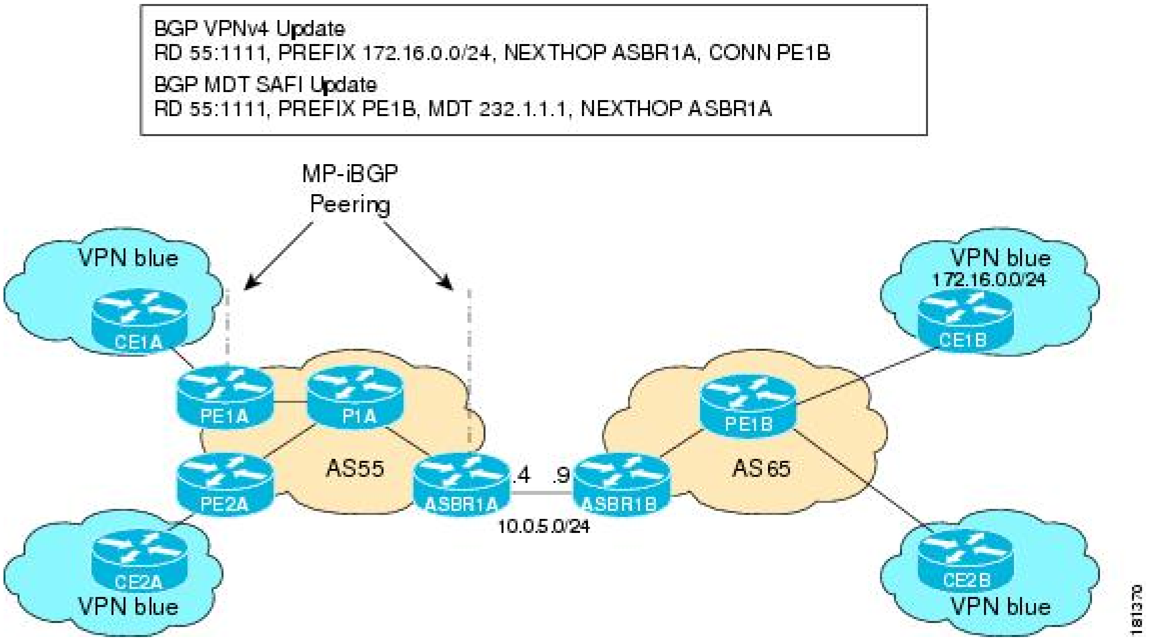
Figure 16 ***BGP Updates from ASBR1B to ASBR1A***



BGP VPNv4 Update
RD 55:1111, PREFIX 172.16.0.0/24, NEXTHOP ASBR1B, CONN PE1B
BGP MDT SAFI Update
RD 55:1111, PREFIX PE1B, MDT 232.1.1.1, NEXTHOP ASBR1B

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

Figure 17 ***BGP Updates from ASBR1A to PE1A***

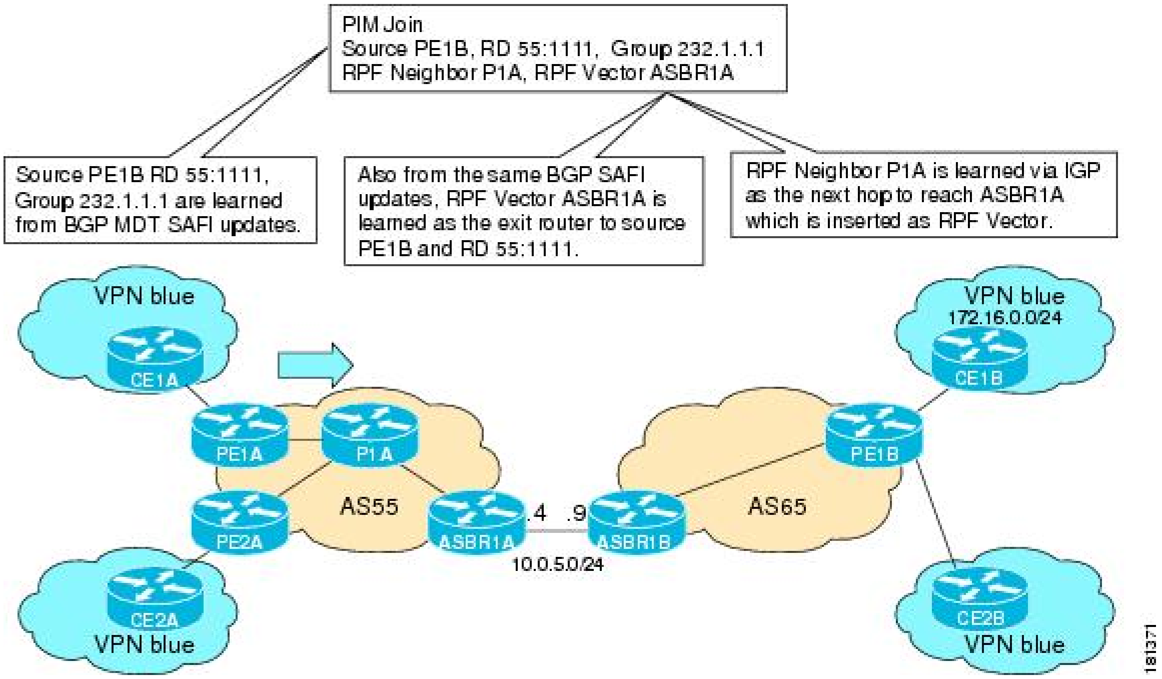


BGP VPNv4 Update
RD 55:1111, PREFIX 172.16.0.0/24, NEXTHOP ASBR1A, CONN PE1B

BGP MDT SAFI Update
RD 55:1111, PREFIX PE1B, MDT 232.1.1.1, NEXTHOP ASBR1A

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 18 ***SSM Default PIM Join from PE1A to P1A***



PIM Join
Source PE1B, RD 55:1111, Group 232.1.1.1
RPF Neighbor P1A, RPF Vector ASBR1A

