Configuring MVPNs

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

This chapter includes the following sections:

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your 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 “New and Changed Information” chapter or the Feature History table below.

Information About MVPNs

You can use an MVPN feature to support multicast over a 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.

This section includes the following topics:

MVPN Overview

An MVPN allows an operator 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). MVPNs do 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 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. 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

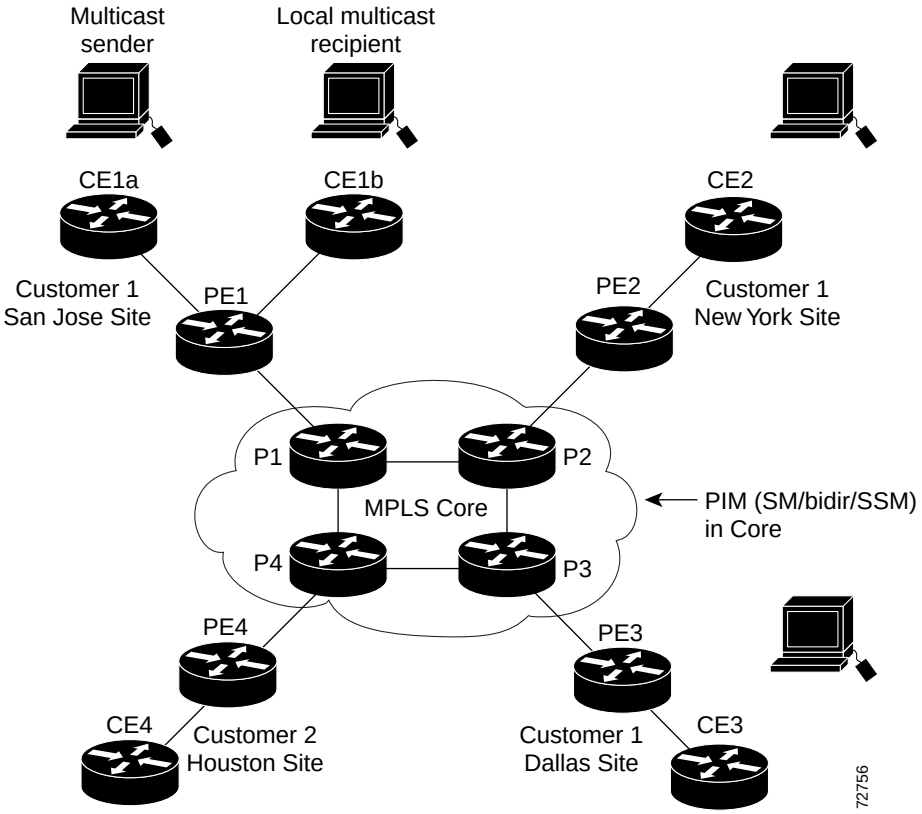
information about the data MDT, to all routers on the default MDT. Once every second, the PE router examines the statistics to determine whether a multicast stream has exceeded the data MDT threshold. After a PE router sends the UDP message, it waits 3 more seconds before switching over.

Data MDTs are created for bidirectional routes if you use the **mdt data bidir-enable** command in that VRF. (Data MDTs are not created for bidirectional customer routes by default.)

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.

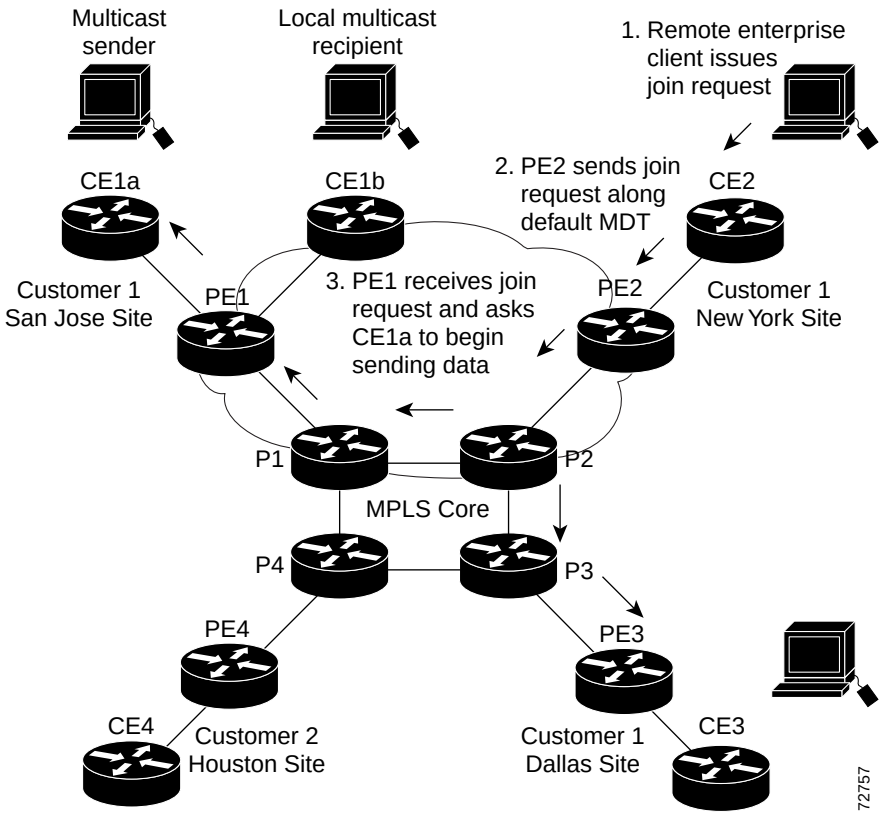
The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. PE4 is not part of the default MDT, because it is associated with a different customer. Figure 34-1 shows that no data flows along the default MDT, because no one outside of San Jose has joined the multicast.

