

Configuring MVPNs

This chapter contains information on how to configure multicast virtual private networks (MVPNs)

- About MVPNs, on page 1
- BGP Advertisement Method - MVPN Support, on page 4
- Prerequisites for MVPNs, on page 4
- Guidelines and Limitations for MVPNs, on page 5
- Default Settings for MVPNs, on page 6
- Configuring MVPNs, on page 6
- Configuration Examples for MVPN, on page 14

About MVPNs

The multicast virtual private networks (MVPNs) feature allows you to support multicast connectivity over Layer 3 VPN. IP multicast is used to stream video, voice, and data to an VPN network core.

Historically, point-to-point tunnels were the only way to connect through an enterprise or service provider network. Although such tunneled networks had scalability issues, they were the only means of passing IP multicast traffic through a virtual private network (VPN). Because Layer 3 VPNs support only unicast traffic connectivity, deploying with a Layer 3 VPN allows operators to offer both unicast and multicast connectivity to Layer 3 VPN customers

MVPNs allows you to configure and support multicast traffic in an MVPN environment. MVPNs support routing and forwarding of multicast packets for each individual virtual routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the enterprise or service provider backbone. IP multicast is used to stream video, voice, and data to a VPN network core.

A VPN allows 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.

MVPNs allow an enterprise to transparently interconnect its private network across the network backbone. Using MVPNs to interconnect an enterprise network does not change the way that an enterprise network is administered and it does not change general enterprise connectivity.

MVPN Routing and Forwarding and Multicast Domains

MVPNs introduce 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, the router

forwards the data or control packets according to the information in the MVPN routing and forwarding (MVRF).

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 that are associated with that enterprise.

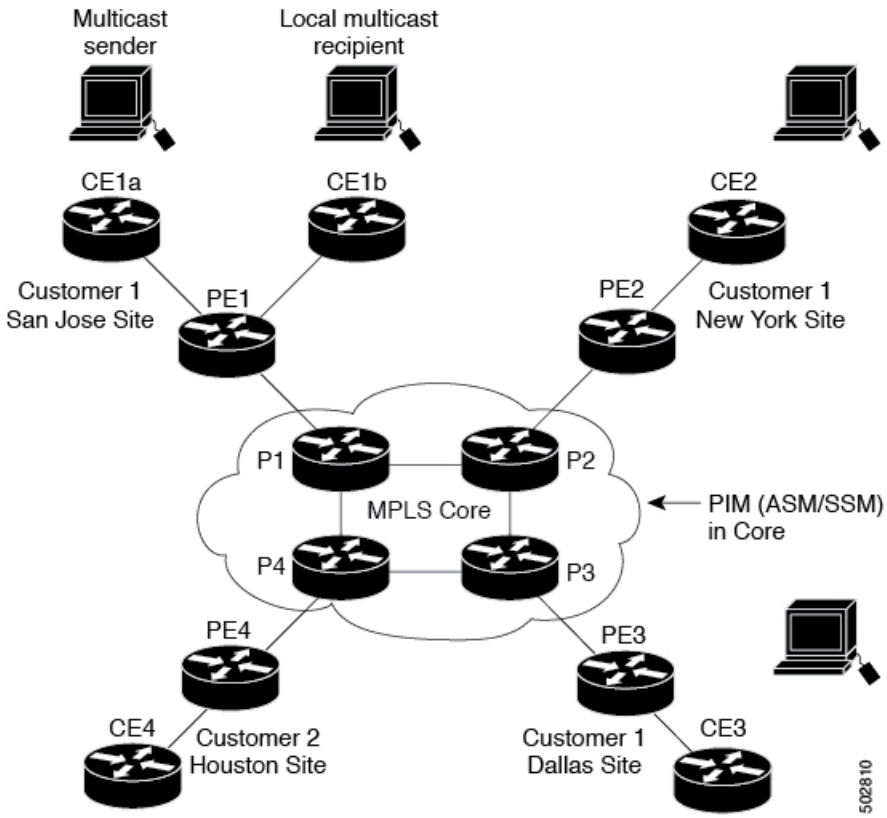
Multicast Distribution Trees

MVPNs establish a static default multicast distribution tree (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.