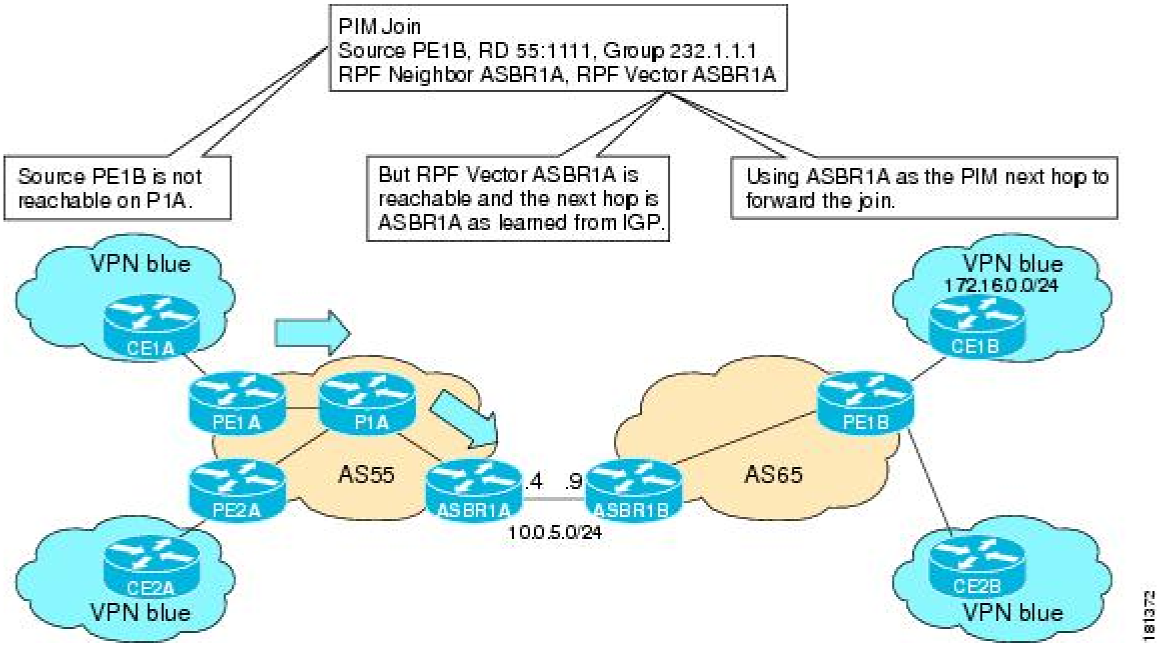
Source PE1B RD 55:1111, Group 232.1.1.1 are learned from BGP MDT SAFI updates.

Also from the same BGP SAFI updates, RPF Vector ASBR1A is learned as the exit router to source PE1B and RD 55:1111.

RPF Neighbor P1A is learned via IGP as the next hop to reach ASBR1A which is inserted as RPF Vector.

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.

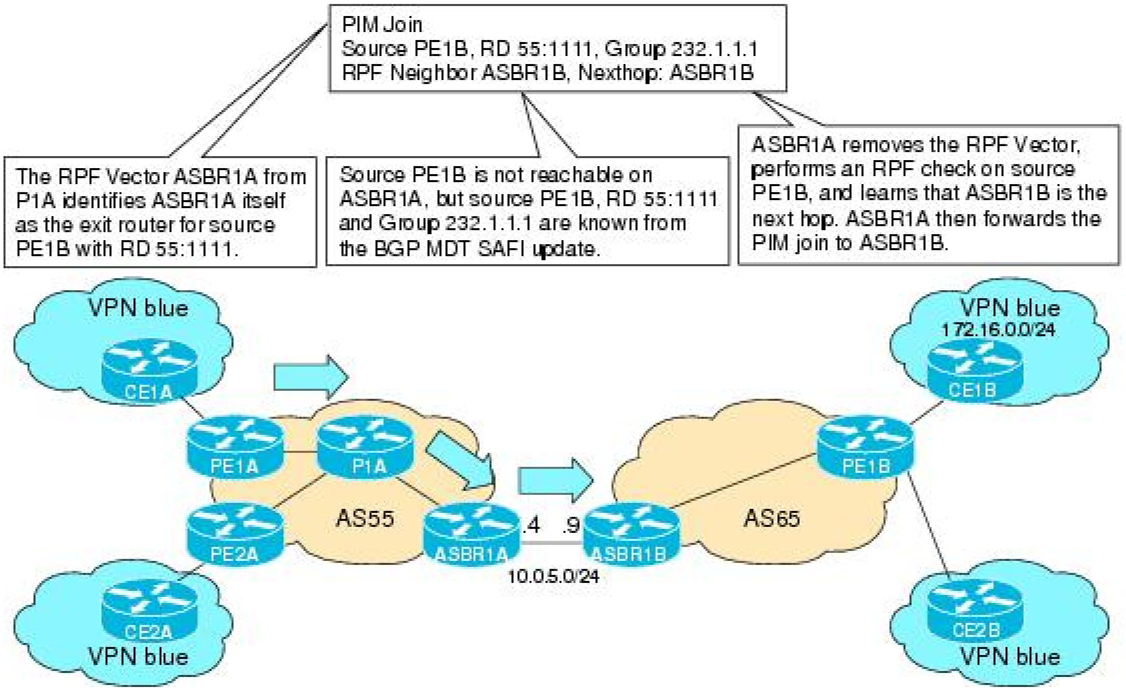
Figure 19 SSM Default MDT PIM Join from P1A to ASBR1A


PIM Join
Source PE1B, RD 55:1111, Group 232.1.1.1
RPF Neighbor ASBR1A, RPF Vector ASBR1A
Source PE1B is not reachable on P1A.
But RPF Vector ASBR1A is reachable and the next hop is ASBR1A as learned from IGP.
Using ASBR1A as the PIM next hop to forward the join.
VPN blue
CE1A
PE1 A
P1A
PE2A
AS55
ASBR1A
.4
.9
ASBR1B
10.0.5.0/24
AS65
VPN blue
172.16.0.0/24
CE1B
PE1B
CE2A
VPN blue
CE2B
VPN blue
181372