Figure 34-1 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. Figure 34-2 depicts that the PE router forwards the request to the CE router that is associated with the multicast source (CE1a).

Figure 34-2 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. Immediately after 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 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

Information About the BGP Advertisement Method for MVPN Support

This section includes the following topics:

Overview

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.

Licensing Requirements for MVPNs

Product	License Requirement
Cisco NX-OS	MVPNs require an MPLS license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for MVPNs

Configuring MVPNs 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

MVPNs have the following configuration guidelines and limitations:

- 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.
- Cisco NX-OS Release 5.2(4) and later 5.x releases as well as Cisco NX-OS Release 6.1(1) and later 6.x releases support multicast GRE tunnel interfaces for PE-CE routing with MVPN.

MDT SAFI has the following configuration and limitations guidelines:

- 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.

Default Settings for MVPNs

Table 34-1 lists the default settings for MVPN parameters.

Table 34-1 Default MVPN Parameters

Parameters	Default
mdt default *address*	No default
mdt enforce-bgp-mdt-safi	Enabled
mdt data *threshold*	0 Kilobits/second

Table 34-1 Default MVPN Parameters (continued)

Parameters	Default
mdt source	No default
mdt mtu *mtu*[1]	1376 bytes
mdt ip pim hello-interval *interval*	30000 ms
mdt ip pim jp-interval *interval*	60000 ms
mdt data bidir-enable[2]	Disabled
mdt default asm-use-shared-tree [only][3]	Disabled

1. The default MDT MTU value for Cisco Catalyst 6000 Series switches is 1500 bytes, which is different from the default value of 1376 bytes for Cisco Nexus 7000 Series switches. To avoid an interoperability issue (especially when migrating from the Cisco Catalyst 6000 Series switches), make sure to use the appropriate MDT MTU value.
2. Enables data MDTs to be created for bidir customer routes.
3. The receiving PE's do not trigger an (S,G) join toward the source for the MDT routes when default MDT is in PIM ASM mode.

Configuring MVPNs

This section includes the following topics:

Enabling Features

You enable required features by using the detailed steps in this section. This procedure is required for enabling features.

Note Some protocols, such as rip/ospf, must be running both on customer VRFs as well as the core.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. **feature pim**
4. **feature mvpn**
5. **feature mpls l3vpn**
6. **feature tunnel**
7. **feature mpls ldp**

DETAILED STEPS

	Command	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 the BGP feature.
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, which is needed to determine unicast routes across sites.
Step 6	**`feature tunnel`** **Example:** `switch(config)# feature tunnel`	Enables the tunnel feature.
Step 7	**`feature mpls ldp`** **Example:** `switch(config)# feature mpls ldp`	Enables the MPLS Label Distribution Protocol (LDP).

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.

Note This procedure is required for enabling PIM on interfaces. For more information on PIM, see the *Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **ip pim sparse-mode**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`ip pim sparse-mode`** **Example:** `switch (config-if)# 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.

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.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **mdt default** *address*

DETAILED STEPS

	Command	Purpose
Step 1	**`configure terminal`** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**`vrf context`** *`vrf-name`* **Example:** `switch(config)# vrf context vrf1`	Sets the VRF context by assigning a VRF name.
Step 3	**`mdt default`** *`address`* **Example:** `switch(config-vrf)# 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.

Enforcing MDT SAFI for a VRF

You can enforce the use of MDT subsequent address family identifiers (SAFI) for a VRF, or you can configure MDT to interoperate with peers that do not support MDT SAFI.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. [**no**] **mdt enforce-bgp-mdt-safi**

DETAILED STEPS

	Command	Purpose

Step 1	`configure terminal`
	
	Example:
	`switch# configure terminal`
		Enters global configuration mode.

Step 2	`vrf context` *vrf-name*
	
	Example:
	`switch(config)# vrf context vrf1`
	`switch(config-vrf)#`
		Sets the VRF context by assigning a VRF name.

Step 3	[`no`] `mdt enforce-bgp-mdt-safi` **Example:** `switch(config-vrf)# mdt enforce-bgp-mdt-safi`	Enforces the use of MDT SAFI for the specified VRF. The **no** form of this command enables MDT to interoperate with peers that do not support MDT SAFI. When the **no** form is used, 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.