MVPNs also support the dynamic creation of MDTs for high-bandwidth transmission. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the VPN core.

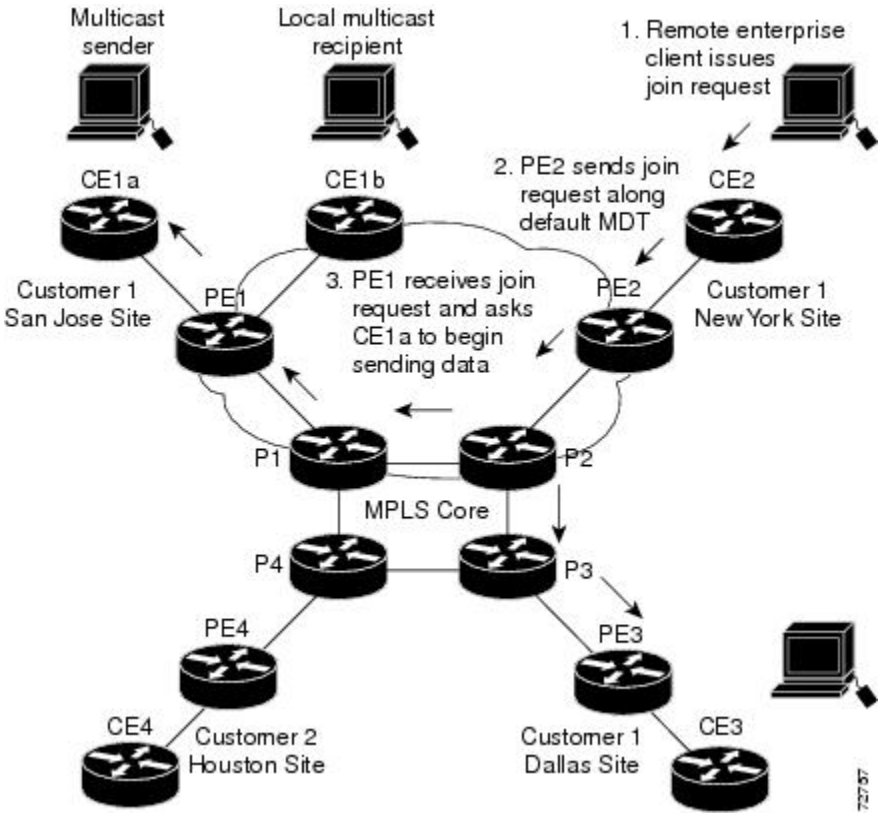
In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. A one-way multicast presentation is occurring in San Jose. The service provider network supports all three sites that are associated with this customer, in addition to the Houston site 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 following 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 that is 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 that is associated with the multicast session source, receives the request. The following figure depicts that the PE router forwards the request to the CE router that is associated with the multicast source (CE1a).

Figure 2: Initializing the Data MDT



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. 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 joins the data MDT and receives traffic on it. (If the data MDT had not been configured and only the default MDT had been configured, all the customer sites would have received the traffic even though they were not interested in it.) PE routers maintain a PIM relationship with other PE routers over the default MDT and a PIM relationship with its directly attached P routers.

Multicast Tunnel Interface

An MVPN routing and forwarding (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 that the MVRF uses to access the multicast domain. The interface is a conduit that connects an MVRF and the global MVRF. One tunnel interface is created per MVRF.

Benefits of MVPNs

The benefits of MVPNs are as follows:

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

BGP Advertisement Method - MVPN Support

When you configure the default MDT 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. 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 a rendezvous point (RP). The source provider edge (PE) address and default MDT address are sent using the Border Gateway Protocol (BGP).