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 20 ***SSM Default MDT PIM Join from ASBR1A to ASBR1B***



PIM Join
Source PE1B, RD 55:1111, Group 232.1.1.1
RPF Neighbor ASBR1B, Nexthop: ASBR1B

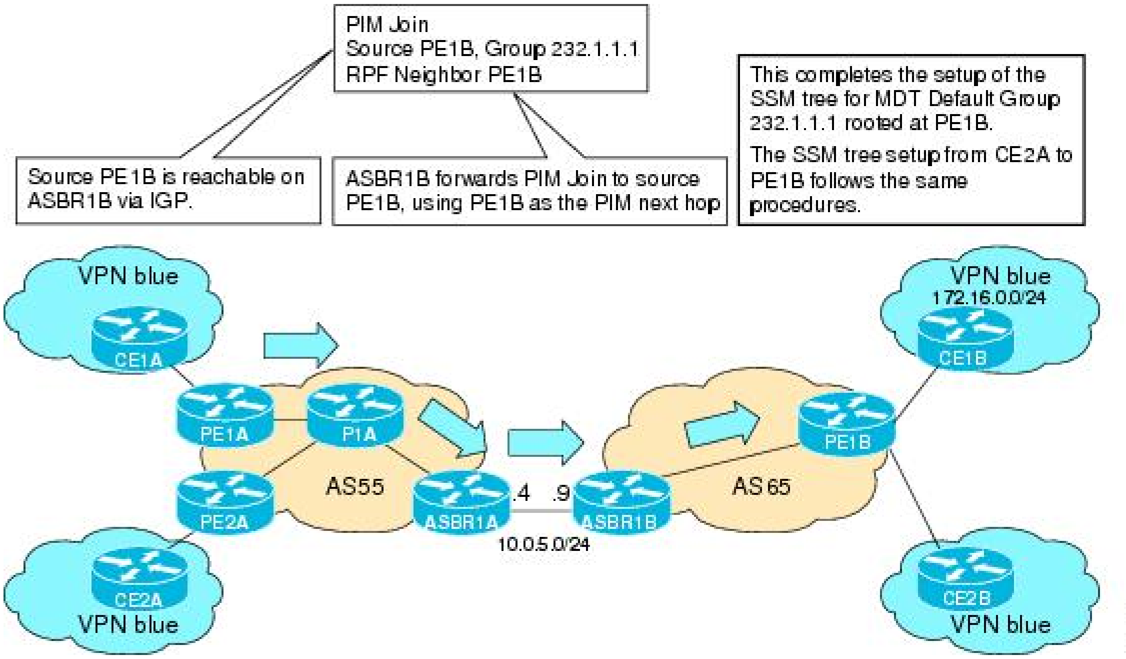
The RPF Vector ASBR1A from P1A identifies ASBR1A itself as the exit router for source PE1B with RD 55:1111.

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 update.

ASBR1A removes the RPF Vector, performs an RPF check on source PE1B, and learns that ASBR1B is the next hop. ASBR1A then forwards the PIM join 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 21 ***SSM Default MDT PIM Join from ASBR1B to PE1B***



PIM Join
Source PE1B, Group 232.1.1.1
RPF Neighbor PE1B

Source PE1B is reachable on ASBR1B via IGP.

ASBR1B forwards PIM Join to source PE1B, using PE1B as the PIM next hop

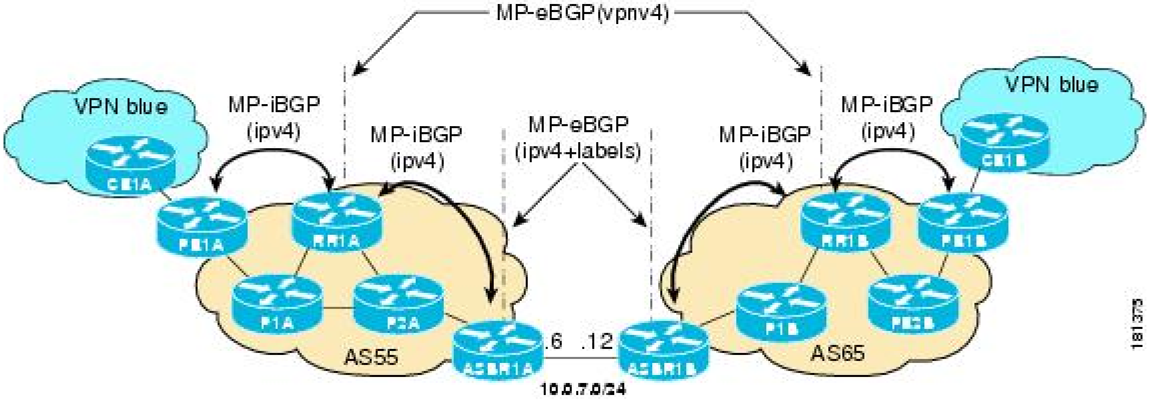
This completes the setup of the SSM tree for MDT Default Group 232.1.1.1 rooted at PE1B.

The SSM tree setup from CE2A to PE1B follows the same procedures.

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.

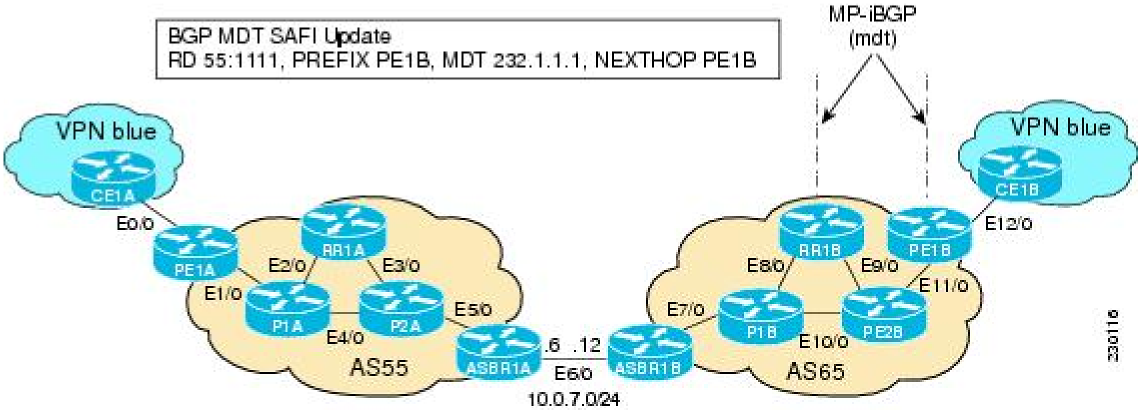
Figure 22 *MVPN Inter-AS Support Option C Sample Topology*