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.

Prerequisites

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.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp** *as-number*
3. **vrf context** *vrf-name*

4. **rd** *route-distinguisher*
5. **address-family ipv4** *unicast*
6. **route-target import** *route-target-ext-community*
7. **route-target export** *route-target-ext-community*
8. **router bgp** *as-number*
9. **address-family ipv4 mdt**
10. **address-family** {**vpnv4**} [**unicast**]
11. **address-family** {**ipv4**} [**unicast**]
12. **neighbor** *neighbor-address*
13. **update source** *interface*
14. **address-family ipv4 mdt**
15. **address-family vpnv4** [**unicast**]
16. **send-community extended**
17. (Optional) **show bgp** {**ipv4**} **unicast neighbors vrf** *vrf-name*
18. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`feature bgp`** *`as-number`* **Example:** `switch(config)# feature bgp 65535`	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 ipv4 unicast`
	`switch(config-vrf-af)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

	Command	Purpose
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, 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 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 list of export 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, 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** {**vpnv4**} [**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.

	Command	Purpose
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-router-neighbor)# update-source loopback 1`	Sets the update source as loopback1.
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	**address-family vpnv4** [**unicast**]
	
	Example:
	`switch(config-router-neighbor-af)# address-family vpnv4`
	`switch(config-router-neighbor-af)#`
		Enters VPNv4 address family configuration mode.

Step 16	**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 17	**show bgp** {**ipv4**} **unicast neighbors vrf** *vrf-name* **Example:** `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors vrf vpn1`	(Optional) Displays information about BGP neighbors. The *vrf-name* argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 18	**copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	(Optional) 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. See Appendix A, "Configuration Limits for Cisco NX-OS MPLS" for information on the maximum supported number of data MDTs per VRF per PE.

Prerequisites

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

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **mdt data data** *prefix* [**threshold** *threshold-value*] [**routemap** *policy-name*]
4. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal	Enters global configuration mode.
Step 2	**vrf context** *vrf-name* **Example:** switch(config)# ip vrf vrf1	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 3	**mdt data data prefix** [***threshold*** *threshold-value*] [**routemap** *policy-name*] **Example:** switch(config-vrf)# mdt data 232.7.7.0/24 threshold 10 route-map rmap2mdt data 239.192.20.32 0.0.0.15 threshold 1	Specifies a range of threshold values as follows: • *Prefix* specifies the range of addresses to be used in the data MDT pool. • *Threshold-value* specifies the threshold in kilobits per second when the stream is switched to the data MDT. • *Policy-name* defines a policy file that defines which customer data streams should be considered for switching onto the data MDT.
Step 4	**exit** **Example:** switch(config-vrf)# exit	Returns to global configuration mode.

Verifying the MVPN Configuration

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

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	Displays the mapping of the customer source, the customer group to data MDT source, and the data MDT group on the receiving side.

Command	Purpose
show ip pim mdt send	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	Displays the encapsulation table in MVPN. This table indicates how MVPN packets are encapsulated when sent out on the default vrf.
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 MVPNs

This section includes the following configuration examples:

- Example: Configuring MVPN, page 34-15
- Example: Configuring the Multicast Address Range for Data MDTs, page 34-16

Example: Configuring 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 rp-address 10.10.1.3 -list 239.0.0.0/8 bidir
  ip pim ssm range 232.0.0.0/8
  mdt source loopback2
  mdt default 232.1.1.1
  mdt data 232.2.2.0/24 threshold 10 route-map rmap2
  mdt data bidir-enable
vrf context vpn4
  ip pim rp-address 10.10.4.2 -list 224.0.0.0/8
  ip pim rp-address 10.10.4.3 -list 239.0.0.0/8 bidir
  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 10.11.0.3 -list 239.0.0.0/8 bidir
ip pim rp-address 10.11.0.4 -list 235.0.0.0/8
ip pim ssm range 232.0.0.0/8
```

Example: Configuring the Multicast Address Range for Data MDTs

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 239.1.0/24 threshold 10
```

Additional References for MVPNs

For additional information related to MVPN configuration, see the following sections:

- Related Documents
- Standards
- MIBs

Related Documents

Related Topic	Document Title
Multicast technology concepts	*IP Multicast Technology Overview*
VDCs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*
CLI commands	*Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference*
Basic IP multicast configuration	*Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide*

Standards

Standards	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
MPLS-VPN-MIB	To locate and download MIBs for selected platforms, releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Feature History for MVPNs

Table 34-2 lists the release history for this feature.

Table 34-2 Feature History for MVPNs

Feature Name	Releases	Feature Information
MVPNs	6.1(1)	Added support for multicast GRE tunnel interfaces for PE-CE routing with MVPN.
MVPNs	5.2(4)	Added support for multicast GRE tunnel interfaces for PE-CE routing with MVPN.
MVPN Intranet support	5.2(1)	This feature was introduced.