BGP MDT SAFI

BGP MDT SAFI is the BGP advertisement method that is used for MVPNs. In the current release, only IPv4 is supported. MDT SAFI has the following settings:

- AFI = 1
- SAFI = 66

In Cisco NX-OS, the source PE address and the MDT address are passed to PIM using BGP MDT SAFI updates. The Route Descriptor (RD) type has changed to RD type 0 and BGP determines the best path for the MDT updates before passing the information to PIM.

You must configure the MDT SAFI address family for BGP neighbors by using the **address-family ipv4 mdt** command. You must still enable neighbors that do not support the MDT SAFI for the MDT SAFI in the local BGP configuration. Prior to the MDT SAFI, additional BGP configuration from the VPNv4 unicast configuration was not needed to support MVPNs.

Prerequisites for MVPNs

MVPNs configuration has the following prerequisites:

- Ensure that you have configured MPLS and Label Distribution Protocol (LDP) in your network. All routers in the core, including the PE routers, must be able to support MPLS forwarding. VPNv4 routes are not installed by BGP if labeled paths do not exist for PE source addresses.
- Ensure that you have installed the correct license for MPLS and any other features you will be using with MPLS.

Guidelines and Limitations for MVPNs

Configuring MVPNs has the following guidelines and limitations:

- MVPNs are supported beginning with Cisco NX-OS Release 9.3(3).
- MVPNs are supported only for Cisco Nexus 9500 platform switches with -R/-RX line cards (except the N9K-X96136YC-R line card).
- Bidirectional Forwarding Detection (BFD) is not supported on the Multicast Tunnel Interface (MTI).
- By default, the BGP update source is used as the source of the MVPN tunnel. However, you can use the mdt source to override the BGP update source and provide a different source to the multicast tunnel.
- MVPN supports a maximum of 16 MDT source interfaces.
- You must configure the MDT SAFI on all routers that participate in the MVPN operations.
- Extended communities are needed for VPNv4 interior BGP (iBGP) sessions to carry the connector attribute.
- MDT MTU configuration is not supported. The maximum customer multicast packet size that can be sent over MVPN is limited by the MTU of the core interfaces. For example:
  - MTU 1500 – Customer IP packet size = 1476
  - MTU 9216 – Customer IP packet size = 9192
- Some of the MVPN multicast control packets are classified into the copp-system-p-class-l2-default CoPP policy. We recommend modifying the CoPP policy to increase the policer rate under this class if the violated count increases.
- MDT bidir-enable is not supported.
- vPCs are not supported for MVPN.
- Data MDT entries are not cached when the transit PE router does not have receivers and is connected to a CE which is a RP. The data MDT entries are cached only when a local receiver is attached to this PE router. However, there is a delay in the switchover because the entries are not pre-downloaded.
- For Date MDT, only 'immediate-switch' mode is supported. Threshold based switching is not supported.
- Sub-interface and SVI support between PE and P /PE devices is not available.
- MVPN Consistency-checker is not supported in Cisco Nexus Release 9.3(3).
- Statistics for MTI interfaces are not supported in Cisco Nexus Release 9.3(3).
- Maximum 40G multicast traffic per ASIC is supported in Cisco Nexus Release 9.3(3).
- You are allowed to configure a non-default MTU on a VRF only after you remove the MDT MTU configuration from the VRF. This occurs when the MTI is down in a switch in which the VRF with the non-default MDT MTU is available.
- Due to a hardware limitation, the MTI TX packet counts are not supported. However, all MTI RX packet and byte counts are supported.

Default Settings for MVPNs

Table 1: Default MVPN Parameters

Parameters	Default
mdt default *address*	No default
mdt enforce-bgp-mdt-safi	Enabled
mdt source	No default
mdt ip pim hello-interval *interval*	30000 ms
mdt ip pim jp-interval *interval*	60000 ms
mdt default asm-use-shared-tree	Disabled

Configuring MVPNs

This chapter describes how to configure multicast virtual private networks (MVPNs) on Cisco NX-OS devices.