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

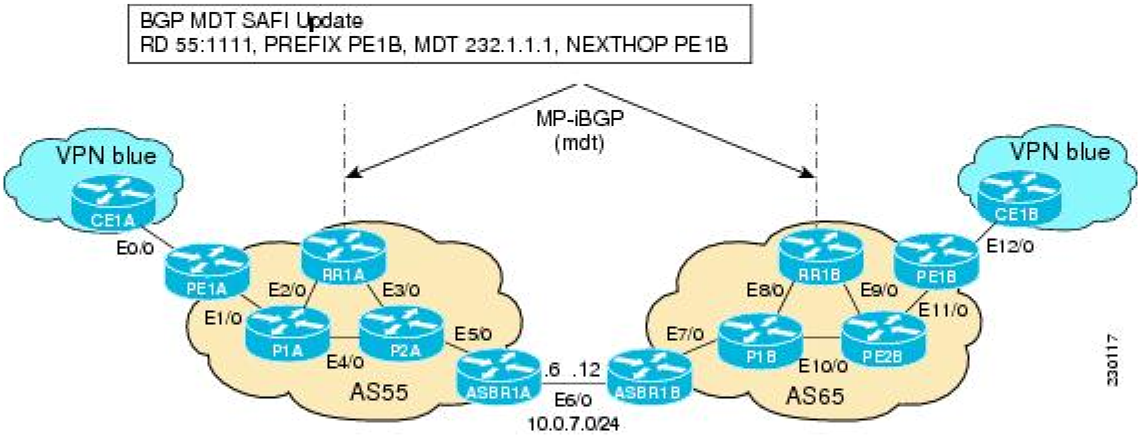
1 As illustrated in the following figure, PE1B advertises the default MDT information for VPN blue to RR1B within the BGP MDT SAFI.

Figure 23 *BGP MDT SAFI Update from PE1B to RR1B*



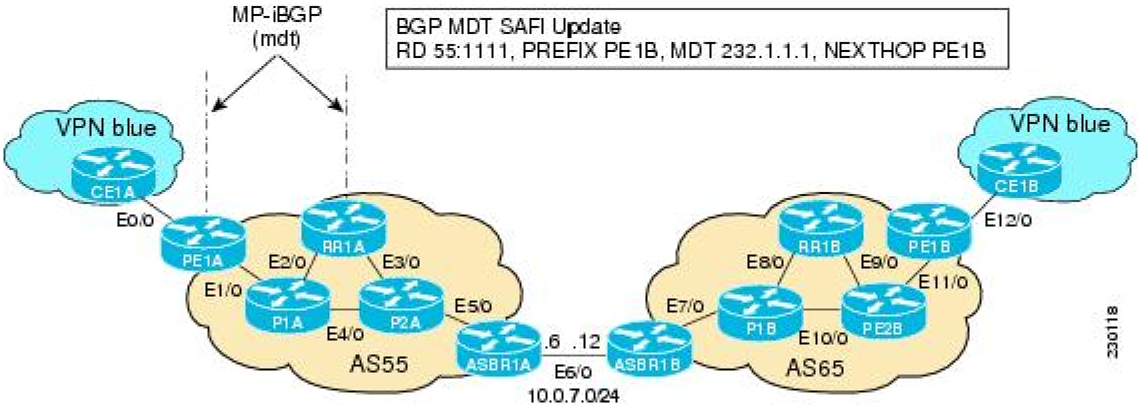
1 As illustrated in the following figure, RR1B receives the MDT SAFI information, and, in turn, advertises this information to RR1A.

Figure 24 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.

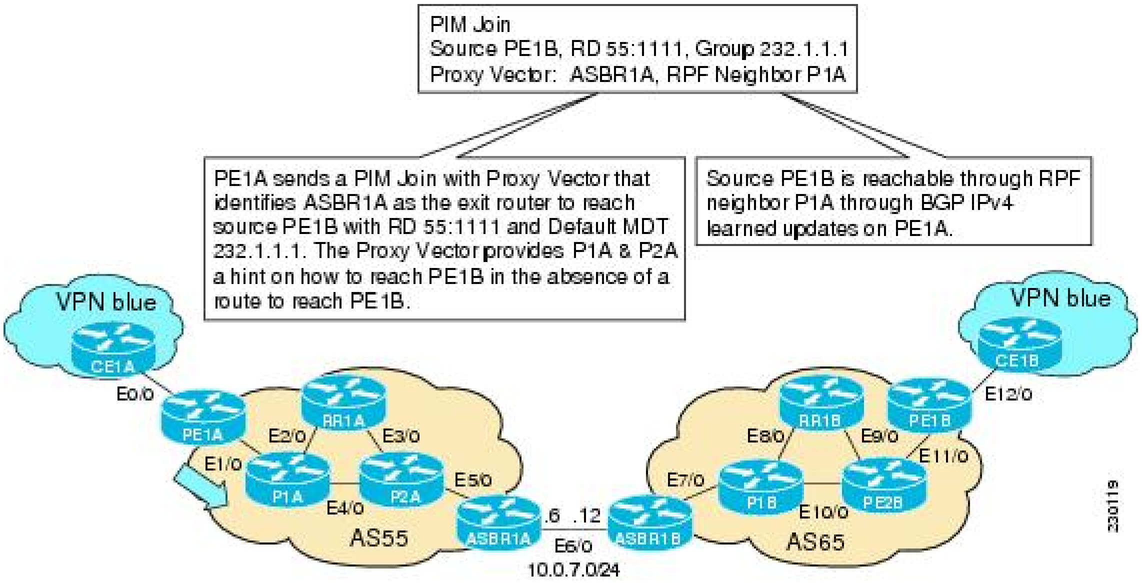
Figure 25 BGP MDT SAFI Update from RR1A to PE1A