Note For MVPN, a new TCAM region "ing-mvpn" is used (with default size of 10). This region is carved automatically hence you need not carve it. To verify if this TCAM region is carved or not, you can use the following commands:

```
switch# show hardware access-list tcam region | i ing-mvpn
Ingress mVPN [ing-mvpn] size = 10
switch#
```

If the region is not carved due to any reason (size shows is 0), you can use the following command to carve the TCAM region to size 10 and reload the device. The TCAM is expected to be carved to size 10.

```
switch (config)# hardware access-list tcam region ing-mvpn 10
WARNING: On module 2,
WARNING: On module 4,
Warning: Please reload all linecards for the configuration to take effect
switch (config)#
```

Enabling MVPNs

Beginning with Cisco NX-OS Release 9.3(3), you can configure MVPNs on Cisco Nexus 9500-R switches.

Before you begin

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

- Ensure that the **router-id force** command is used when configuring MPLS LDP. Cisco strongly recommends explicitly specifying the router ID with a loopback interface to ensure consistent LDP operation and avoid potential issues with interface initialization order.

Procedure

	Command or Action	Purpose

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

Step 2	**feature bgp** **Example:** `switch(config)#feature bgp`	Enables BGP feature and configurations.
Step 3	**feature pim** **Example:** `switch(config)#feature pim`	Enables the PIM feature.
Step 4	**feature mvpn** **Example:** `switch(config)#feature mvpn`	Enables the MVPN feature.
Step 5	**feature mpls l3vpn** **Example:** `switch(config)#feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature. This determines the unicast routes across sites.
Step 6	**feature mpls ldp** **Example:** `switch(config)#feature mpls ldp`	Enables the MPLS Label Distribution Protocol (LDP). Ensure to configure the **router-id force** command for deterministic LDP router ID selection.

Enabling PIM on Interfaces

You can configure Protocol Independent Multicast (PIM) on all interfaces that are used for IP multicast. We recommend that you configure PIM sparse mode on all physical interfaces of provider edge (PE) routers that connect to the backbone. We also recommend that you configure PIM sparse mode on all loopback interfaces if they are used for BGP peering or if their IP address is used as an RP address for PIM.

Procedure

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

	Command or Action	Purpose
	`switch#configure terminal` `switch(config)#`	
Step 2	**ip pim sparse-mode** **Example:** `switch(config)#ip pim sparse-mode`	Enables PIM sparse mode on the interface.

Configuring a Default MDT for a VRF

You can configure a default MDT for a VRF.

Before you begin

The default MDT must be the same that is configured on all routers that belong to the same VPN. The source IP address is the address that you use to source the BGP sessions.

Procedure

	Command or Action	Purpose

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

Step 2	**vrf context** *VRF_NAME* **Example:** `switch(config)#vrf context vrf1`	Configures the VRF.
Step 3	**mdt default** *address* **Example:** `switch(config)#mdt default 232.0.0.1`	Configures the multicast address range for data MDTs for a VRF as follows: • A tunnel interface is created as a result of this command. • By default, the destination address of the tunnel header is the address argument.

Configuring MDT SAFI for a VRF

By default, MDT subsequent address family identifiers (SAFI) for a VRF are enforced. If desired, you can configure MDT to interoperate with peers that do not support MDT SAFI.

Procedure

	Command or Action	Purpose

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

Step 2	**vrf context** *VRF_NAME*
	Example:
	`switch(config)#vrf context vrf1`
	`switch(config-vrf)#`
		Configures the VRF.

Step 3	**no mdt enforce-bgp-mdt-safi** **Example:** `switch(config-vrf)#no mdt enforce-bgp-mdt-safi`	Enables MDT to interoperate with peers that do not support MDT SAFI. Initially only the (*,G) entry for the default MDT group is populated if it falls within the Any Source Multicast (ASM) range. Then later, based on traffic, the (S,G) entries are learned like regular ASM routes. Removing the **no** option from the command enforces the use of MDT SAFI for the specified VRF.

Configuring the MDT Address Family in BGP for MVPNs

You can configure an MDT address family session on PE routers to establish MDT peering sessions for MVPNs.

Use the **address-family ipv4 mdt** command under neighbor mode to configure an MDT address-family session. MDT address-family sessions are used to pass the source PE address and MDT address to PIM using BGP MDT Subaddress Family Identifier (SAFI) updates.

Before you begin

Before MVPN peering can be established through an MDT address family, you must configure MPLS in the BGP network and multiprotocol BGP on PE routers that provide VPN services to CE routers.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**feature bgp** *as-number* **Example:** `switch(config)#feature bgp 65635`	Enters switch configuration mode and creates a BGP routing process.

Step 3	**vrf context** *VRF_NAME*
	Example:
	`switch(config)#vrf context vpn1`
	`switch(config-vrf)#`
		Defines a VPN routing instance identified by vrf-name and enters VRF configuration mode. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 4	**rd** *route-distinguisher* **Example:** `switch(config-vrf)#rd 1.2.1`	Assigns a route distinguisher to the VRF vrf-name. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3 • 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 5	**address-family ipv4 unicast**
	Example:
	`switch(config-vrf)#address-family ipv4unicast`
	`switch(config-vrf-af)#`
		Specifies the IPv4 address family type and enters address family configuration mode

Step 6	**route-target import** *route-target-ext-community* **Example:** `switch(config-vrf-af)# route-target import 1.0.1`	Specifies 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 list of import route-target extended communities. You can enter the *route-target-ext-community* argument in either of these formats: • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3 • 32-bit IP address: your 16-bit number, for example, for example, 192.0.2.1:1
Step 7	**route-target export** *route-target-ext-community* **Example:** `switch(config-vrf-af)# route-target export 1.0.1`	Specifies a route-target extended community for a VRF. The **export** keyword imports routing information from the target VPN extended community.

	Command or Action	Purpose
		The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF list of import route-target extended communities. You can enter the *route-target-ext-community* argument in either of these formats: • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3 • 32-bit IP address: your 16-bit number, for example, for example, 192.0.2.1:1

Step 8	**router bgp** *as-number*
	Example:
	`switch(config)#router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 9	**address-family ipv4 mdt** **Example:** `switch(config-router)#address-family ipv4 mdt`	Enters IPv4 MDT address family configuration mode.

Step 10	**address-family** {**vpn4**} [**unicast**]
	Example:
	`switch(config-router-af)# address-family vpnv4`
	`switch(config-router-af)#`
		Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 or VPNv6 address prefixes. The optional **unicast** keyword specifies VPNv4 or VPNv6 unicast address prefixes.

Step 11	**address-family** {**ipv4**} **unicast**
	Example:
	`switch(config-router-af)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters address family configuration mode for configuring routing sessions that use standard IPv4 or IPv6 address prefixes.

Step 12	**neighbor** *neighbor-address* **Example:** `switch(config-switch-af)# neighbor 192.168.1.1`	Enters neighbor configuration mode.
Step 13	**update source** *interface* **Example:** `switch(config-switch-neighbor)# update-source loopback 1`	Sets the update source as loopback1.