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

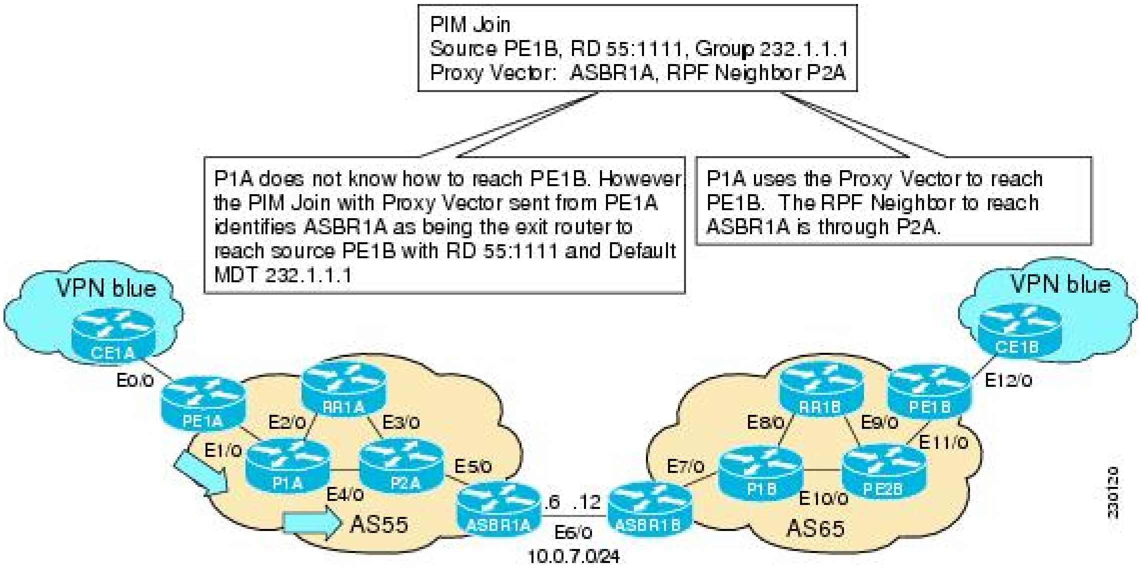
PE1B. Source PE1B is reachable through RPF neighbor P1A through BGP IPv4 learned updates on PE1A.

Figure 26 PIM SSM Join for Default MDT with Proxy Vector from PE1A to P1A


PIM Join
Source PE1B, RD 55:1111, Group 232.1.1.1
Proxy Vector: ASBR1A, RPF Neighbor P1A
PE1A sends a PIM Join with 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 & 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.
VPN blue
CE1A
E0/0
PE1A
RR1A
E2/0
E3/0
E1/0
P1A
P2A
E4/0
E5/0
AS55
ASBR1A
.6
.12
E6/0
10.0.7.0/24
ASBR1B
AS65
E7/0
P1B
E8/0
RR1B
E9/0
PE1B
E11/0
PE2B
E10/0
E12/0
CE1B
VPN blue
230119

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.

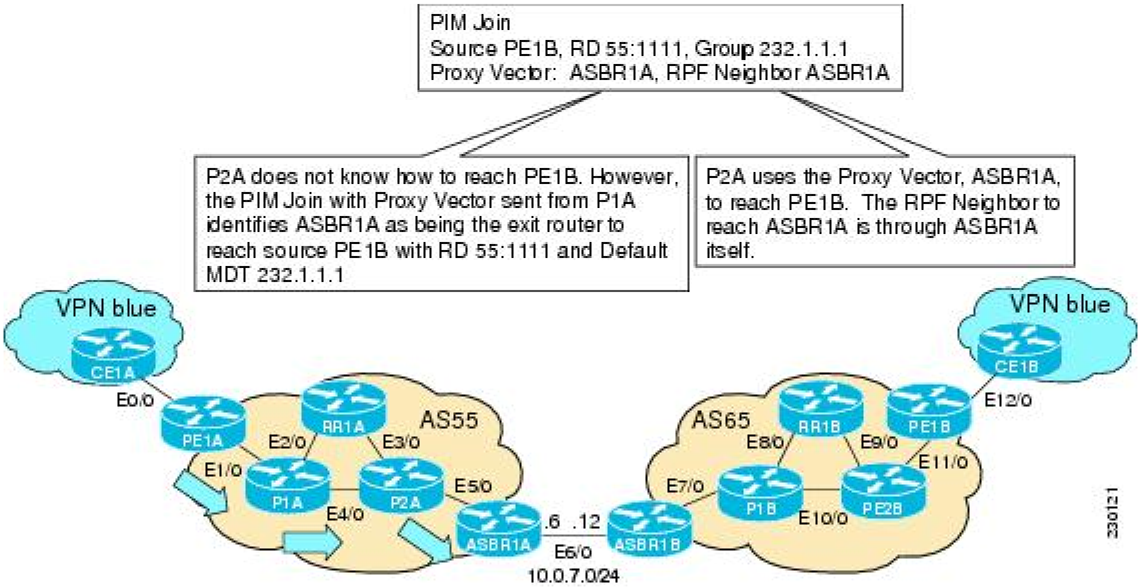
Figure 27 PIM SSM Join for Default MDT with Proxy Vector from P1A to P2A


PIM Join
Source PE1B, RD 55:1111, Group 232.1.1.1
Proxy Vector: ASBR1A, RPF Neighbor P2A
P1A does not know how to reach PE1B. However, the PIM Join with 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.
VPN blue
CE1A
E0/0
PE1A
RR1A
E2/0
E3/0
E1/0
P1A
P2A
E4/0
E5/0
AS55
ASBR1A
.6
.12
E6/0
10.0.7.0/24
ASBR1B
AS65
E7/0
P1B
E8/0
RR1B
E9/0
PE1B
E11/0
PE2B
E10/0
E12/0
CE1B
VPN blue
230120

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 28 ***PIM SSM Join for Default MDT with Proxy Vector from P2A to ASBR1A***



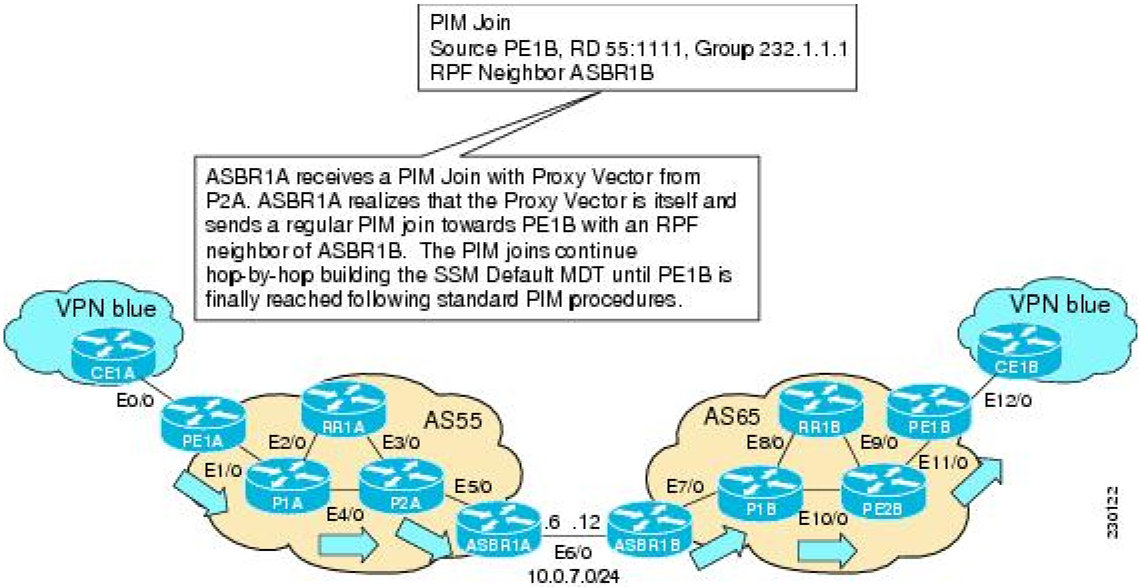
PIM Join
Source PE1B, RD 55:1111, Group 232.1.1.1
Proxy Vector: ASBR1A, RPF Neighbor ASBR1A

P2A does not know how to reach PE1B. However, the PIM Join with 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 ASBR1A is through ASBR1A itself.

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 29 ***PIM SSM Join for Default MDT with Proxy Vector from ASBR1A to PE1B***



PIM Join
Source PE1B, RD 55:1111, Group 232.1.1.1
RPF Neighbor ASBR1B

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.

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 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

	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.
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.

	Command or Action	Purpose
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

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

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

	Command or Action	Purpose

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*]
	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`
		(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.
		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

	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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose

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

	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.

	Command or Action	Purpose
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.
Step 9	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.

	Command or Action	Purpose
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

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, page 79
- Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support, page 79
- Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support, page 80
- Configuring Back-to-Back ASBR PEs - Option A Example, page 81
- Configuring the Exchange of VPNv4 Routes Directly Between ASBRs - Option B Example, page 92
- Configuring the Exchange of VPNv4 Routes Between RRs Using Multihop MP-EBGP, page 100

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
!
```

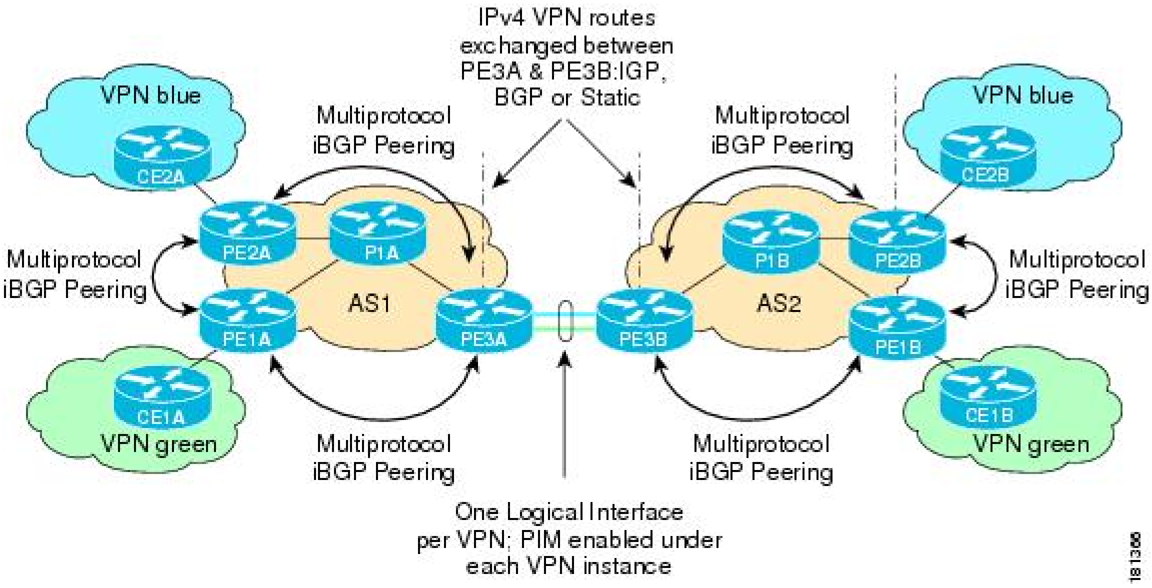
Configuring Back-to-Back ASBR PEs - Option A Example

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 30 Topology for MVPN Inter-AS Support Option A Configuration Example


IPv4 VPN routes exchanged between PE3A & PE3B:IGP, BGP or Static
VPN blue
CE2A
Multiprotocol iBGP Peering
Multiprotocol iBGP Peering
VPN blue
CE2B
PE2A
P1A
Multiprotocol iBGP Peering
AS1
PE1A
PE3A
PE3B
P1B
PE2B
AS2
PE1B
Multiprotocol iBGP Peering
CE1A
VPN green
Multiprotocol iBGP Peering
Multiprotocol iBGP Peering
CE1B
VPN green
One Logical Interface per VPN: PIM enabled under each VPN instance
181366

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

Table 4 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

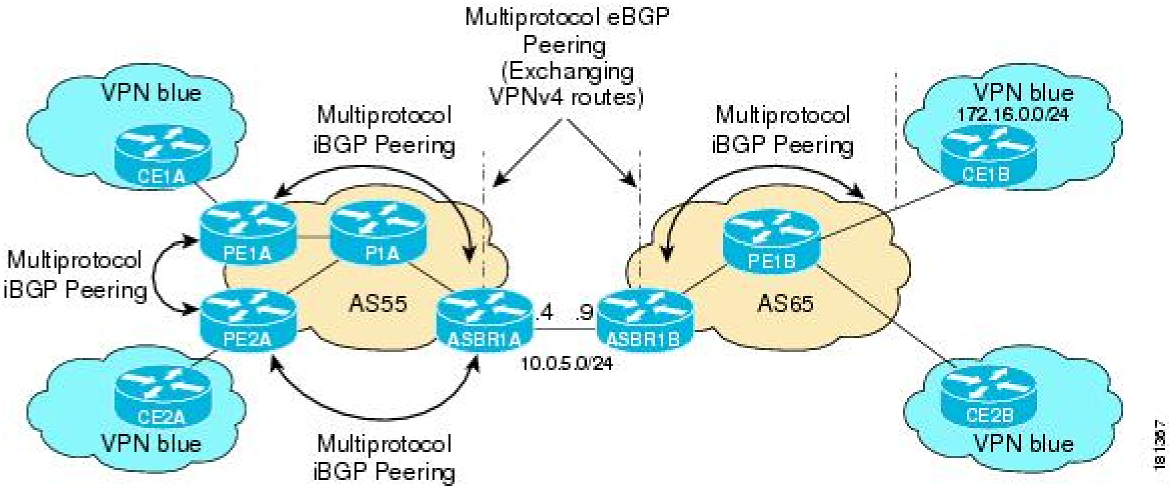
```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE1B
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf green
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 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 31 Topology for MVPN Inter-AS Support Option B Configuration Example