	Command or Action	Purpose
Step 14	**address-family ipv4 mdt** **Example:** `switch(config-router-neighbor)# address-family ipv4 mdt`	Enters address family configuration mode to create an IP MDT address family session.
Step 15	**send-community extended** **Example:** `switch(config-router-neighbor-af)#send-community extended`	Specifies that extended communities attribute should be sent to a BGP neighbor.
Step 16	**show bgp** {**ipv4**} **unicast neighbors vrf***VRF_NAME* **Example:** `switch(config-router-neighbor-af)#show bgp ipv4 unicast neighbors vrf vpn1`	Displays information about BGP neighbors. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 17	**copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)#copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Data MDT

You can configure a data MDT. Multicast groups that are used to create the data MDT are dynamically chosen from a pool of configured IP addresses. If the number of streams is greater than the maximum number of data MDTs per VRF per PE, multiple streams share the same data MDT.

Before you begin

Before configuring a data MDT, you must configure the default MDT on the VRF.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch#configure terminal switch(config)#`	Enters global configuration mode.
Step 2	**vrf context** *VRF_NAME* **Example:** `switch#ip vrf vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 3	**mdt data** *prefix* [**immediate-switch**] [**route-map** *policy-name*]	Specifies a range of values as follows:

	Command or Action	Purpose

	Example:
	`switch(config-vrf)# mdt data 225.1.1.1/32 immediate-switch route-map test`
	Example:
	`switch(config-vrf)# mdt data 225.1.1.1/32 route-map test`
		• The *prefix* specifies the range of addresses to be used in the data MDT pool.
		• The *policy-name* defines a policy file that defines which customer data streams should be considered for switching onto the data MDT.
		Note
		Entering this command with or without the immediate-switch option has the same effect.

Step 4	**exit** **Example:** `switch(config)#exit`	Returns to global configuration mode.

Verifying the MVPN Configuration

To display the MVPN configuration, perform one of the following tasks:

Table 2: Verifying the MVPN Configuration

Command	Purpose
show interface	Displays details of an interface.
show ip mroute vrf	Displays multicast routes.
show ip pim event-history mvpn	Displays the details of the MVPN event history logs.
show ip pim mdt	Displays the details of MTI tunnels created by MVPN.
show ip pim mdt receive vrf vrf-name	Displays the mapping of the customer source, the customer group to data MDT source, and the data MDT group on the receiving side.
show ip pim mdt send vrf vrf-name	Displays the mapping of the customer source, the customer group to data MDT source, and the data MDT group on the sending side.
show ip pim neighbor	Displays details of established PIM neighbors.
show ip route detail	Displays the details of the unicast routing tables.
show mvpn bgp mdt-safi	Displays the BGP MDT SAFI database in MVPN.
show mvpn mdt encap vrf *vrf*	Displays the encapsulation table in MVPN. This table indicates how MVPN packets are encapsulated when sent out on the default vrf.

Command	Purpose
show mvpn mdt route	Displays details of the default and MDT routes. This data determines how customer data and control traffic is sent on the default VRF.
show routing [**ip**] **multicast mdt encap**	Displays the encapsulation table in the MRIB. This table indicates how MVPN packets are encapsulated when sent out on the default vrf.

Configuration Examples for MVPN

The following example shows how to configure an MVPN with two contexts:

```
vrf context vpn1
  ip pim rp-address 10.10.1.2 -list 224.0.0.0/8
  ip pim ssm range 232.0.0.0/8
  rd auto
  mdt default 232.1.1.1
  mdt source loopback1
  mdt data 225.122.111.0/24 immediate-switch
vrf context vpn4
  ip pim rp-address 10.10.4.2 -list 224.0.0.0/8
  ip pim ssm range 232.0.0.0/8
  mdt default 235.1.1.1
  mdt asm-use-shared-tree
ip pim rp-address 10.11.0.2 -list 224.0.0.0/8
ip pim rp-address 10.11.0.4 -list 235.0.0.0/8
ip pim ssm range 232.0.0.0/8
```

The following example shows how to assign to the VPN routing instance a VRF named blue. The MDT default for a VPN VRF is 10.1.1.1, and the multicast address range for MDTs is 10.1.2.0 with wildcard bits of 0.0.0.3:

```
Vrf context blue
mdt data 225.122.111.0/24 immediate-switch
```