Multiprotocol eBGP Peering (Exchanging VPNv4 routes)
VPN blue
CE1A
Multiprotocol iBGP Peering
Multiprotocol iBGP Peering
VPN blue
172.16.0.0/24
CE1B
Multiprotocol iBGP Peering
PE1A
P1A
PE1B
AS55
AS65
PE2A
ASBR1A
.4
.9
ASBR1B
10.0.5.0/24
CE2A
VPN blue
CE2B
VPN blue
Multiprotocol iBGP Peering

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

Table 5 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
```

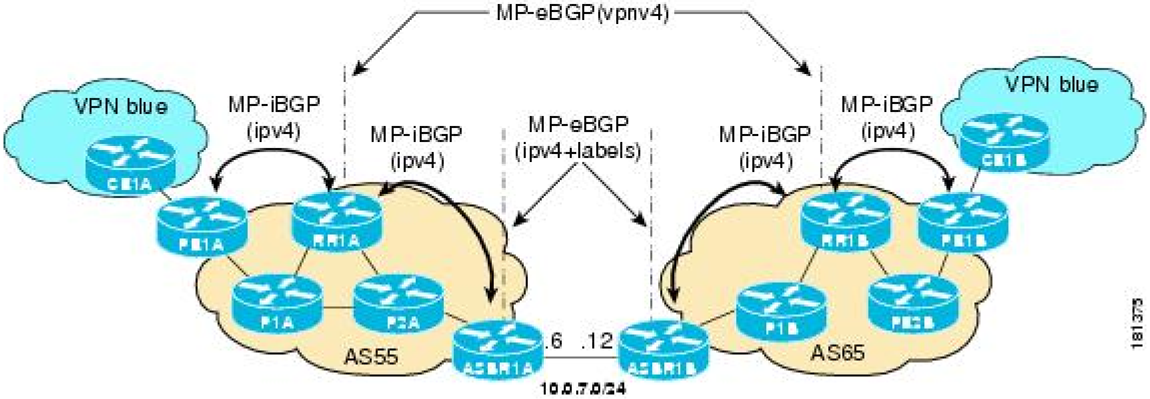
Configuring the Exchange of VPNv4 Routes Between RRs Using Multihop MP-EBGP

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 32 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 6 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
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*
draft-ietf-l3vpn-rfc2547bis[1]	*BGP/MPLS IP VPNs*

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)

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

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

RFC	Title
RFC 5496	The Reverse Path Forwarding (RPF) Vector TLV

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7 Feature Information for Configuring Multicast VPN Inter-AS Support

Feature Name	Releases	Feature Information
BGP Multicast Inter-AS VPN	Cisco IOS XE Release 2.5	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	Cisco IOS XE Release 2.5	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**, **show ip mroute**, **show ip pim neighbor**, **show ip rpf**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Multicast VPN MIB

The Multicast VPN MIB feature introduces the capability for Simple Network Management Protocol (SNMP) monitoring of a Multicast VPN (MVPN) using the MVPN MIB (CISCO-MVPN-MIB).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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

Prerequisites for Multicast VPN MIB

- Before performing the tasks in this module, you must configure MVPN.
- You must configure SNMP on the routers on which the MVPN MIB is to be used.

Restrictions for Multicast VPN MIB

- Currently only IPv4 is supported.
- For all MIB objects with “read-create” access privileges, currently only “read-only” access is supported.

Information About Multicast VPN MIB

Overview of the MVPN MIB

In an MVPN network, a provider pdge (PE) router has a multicast routing table and a Protocol Independent Multicast (PIM) instance associated with every VPN routing and forwarding (VRF) table that is used to define the VPN membership of customer sites attached to the router. There is one global multicast routing table and a table per multicast VRF (MVRF) used to route multicast packets received from a customer edge (CE) router. A set of MVRFs form a multicast domain (MD) when they are connected to potential sources and receivers of multicast traffic. A distinct group address, also known as the Multicast Distribution Tree (MDT) group address, obtained from an administrative pool, is assigned to each multicast domain. MDT groups are used by PE routers to encapsulate and transport multicast traffic within an MD through multicast tunnel interfaces (MTIs).

Initially all multicast data is forwarded using preconfigured MDT default groups. When certain multicast streams exceed a configured bandwidth threshold on the PE router, the multicast data is moved to an MDT data group that is dynamically chosen from an available pool of multicast addresses.

Using the MVPN MIB, network administrators can access MVRF information from PE routers for VPN traffic across multiple CE sites in real time. SNMP operations can be performed to monitor the MVRFs on the PE routers using **get** and **set** commands entered on the network management system (NMS) workstation for which SNMP has been implemented. The NMS workstations is also known as the SNMP manager.

MVPN Information Retrieval Using SNMP and the MVPN MIB

SNMP has historically been used to collect network information. SNMP permits retrieval of critical information from network elements such as routers, switches, and workstations. The MVPN MIB uses SNMP to configure MVRF trap notifications and to gather useful MVPN information in real time.

The MVPN MIB allows MVPN data for the managed devices on your system to be retrieved by SNMP. You can specify the retrieval of MVPN information from a managed device (for example, a router) either by entering commands on that managed device or by entering SNMP commands from the NMS workstation to gather MVPN information. MVPN MIB requests for information are sent from an NMS workstation to the router using SNMP and are retrieved from the router. This information can then be stored or viewed, thus allowing MVPN information to be easily accessed and transported across a multivendor programming environment.

MVPN MIB Objects

The MVPN MIB defines managed objects that enable a network administrator to remotely monitor the following MVPN information:

- The state of the MVRFs including the name of the MVRF, whether they are active, and the number of active multicast-enabled interfaces
- MDT default group address and encapsulation information
- Next hop information used to receive Border Gateway Protocol (BGP) MDT updates for Source Specific Multicast (SSM) mode

- Traffic threshold that determines switchover to an MDT data group
- Type of MDT group being used for a given (S, G) multicast route entry that exists on each configured MVRF, source address, and group address of the multicast route entry
- Source and group address used for encapsulation
- Information on MDT data groups currently joined
- Information on MVPN-specific MDT tunnels present in the device
- Trap notifications enabled on the router

Note For a complete description of the objects supported by the MVPN MIB, see the CISCO_MVPN_MIB.my file, available on Cisco.com at http://www.cisco.com/go/mibs .

MVRF Trap Notifications

An MVPN router can be configured to send MVRF (ciscoMvpnMvrfChange) trap notifications. A ciscoMvpnMvrfChange trap notification signifies a change about an MVRF in the device. The change event can be the creation of an MVRF, the deletion of an MVRF, or an update on the default or data multicast distribution tree (MDT) configuration of an MVRF. The change event is indicated by the ciscoMvpnGenOperStatusChange object embedded in the trap notification.

How to Configure Multicast VPN MIB

Configuring the Router to Send MVRF Trap Notifications

Perform this task to configure the router to use SNMP to send MVRF trap notifications.

Note Before the MVPN MIB can be used, the SNMP server for the router must be configured. To enable the SNMP server on the router, perform Steps 3 and 4. If an SNMP server is already available, omit Steps 3 and 4 and proceed to Step 5.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **snmp-server community** *string* **ro**
   -
   - **snmp-server community** *string* **rw**
4. **snmp-server host {** *hostname* | *ip* - *address* **} version 2c** *community* - *string*
5. **snmp-server enable traps mvpn**
6. **end**

DETAILED STEPS

	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	Do one of the following:
	• **snmp-server community** *string* **ro**
	•
	• **snmp-server community** *string* **rw**
	Example:
	`Router(config)# snmp-server community public ro`
	Example:
	Example:
	`Router(config)# snmp-server community public rw`
		Sets up the community access string to permit access to SNMP.
		• The *string* argument is a community string that consists of from 1 to 32 alphanumeric characters and functions much like a password, permitting access to the SNMP protocol. Blank spaces are not permitted in the community string.
		• Specifying the **snmp-server community** command with the **ro** keyword configures read-only access. SNMP management stations using this string can only retrieve MIB objects.
		or
		• Specifying the **snmp-server community** command with the **rw** keyword configures read-write access. SNMP management stations using this string can retrieve and modify MIB objects.

	Command or Action	Purpose
Step 4	**snmp-server host {** *hostname* | *ip - address* **}** **version 2c** *community - string* **Example:** `Router(config)# snmp-server host 192.168.1.1 version 2c public`	Specifies the recipient of an SNMP notification operation.
Step 5	**snmp-server enable traps mvpn** **Example:** `Router(config)# snmp-server enable traps mvpn`	Enables the router to send MVRF trap notifications.

Step 6	**end**
	Example:
	`Router(config)# end`
		Exits the current configuration mode and returns to privileged EXEC mode.

Configuration Examples for Multicast VPN MIB

- Configuring the Router to Send MVRF Trap Notifications Example, page 117

Configuring the Router to Send MVRF Trap Notifications Example

The following example shows how to configure a router to use SNMP to send MVRF trap notifications:

```
!
snmp-server community public rw
snmp-server enable traps mvpn
snmp-server host 10.3.32.154 version 2c public
!
```

Additional References

Related Documents

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

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
• CISCO-MVPN-MIB.my	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
No new or modified RFCs are supported by this feature.	--

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 Multicast VPN MIB

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

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

Table 8 Feature Information for Multicast VPN MIB

Feature Name	Releases	Feature Information
Multicast VPN MIB	Cisco IOS XE Release 2.5	The Multicast VPN MIB feature introduces the capability for SNMP monitoring of an MVPN using the MVPN MIB (CISCO-MVPN-MIB). The following command was introduced by this feature: **snmp server enable traps mvpn.**

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.